UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/019,244	09/01/2023	7627781		7031

199393 7590 07/26/2024

Atlantic IP
c/o Lombard Geliebter LLP
1325 Avenue of the Americas
28th Floor
New York, NY 10019

EXAMINER	
BONSHOCK, DENNIS G	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
07/26/2024	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

GREENBERG TRAURIG (NJ)
500 CAMPUS DRIVE, SUITE 400
P.O. BOX 677
FLORHAM PARK NJ 07932

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/019,244* .

PATENT UNDER REEXAMINATION *7627781* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

Final Office Action

This is a Final action of claims 1-30 of United States Patent Number: 7,627,781 issued to Michaelis et al. (hereinafter the '781 Patent). This action is responsive to Reply filed 5/20/2024. Reexamination was ordered on 10/10/2023 responsive to the Request for *ex parte* reexamination filed 9/1/2023. Responsive to the Office ordering reexamination, the Patent Owner chose not to file a Patent Owner statement.

References

- U.S. Patent No. 7,085,959 to Safford (hereinafter Safford)
- U.S. Patent No. 5,450,576 to Kennedy et al. (hereinafter Kennedy)
- U.S. Patent No. 7,251,746 to Fox (hereinafter Fox)
- U.S. Patent No. 7,328,371 to Kalyanasundharam et al. (hereinafter Kalyanasundharam)
- U.S. Patent Application Pub. No. 2004/0123201 to Nguyen et al. (hereinafter Nguyen)
- U.S. Patent No. 5,491,788 to Cepulis (hereinafter Cepulis)

Prosecution History

Original Examination:

The '781 Patent was filed as application 10/973,077 (hereinafter the '077 application) on October 25, 2004.

During the original prosecution of Michaelis patent, a Non-Final rejection was mailed on 9/20/2007. In this initial action the Office rejected the claims under obvious-type double patenting over claims 1-30 of US application 10/972,588 and further under 35 USC 102 over Safford et al. (US patent 7,085,959), while further rejecting other claims over a combination of Safford and Marshal (US patent 5,915,082).

On 12/12/2007, Patent Owner responded arguing specifically that: ***"In contrast, in Safford, transfer of the architected state of a processor is a transfer of information from one processor to another, as required for one processor to start processing instructions at the point that another processor was halted, while the system is running. Safford does not discuss transfer of system information (which processer is the boot processor) that remains permanent and survives a system reset. Safford discusses rebooting individual processors while a system is running (for example, column 4, lines 57-60, and claim3), but that is not the same as rebooting the entire system, which is initiated by the one designated boot processor after a system reset or power shutdown."***

In the Office's 3/19/2008 Non-Final rejection Examiner Le maintained the Safford based rejections while further adding an additional grounds of rejection under 35 USC 103 over Fox et al. (US patent 7,251,749) in further view of Quach (US patent

6,625,749). There was some debate as to whether Safford was valid prior art, but Patent Owner (then Applicant) eventually admitted that Safford was eligible:

> NOTE: In applicant's reply to the previous office action, applicant asserted that Safford was not proper prior art under 35 U.S.C. § 103. However, that assertion was not correct, and is hereby retracted, with apologies for the mistake. Safford was published on January 8, 2004, before the filing of the present application on October 25, 2004.
>
> *File History* (Ex. 1002), p. 229

On 9/10/2008, Patent Owner filed a Terminal Disclaimer that was approved by the Office on 9/11/2008 and further amended the claims as follows:

1. *(currently amended) A system comprising:*

a plurality of processors;

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; and

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, wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said bool processor.

In the Office's 9/23/2008 Notice of Allowance, the claims were allowed apparently in view of the claim amendments of 9/10/2008, corresponding arguments presented, and discussion in an interview on 9/5/2008.

Litigation History

The '781 Patent is involved in the following litigation:

- Foras Technologies Ltd. v. Nissan Motor Company, LTD et al., Case No. 2:23-cv-00640 (W.D. Tex., June 6, 2023)
- Foras Technologies Ltd. v. Toyota Motor Company, Inc. et al., Case No. 2:23-cv-00321 (E.D. Tex., July 12, 2023)
- Foras Technologies Ltd. v. Toyota Motor Company, Inc. et al., Case No. 2:23-cv-00150 (E.D. Tex., April 5, 2023)

Claim Rejections

In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the

time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Rejection 1 – as proposed in Grounds 2 - Safford (Ex. 1004) in view of Kennedy (Ex. 1005) - (see Request pages 73-87 and in Appendix B)

Claims 1-2, 5-6, and 9-15 are rejected under 35 U.S.C. 103(a) as being obvious over the combination of Safford (Ex. 1004) in view of Kennedy (Ex. 1005).

With regard to claim 1, teaching [pre] **A system comprising:,** Safford disclosed a computer system. For example, Safford disclosed that "a computer system employs lock stepped processor cores that operate in a master/check pair." Safford (Ex. 1004) at 2:55-57.;

With regard to claim 1, teaching [1.1] **a plurality of processors**, Safford disclosed a plurality of processors included in the computer system. For example, Safford taught that "a computer system employs lock stepped processor cores that operate in a master/check pair. Each of two processors in the pair processes the same code sequences, and the resulting outputs of the processors are compared by a logic circuit located near external interfaces of the two processors." Safford (Ex. 1004) at 2:55-60. "For example, the node controller 130 may store the architected state of the processors 111, 113, 121, 123, 125, and 127, either internally in the node controller 130, or in another component of the computer system 100." Id. at 4:6-10. Accordingly, Safford disclosed a plurality of processors included in the computer system.

With regard to claim 1, teaching [1.2] **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**, Safford disclosed, or at least rendered obvious, this limitation by disclosing memory that stores the architected states of microprocessors, including the designation of a hot standby processor. For example, Safford taught that "the architected states of each of the microprocessors 22 and 24 is copied from the memory 25 into the microprocessors 22 and 24, respectively." Safford (Ex. 1004) at 2:2-4. "For example, the node controller 130 may store the architected state of the processors 111, 113, 121, 123, 125, and 127, either internally in the node controller 130, or in another component of the computer system 100." Id. at 4:6-10. "To improve reliability of processing assets, a computer system employs lock stepped processor cores that operate in a master/check pair." Id. at 2:55-57. "The processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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. at 4:25-30. Accordingly, Safford described assigning the roles of master and checker to certain processors and the role of hot standby to another processor. See Wolfe Decl. (Ex. 1003) ¶187. Safford further described one pair of these processors "operating in a lock step mode" as initially "appear[ing] to [a] computer system...to be...a logical CPU 0" but later that logical CPU 0 designation is reassigned to another processor. See Safford (Ex. 1007) at 3:4-5, 4:52-53 ("The processor pair 125/127 then becomes the logical CPU 0 in the computer system 100'). A POSITA would have recognized Safford's logical CPU 0 as being the

designation used by computing system 100 to identify its boot processor at least because, in multi-processor system architectures like Safford's, "logical CPU 0" was commonly used to indicate that a processor had been selected as a boot processor. See Wolfe Decl. (Ex. 1003) ¶ 187. Safford lacks an explicit disclosure identifying which processor was the boot processor, Safford as modified by Kennedy would have stored information about the assigned roles of boot processor and spare processor. See, e.g., Kennedy (Ex. 1005) at 7:5-13. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In view of the above citations, Safford taught data storage that stores the architected states, such as role assignments (e.g., master/checker), of paired processors, and Kennedy taught that such assignments could be the roles of boot processor and spare processor. See Wolfe Decl. (Ex. 1003) ¶ 187. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by storing information that assigns the roles of boot processor and spare processor in case of a reboot. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of spare processor to another one of the processors. Id.

With regard to claim 1, teaching [1.3] **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, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the failover of a processor to a hot standby processor in response to a loss of lockstep. For example, Safford taught that, "[u]sing the impending loss of lock step signal, the computer system 100 may continue operating (processing) using one of the processors 111 and 113 that did not experience an error." Safford (Ex. 1004) at 3:27-30; "The six processors 111, 113, 121, 123, 125, and 127 operate in lock step (i.e., are processing code sequences). For example, the processor 111 operates in lockstep with the processor 113, and the processor 121 operates in lock step with the processor 123 and the processor 125 operates in lock step with the processor 127." Id. at 4:19-24. "The processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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. at 4:25-30. "In block 210, the processor 111 signals the node controller 130 that the first processor pair 111/113 has broken lock step and that the first processor pair 111/113 should be taken 'off-line.'" Id. at 4:37-40. Accordingly, Safford described replacing a failed processor with a hot spare processor in response to a loss of lockstep. See Wolfe Decl. (Ex. 1003) ¶ 188. The '781 Patent notes that a firmware component known as a processor abstraction layer (PAL) "copies all the needed information from memory into the internal registers of the hot spare processor so that the hot spare processor becomes a new active processor, transforming the role of the hot spare to boot processor." Ex. 1001 at 17:57-61. Safford disclosed this limitation through an identical process for transferring the role of boot processor to the spare

processor. See Safford (Ex. 1004) at 32:29-32. For example, Safford taught "moving an architected state of the processor unit experiencing the loss of lock step to the spare processor unit, wherein the spare processor unit becomes an active processor unit in the computer system." Id. In view of these citations and the knowledge that a boot processor was a common component in computer systems (e.g., Safford's system), a POSITA would have understood Safford as disclosing the failover of a boot processor to a hot spare processor in response to a loss of lockstep. See Wolfe Decl. (Ex. 1003) ¶188. However, even if Safford did not explicitly identify the failed processor as having been the boot processor, Safford as modified by Kennedy would have performed boot processor failover. See, e.g., Kennedy (Ex. 1005) at 7:5-13. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Id. In view of the above citations, Safford taught logic that causes its system to replace a failed processor with a hot spare processor in response to a loss of lockstep, and Kennedy taught logic that causes its system to fail over the boot processor to an alternate processor. See Wolfe Decl. (Ex. 1003) ¶ 188. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by providing fault tolerance and high availability in the event that the boot processor fails before a reboot occurs. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught logic operable for transferring the role of boot processor to the spare processor

during system runtime in response to detecting loss of lockstep for the boot processor. Id.

With regard to claim 1, teaching [1.4] **wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the failover to a hot standby processor that is sitting idle until the loss of lockstep. For example, Safford taught that "[t]he processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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." Safford (Ex. 1004). at 4:25-30. Accordingly, Safford as modified by Kennedy taught that the processor assigned the role of spare is held in reserve until a loss of lockstep is detected.

With regard to claim 2, teaching **The system of claim 1 wherein said data storage is non-volatile data storage**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing memory that stores the architected states of microprocessors. For example, as explained above in connection with claim 1, Safford as modified by Kennedy disclosed data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of spare processor to another one of the processors. Even if Safford did not explicitly identify the data storage as being non-volatile data storage, Safford as modified by Kennedy would have included non-volatile data storage that stores information about the assignments

of the boot processor and the spare processor. For example, "[t]he CPU1 120 has a corresponding non-volatile memory 123 which contains configuration information and CPU specific initialize and self-test code (ISTC) for the CPU1 120[.]" Kennedy (Ex. 1005), 5:15-18. From these disclosures, a POSITA would have understood that the data storage that stores the information assigning the roles of boot processor and spare processor included non-volatile memory. See Wolfe Decl. (Ex. 1003) ¶ 189. Accordingly, Safford as modified by Kennedy would have rendered the subject matter of claim 2 obvious to a POSITA.

With regard to claim 5, teaching **The system of claim 1 wherein each of said plurality of processors is a processor module that includes a master processor and a slave processor**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the selection of a processor type to boot the system. For example, Safford taught that "a computer system employs lock stepped processor cores that operate in a master/check pair." Safford (Ex. 1004) at 2:55-57. "The processors operate as pairs when in lock step (i.e., the processors 111 and 113 are a pair; the processors 121 and 123 are a second pair ; and the processors 125 and 127 are a third pair). From the node controller's perspective, each pair of processors appears as a single (logical) processor." Id. at 3:57-63. Accordingly, Safford as modified by Kennedy taught a plurality of processor modules that each include a master processor and a slave processor. See Wolfe Decl. (Ex. 1003) ¶ 192.)

With regard to claim 6, teaching **The system of claim 1 wherein said another of said plurality of processors that is assigned said role of spare processor is an active spare before being transferred said role of boot processor**, Safford as

modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a spare processor being active before a role transfer. For example, Safford taught that "the processor 111 signals the node controller 130 that the first processor pair 111/113 has broken lock step and that the first processor pair 111/113 should be taken 'off-line.'" Safford (Ex. 1004) at 4:37-40. "In block 230, the node controller 130 'reboots' the processors 111 and 113, and the processors 111 and 113 become the new 'hot standby' processor pair on the system 100." Id. at 4:57-60. Safford also taught that "the spare processor unit becomes an active processor unit in the computer system" Id., claim 1. In view of the above citations, a POSITA would have understood Safford as disclosing that a spare processor can be active before a role transfer. See Wolfe Decl. (Ex. 1003) ¶ 193. Moreover, a POSITA would have understood that the role at issue could be the boot processor assignment in view of Safford as modified by Kennedy for at least the same reasons provided above in connection with claim 1. Id. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by providing high availability through active and/or inactive spare processors. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught that the spare processor is an active spare before being transferred the role of boot processor. Id.

With regard to claim 9, teaching [9.pre] **A method comprising:**, Safford disclosed a method. "An apparatus, operating on an advanced multi-core processor

architecture, and a corresponding method, are used to enhance recovery from loss of lock step in a multi-processor computer system." Safford (Ex. 1004), Abstract

With regard to claim 9, teaching [9.1] **assigning a role of boot processor to one of a plurality of processors in a system**, Safford disclosed, or at least rendered obvious, this limitation by disclosing the architected states of microprocessors that operate as a master/checker pair as well as the designation of the logical CPU 0. For example, Safford taught that "the architected states of each of the microprocessors 22 and 24 is copied from the memory 25 into the microprocessors 22 and 24, respectively." Safford (Ex. 1004) at 2:2-4. "For example, the node controller 130 may store the architected state of the processors 111, 113, 121, 123, 125, and 127, either internally in the node controller 130, or in another component of the computer system 100." Id. at 4:6-10. "To improve reliability of processing assets, a computer system employs lock stepped processor cores that operate in a master/check pair." Id. at 2:55-57. Accordingly, Safford described assigning the roles of master and checker to certain processors. See Wolfe Decl. (Ex. 1003) ¶ 194. Safford further described one pair of these processors "operating in a lock step mode" as initially "appear[ing] to [a] computer system...to be...a logical CPU 0" but later that logical CPU 0 designation is reassigned to another processor. See Safford (Ex. 1007) at 3:4-5, 4:52-53 ("The processor pair 125/127 then becomes the logical CPU 0 in the computer system 100"). A POSITA would have recognized Safford's logical CPU 0 as being the designation used by computing system 100 to identify its boot processor at least because, in multi-processor system architectures like Safford's, "logical CPU 0" was commonly used to indicate that a processor had been selected as a boot processor. See Wolfe Decl. (Ex. 1003) ¶ 194.

However, even if Safford is interpreted as not explicitly identifying which processor was the boot processor, Safford as modified by Kennedy would have assigned the role of boot processor. See, e.g., Kennedy (Ex. 1005) at 7:5-13. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Id. at 7:9-13. In view of the above citations, Safford taught assigning roles (e.g., master/checker) to paired processors, and Kennedy taught that one such assignment could be the role of boot processor. See Wolfe Decl. (Ex. 1003) ¶ 194. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by assigning the role of boot processor to a certain processor in case of a reboot. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught assigning a role of boot processor to one of a plurality of processors in a system. Id.

With regard to claim 9, teaching [9.2] **assigning a role spare processor to another of said plurality of processors**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the designation of a hot standby processor. For example, Safford taught that "[t]he processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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." Safford (Ex. 1004) at 4:25-30. Accordingly, Safford as modified by Kennedy taught assigning a role of spare processor to another one of the plurality of processors in the system. Id.

With regard to claim 9, teaching [9.3] **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**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing failing over a processor to a hot standby processor in response to a loss of lockstep. For example, Safford taught that, "[u]sing the impending loss of lock step signal, the computer system 100 may continue operating (processing) using one of the processors 111 and 113 that did not experience an error." Safford (Ex. 1004) at 3:27-30. "The processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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. at 4:25-30. "In block 210, the processor 111 signals the node controller 130 that the first processor pair 111/113 has broken lock step and that the first processor pair 111/113 should be taken 'off-line.'" Id. at 4:37-40. Accordingly, Safford described replacing a failed processor with a hot spare processor in response to a loss of lockstep. See Id. at 3:27-30, 4:37-54. The '781 Patent notes that a firmware component known as a processor abstraction layer (PAL) "copies all the needed information from memory into the internal registers of the hot spare processor so that the hot spare processor becomes a new active processor, transforming the role of the hot spare to boot processor." Ex. 1001 at 17:57-61. Safford disclosed this claim limitation through an identical process for transferring the role of boot processor to the spare processor. Safford (Ex. 1004) at 32:29-32. For example, Safford taught "moving an architected state of the processor unit experiencing the loss of lock step to the spare

processor unit, wherein the spare processor unit becomes an active processor unit in the computer system." Id. In view of this disclosure and the knowledge that a boot processor was a common component in computer systems (e.g., Safford's system), a POSITA would have understood Safford as disclosing the failover of a boot processor to a hot spare processor in response to a loss of lockstep. See Wolfe Decl. (Ex. 1003) ¶ 195. However, even if Safford is not interpreted as explicitly identifying the failed processor as having been the boot processor, Safford as modified by Kennedy would have performed boot processor failover. See, e.g., Kennedy (Ex. 1005) at 7:5-13. In view of the above citations, Safford taught replacing a failed processor with a hot spare processor in response to a loss of lockstep, and Kennedy taught failing over the boot processor to an alternate processor. See Wolfe Decl. (Ex. 1003) ¶ 195. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by providing high availability in the event that the boot processor fails before a reboot occurs. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught changing, during system runtime, the assigned role of boot processor to the spare processor in response to loss of lockstep for the boot processor. Id.

With regard to claim 9, teaching [9.4] **responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system**, Safford as modified by Kennedy disclosed, or at

least rendered obvious, this limitation by disclosing the improved availability of processor assets to avoid a computer system crash and by disclosing a central boot ROM that is used by the operating system and stores the boot code executed in conjunction with the operating system to boot the system. For example, in focusing on "improv[ing] availability of the processor assets of the computer system", Safford notes that "loss of lock step, if not promptly corrected, may cause the computer system 18 to crash." Safford (Ex. 1004) at 3:21-22, 1:45-47. Safford lacks an explicit disclosure of an operating system that recognizes the spare processor as the boot processor without requiring shutdown, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA. For example, according to Kennedy, "a central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53-55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. A POSITA would have understood that, because Kennedy's central boot ROM 150 is used by the operating system and stores the boot code executed in conjunction with the operating system to boot the system, the operating system would recognize the spare processor as the boot processor after failover without requiring shutdown of the operating system, especially if Kennedy is to achieve its objective of providing fault tolerance and high availability. See Kennedy (Ex. 1005) at 9:1-5, 9:53-55, 10:42-45; see also Wolfe Decl. (Ex. 1003) ¶ 196. This conclusion of a POSITA's understanding is also supported by Safford's objective of avoiding system crashes. See Safford (Ex. 1004) at 3:21-22, 1:45-47. A POSITA would have been motivated to combine these disclosures from Safford

and Kennedy to improve the reliability of Safford's system by providing fault tolerance and high availability in the event that the boot processor fails before a reboot occurs. See Wolfe Decl. (Ex. 1003) ¶ 196. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy would have rendered obvious to a POSITA an operating system that recognizes the spare processor as the boot processor without requiring shutdown. Id.

With regard to claim 10, **Claim 10 recites subject matter that is substantially similar to claim 2** and is therefore rendered obvious for the same reasons as claim 2 above.

With regard to claim 11, **Claim 11 recites subject matter that is substantially similar to claim 2** and is therefore rendered obvious for the same reasons as claim 2 above.

With regard to claim 12, teaching **The method of claim 9 further comprising:** [12.1] **determining at least one processor that is coupled to a common communication bus with said one of said plurality processors assigned to the role of boot processor**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a determinization that the alternate processor is installed in one of the slots of a backplane bus and that the boot processor is installed in another one of the slots of the backplane bus. Safford does not explicitly disclose determining a processor that is coupled to a common communication bus with the boot processor, however Safford as modified by Kennedy would have determined that the alternate processor and the boot processor are coupled to the same backplane

bus. For example, according to Kennedy, "the system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. In view of these citations, Kennedy taught that the boot processor and the alternate boot processors are installed in specific slots along the same backplane bus. Based on this disclosure from Kennedy, a POSITA would have understood that any alternate processor installed in one of the slots of the backplane bus would have been coupled to a common communication bus with the boot processor. See Wolfe Decl. (Ex. 1003) ¶ 197. As a result, any form of identifying such an alternate processor would constitute a determination of an alternate processor coupled to a common communication bus with the boot processor. Id. A POSITA would have been motivated to combine these disclosures from Kennedy with Safford to improve the communication capabilities of Safford's system for fault tolerance and high availability. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught this claim limitation. Id.

With regard to claim 12, teaching [12.2] **selecting said another of said plurality of processors to be assigned the role of spare processor from said determined at least one processor**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the selection of one or more alternate (e.g., spare) boot processors by installing the alternate boot processor into a specific slot corresponding to the alternate boot role on the backplane bus. Even if Safford did not

explicitly disclose selecting the processor coupled to the common communication bus as the spare processor, Safford as modified by Kennedy would have selected the alternate processor installed in the same backplane bus as the boot processor to be the spare. For example, according to Kennedy, "the system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. A POSITA would have been motivated to combine these disclosures from Kennedy with Safford to improve the reliability of Safford's system by providing fault tolerance and high availability in the event that the boot processor fails before a reboot occurs. See Wolfe Decl. (Ex. 1003) ¶ 198. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught this claim limitation. Id.

With regard to claim 13, teaching **The method of claim 9 further comprising: during system boot-up, system firmware identifying the spare processor and holding said spare processor in reserve as a hot spare**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a BIOS responsible for identifying a spare processor and holding the same in reserve as a hot spare. Safford taught identifying a spare processor and holding the spare processor in reserve as a hot spare for at least the same reasons provided above in connection with claim 1. Even if Safford did not explicitly disclose doing so via system firmware, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA.

For example, according to Kennedy, "central boot ROM 150 comprises a read-only-memory with the basic input/output system (BIOS) instruction set for an INTEL 80486 or 80386 microprocessor. Accordingly, in the present embodiment, at least one CPU connected to the 64-bit bus is, or emulates, an INTEL 80486 or 80386 microprocessor in order to boot the system." Kennedy (Ex. 1005) at 4:25-33. "[A] central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53-55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 199. In view of the above citations and this common knowledge, a POSITA would have understood that Kennedy's BIOS constitutes firmware responsible for identifying a spare processor and holding the same in reserve as a hot spare. Id. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to achieve cost savings by performing these tasks with firmware instead of using additional hardware. See id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy would have rendered the subject matter of claim 13 obvious to a POSITA. Id.

With regard to claim 14, teaching **The method of claim 9 further comprising: during system boot-up, system firmware booting the spare processor as an active spare**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a BIOS responsible for booting the spare processor as an

active spare. Safford taught booting the spare processor as an active spare for at least the same reasons provided above in connection with claim 6. Even if Safford did not explicitly disclose doing so via system firmware, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA. For example, according to Kennedy, "central boot ROM 150 comprises a read-only-memory with the basic input/output system (BIOS) instruction set for an INTEL 80486 or 80386 microprocessor. Accordingly, in the present embodiment, at least one CPU connected to the 64-bit bus is, or emulates, an INTEL 80486 or 80386 microprocessor in order to boot the system." Kennedy (Ex. 1005) at 4:25-33. "[A] central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53-55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 200. In view of the above citations and this common knowledge, a POSITA would have understood that Kennedy's BIOS constitutes firmware responsible for booting the spare processor as an active spare. Id. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to enable Safford's system to dynamically modify the activity of spare processors (e.g., between active spare and hot spare) to increase efficiency and/or throughput. See id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy would have rendered the subject matter of claim 14 obvious to a POSITA. Id.

With regard to claim 15, teaching **The method of claim 9 wherein system firmware performs said changing operation**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a BIOS included in an IOSM responsible for changing the role of boot processor to the spare processor. Safford taught performing the changing operation for at least the same reasons provided above in connection with claim 9. Even if Safford did not explicitly disclose doing so via system firmware, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA. For example, Kennedy taught that each CPU includes a slot selection (sometimes referred to as "SS") logic for selecting which CPU is designated as the boot processor. "In general, the slot select logic 148 on the default boot CPU allows this CPU to boot." Kennedy (Ex. 1005) at 10:29-31, FIG. 3. Kennedy's input/output services module ("IOSM") is responsible for changing the role of boot processor to the spare processor in conjunction with the CPUs' slot selection logic. See Id. at 6:36-38, FIG. 3. For example, "[a]ny other slot selection circuits on CPUs installed in the system have a corresponding slot selection line from the IOSM 108." Id. As demonstrated by FIG. 3 of Kennedy reproduced below, Kennedy's IOSM includes the BIOS.

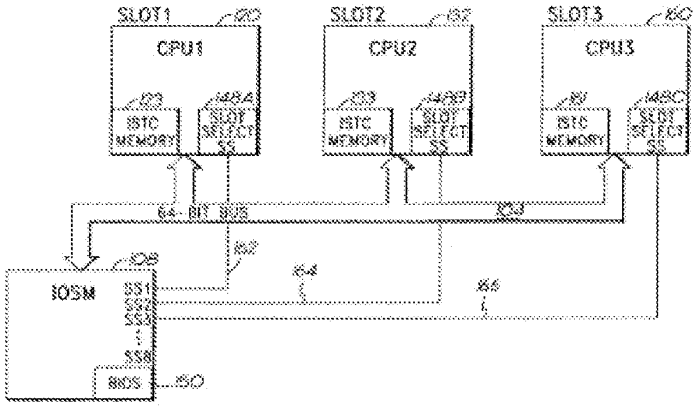


FIG. 3

The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 201. In view of the above citations and this common knowledge, a POSITA would have understood that that Kennedy's BIOS, which constitutes firmware, is responsible for changing the role of boot processor to the spare processor in conjunction with the CPUs' slot selection logic. Id. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to achieve cost savings by performing the change operation with firmware instead of using additional hardware. See id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy would have rendered the subject matter of claim 15 obvious to a POSITA.

Rejection 2 – as proposed in Ground 3 - Kennedy (Ex. 1005) in view of Nguyen (Ex. 1008) and Fox (Ex. 1006)) - (see Request pages 88-106 and in Appendix C)

Claims 1-2, 5-18, and 21-30 are rejected under 35 U.S.C. 103(a) as being obvious over the combination of Kennedy (Ex. 1005) in view of Nguyen (Ex. 1008) and Fox (Ex. 1006).

With regard to claim 1, teaching [1.pre] **A system comprising:,** Kennedy disclosed a system. For example, Kennedy taught "multiprocessor computer systems." Kennedy (Ex. 1005) at 1:14-15.

With regard to claim 1, teaching [1.1] **a plurality of processors**, Kennedy disclosed this limitation by disclosing multiprocessor computer systems that enable a boot processor to fail over to an alternate processor. For example, Kennedy taught "multiprocessor computer systems." Kennedy (Ex. 1005) at 1:14-15. Kennedy's disclosure "relates to a method and apparatus to automatically control the determination of which processor in the system will boot the system." Id. at 1:15-17. Kennedy further disclosed that, "[i]f the boot processor fails, control automatically shifts to an alternate processor." Id. at 1:17-19. Accordingly, Kennedy taught a plurality of processors.

With regard to claim 1, teaching [1.2] **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**, Kennedy disclosed, or at least rendered obvious, this limitation by disclosing memories

that hold configuration information about processors that are assigned the roles of boot processor and spare processor. For example, Kennedy taught that "[t]he CPU ISTC memories hold configuration information." Kennedy (Ex. 1005) at 7:33-35. Kennedy also taught that, "[a]s previously explained, the ISTC memory associated with each CPU contains...configuration information about the CPU. Thus, the boot CPU determines how many CPUs are present on the bus." Id. at 9:16-21. Kennedy further taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. In view of the above citations, Kennedy taught memories that store CPU configuration information that includes the roles of boot processor and alternate processor in the system. See Wolfe Decl. (Ex. 1003) ¶ 204. Accordingly, Kennedy taught data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of spare processor to another one of the processors. Id.

With regard to claim 1, teaching [1.3] **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**, Kennedy as modified by Nguyen disclosed, or at least rendered obvious, this limitation by disclosing the transfer of the boot processor role to the alternate processor in response to a loss of lockstep. For example, Kennedy taught that, "if this 'default'? boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system."

Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. Even if Kennedy did not explicitly identify the boot processor's failure as being a loss of lockstep, Kennedy as modified by Nguyen would have detected a loss of lockstep. See, e.g., Nguyen (Ex. 1008) ¶ [0008], [0024], [0054]-[0055]. For example, Nguyen taught that "[e]xecution cores 120 operate in lock step." Nguyen (Ex. 1008) ¶ [0024]. "FRC- hardware compares results generated by each core, and if a discrepancy is detected, the FRC system passes control to an error-handling routine." Id. ¶ [0006]. "However, for FRC-enabled processors, such errors are likely to be manifested as FRC errors, since they take the execution cores out of lock-step." Id. ¶ [0008]. "[E]ach execution core may include a status bit that is set if an error is detected. The master execution core may check this bit to determine if it has also generated an error." Id. ¶ [0050]. Accordingly, Nguyen described detecting a loss of lockstep. In view of the above citations, Kennedy taught logic that causes its system to replace the boot processor with an alternate processor in response to a failure of the boot processor, and Nguyen taught detecting an error (i.e., a loss of lockstep). See Wolfe Decl. (Ex. 1003) ¶ 205. A POSITA would have been motivated to combine these disclosures from Kennedy and Nguyen to improve the reliability of Kennedy's system by providing fault tolerance and high availability in the event that the boot processor fails through a loss of lockstep. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen in this manner, and thus this combination of Kennedy and Nguyen would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen taught logic

operable for transferring the role of boot processor to the spare processor during system runtime in response to detecting loss of lockstep for the boot processor. Id.

With regard to claim 1, teaching [1.4] **wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the transfer of the boot processor role to the spare processor in response to detecting loss of lockstep and by disclosing the automatic failover to a hot spare processor. For example, Kennedy as modified by Nguyen taught a spare processor that assumes the role of boot processor in response to a loss of lockstep for at least the reasons provided above. Even if Kennedy did not explicitly identify the spare processor as having been a hot spare processor that is held in reserve, Kennedy as modified by Nguyen and Fox would have transferred the role of boot processor to a hot spare processor. See, e.g., Fox (Ex. 1006), Abstract. For example, according to Fox, "[t]he invention provides a substantially seamless transfer of executing processes from the failing processor to the replacement processor, without halting the execution of the operating system (OS) and/or applications on the overall MP. That is, a failing processor is able to automatically fail-over to a hot-spare processor without affecting the OS or executing applications." Fox (Ex. 1006) at 4:16-23. In view of the above citations, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, transferring the role of boot processor to a hot spare processor during system runtime in response to detecting loss of lockstep for the boot processor. See Wolfe Decl. (Ex. 1003) ¶ 206. A POSITA would have been motivated to

combine these disclosures from Kennedy, Nguyen, and Fox to decrease delays in failing over from the boot processor to the spare processor. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught that the processor assigned the role of spare is held in reserve as a hot spare processor until the loss of lockstep is detected for the boot processor. Id.

With regard to claim 2, teaching **The system of claim 1 wherein said data storage is non-volatile data storage**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing non-volatile memory that stores the configuration information in the system. For example, as explained above in connection with claim 1, Kennedy as modified by Nguyen and Fox taught data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of spare processor to another one of the processors. Kennedy further taught that "[t]he CPU1 120 has a corresponding non-volatile memory 123 which contains configuration information and CPU specific initialize and self-test code (ISTC) for the CPU1 120." Kennedy (Ex. 1005), 5:15-18. From these disclosures, a POSITA would have understood that the data storage that stores the information assigning the roles of boot processor and spare processor included non-volatile memory. See Wolfe Decl. (Ex. 1003) ¶ 207. Accordingly, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 2 obvious to a POSITA. Id.

With regard to claim 5, teaching **The system of claim 1 wherein each of said plurality of processors is a processor module that includes a master processor and a slave processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing multicore processors that each include master and slave execution cores. Even if Kennedy did not explicitly disclose that each of its processors is a processor module that includes master and slave processors, Kennedy as modified by Nguyen and Fox would have incorporated processor modules that each include master and slave execution cores. For example, Nguyen "relates to microprocessors and, in particular, to mechanisms for handling errors in FRC-enabled processors." Nguyen (Ex. 1008) ¶ [0002]. Nguyen also taught "a mechanism for handling recoverable and non-recoverable errors in multi-core processors." Id. ¶ [0058]. "Processor 110 includes first and second execution cores 120(a), 120(b)." Id. ¶ [0020]. "For one embodiment, one of execution cores may be designated as the master core and the other as the slave core when processor 10 is operating in FRC mode." Id. ¶ [0047]. In view of the above citations, Kennedy as modified by Nguyen and Fox taught multicore processor modules that each include master and slave execution cores. See Wolfe Decl. (Ex. 1003) ¶ 210. A POSITA would have been motivated to combine these disclosures from Kennedy, Nguyen, and Fox to improve the spatial efficiency of Kennedy's system by consolidating the amount of space occupied by the processors. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox

taught that each processor is a processor module including a master processor and a slave processor.

With regard to claim 6, teaching **The system of claim 1 wherein said another of said plurality of processors that is assigned said role of spare processor is an active spare before being transferred said role of boot processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the spare as active before the role of boot processor is transferred. Even if Kennedy did not explicitly disclose that the alternate processor is an active spare before being transferred the role of boot processor, Kennedy as modified by Nguyen and Fox would have included the alternate processor as an active spare before being transferred the role of boot processor. For example, Fox's BIOS "brings the standby processor out of the held-off state and into an active state." Fox (Ex. 1006) at 7:8-10. In view of this disclosure, Kennedy as modified by Nguyen and Fox taught that the alternate processor is an active spare before being transferred the role of boot processor. See Wolfe Decl. (Ex. 1003) ¶ 211. A POSITA would have been motivated to combine these disclosures from Kennedy, Nguyen, and Fox to improve the usage and/or efficiency of the processors in the system. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught that the spare is active before being transferred the role of boot processor. Id.

With regard to claim 7, teaching **The system of claim 1 further comprising: an operating system, wherein, responsive to the logic transferring the role of boot**

processor to the spare processor, the operating system recognizes the spare processor as the boot processor, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the failover to a hot spare processor without affecting the OS or executing applications. Even if Kennedy did not explicitly disclose an operating system that recognizes the spare processor as the boot processor after the role transfer, Kennedy as modified by Nguyen and Fox would have rendered this claim limitation obvious to a POSITA. For example, Fox taught that "[t]he invention provides a substantially seamless transfer of executing processes from the failing processor to the replacement processor, without halting the execution of the operating system (OS) and/or applications on the overall MP. That is, a failing processor is able to automatically fail-over to a hot-spare processor without affecting the OS or executing applications." Fox (Ex. 1006) at 4:16-25. Fox also taught that, "once the spare processor has been successfully brought on line, the BIOS (table) is updated and the OS [and] bus controllers...are messaged to change their configurations to reflect a processor change from the failing processor to the spare processor." Id. at 8:10-14. "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., Abstract, 3:28-31, 6:64-67. In view of the above citations, a POSITA would have understood that, because Fox's failing processor automatically fails over to a hot spare processor without affecting the operating system and then control is returned to the operating system which updates the "active" processor list, the operating system would recognize the spare processor as having assumed the role of the failing processor after the role transfer. See Wolfe Decl. (Ex. 1003) ¶ 212. Or at the very least,

Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 7 obvious to a POSITA due at least in part to Fox's failing processor automatically being replaced by a hot spare processor without affecting the operating system and then having control returned to the operating system which updates the "active" processor list. Id. It was well within the skill of a POSITA to modify Nguyen's system by Kennedy and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id.

With regard to claim 8, teaching **The system of claim 1 wherein said plurality of processors are Itanium Processor Family (IPF) processors**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing IPF processors. Even if Kennedy did not explicitly disclose that its processors are IPF processors, Kennedy as modified by Nguyen and Fox would have included IPF processors. For example, Nguyen taught that "[o]ne embodiment of the recovery mechanism employs code that is closely associated with the processor. For example, the Itanium Processor Family of Intel Corporation employs a layer of firmware called processor abstraction layer (PAL), which provides an abstraction of the processor to the rest of the computing system." Nguyen (Ex. 1008) ¶ [0045]. In view of the above citations, Kennedy as modified by Nguyen and Fox taught that the processors would have been IPF processors. See Wolfe Decl. (Ex. 1003) 7 213. A POSITA would have been motivated to combine these disclosures from Kennedy, Nguyen, and Fox to improve the spatial efficiency of the system by consolidating the processors into the multicore processor package used by the IPF. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this

combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught this claim limitation.

With regard to claim 9, teaching [9.pre] **A method comprising:**, Kennedy disclosed a method. For example, Kennedy taught "a method and apparatus to automatically control the determination of which processor in the system will boot the system." Kennedy (Ex. 1005) at 1:15-17.

With regard to claim 1, teaching [9.1] **assigning a role of boot processor to one of a plurality of processors in a system**, Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a processor that is assigned the role of boot processor in a multiprocessor system. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. Kennedy also taught "multiprocessor computer systems." Id. at 1:14-15. In view of the above citations, Kennedy taught assigning the role of boot processor to one of the processors in the system. See Wolfe Decl. (Ex. 1003) ¶ 214. Accordingly, Kennedy taught assigning the role of boot processor to one of the processors in the system. Id.

With regard to claim 9, teaching [9.2] **assigning a role spare processor to another of said plurality of processors**, Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a processor that is assigned the role of alternate processor in a multiprocessor system. For example, Kennedy taught that, "if this

'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. Kennedy also taught "multiprocessor computer systems." Id. at 1:14-15. In view of the above citations, Kennedy taught assigning the role of alternate processor to one of the processors in the system. See Wolfe Decl. (Ex. 1003) ¶ 215. Accordingly, Kennedy taught assigning the role of spare processor to one of the processors in the system. Id.

With regard to claim 9, teaching [9.3] **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**, Kennedy as modified by Nguyen disclosed, or at least rendered obvious, this limitation by disclosing the replacement of the boot processor with the alternate processor in response to a loss of lockstep for the boot processor. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. Even if Kennedy did not explicitly identify the boot processor's failure as being a loss of lockstep, Kennedy as modified by Nguyen would have detected a loss of lockstep. See, e.g., Nguyen (Ex. 1008) ¶ [0008], [0024], [0054]-[0055]. For example, Nguyen taught that "[e]xecution cores 120 operate in lock step." Nguyen (Ex. 1008) ¶ [0024]. "FRC-

hardware compares results generated by each core, and if a discrepancy is detected, the FRC system passes control to an error-handling routine." Id. ¶ [0006]. "However, for FRC-enabled processors, such errors are likely to be manifested as FRC errors, since they take the execution cores out of lock-step." Id. ¶ [0008]. "[E]ach execution core may include a status bit that is set if an error is detected. The master execution core may check this bit to determine if it has also generated an error." Id. ¶ [0050]. Accordingly, Nguyen described detecting a loss of lockstep. In view of the above citations, Kennedy taught replacing the boot processor with an alternate processor in response to a failure of the boot processor, and Nguyen taught detecting an error (i.e., a loss of lockstep). See Wolfe Decl. (Ex. 1003) ¶ 216. A POSITA would have been motivated to combine these disclosures from Kennedy and Nguyen to improve the reliability of Kennedy's system by providing fault tolerance and high availability in the event that the boot processor fails through a loss of lockstep. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen in this manner, and thus this combination of Kennedy and Nguyen would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen taught changing the assignment of the role of boot processor to the spare processor during system runtime in response to detecting loss of lockstep for the boot processor. Id.

With regard to claim 9, teaching [9.4] **responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the failover to a hot

spare processor without affecting the OS or executing applications. Even if Kennedy did not explicitly disclose an operating system that recognizes the spare processor as the boot processor after the role transfer, Kennedy as modified by Nguyen and Fox would have rendered this claim limitation obvious to a POSITA. For example, Fox taught that "[t]he invention provides a substantially seamless transfer of executing processes from the failing processor to the replacement processor, without halting the execution of the operating system (OS) and/or applications on the overall MP. That is, a failing processor is able to automatically fail-over to a hot-spare processor without affecting the OS or executing applications." Fox (Ex. 1006), 4:16-25. Fox also taught that, "once the spare processor has been successfully brought on line, the BIOS (table) is updated and the OS [and] bus controllers...are messaged to change their configurations to reflect a processor change from the failing processor to the spare processor." Id. 8:10-14. "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., Abstract, 3:28-31, 6:64-67. In view of these citations, a POSITA would have understood that, because Fox's failing processor automatically fails over to a hot spare processor without affecting the operating system and then control is returned to the operating system which updates the "active" processor list, the operating system would recognize the spare processor as having assumed the role of the failing processor after failover. See Wolfe Decl. (Ex. 1003) ¶ 217. Or at the very least, Kennedy as modified by Nguyen and Fox would have rendered this subject matter obvious to a POSITA due at least in part to Fox's failing processor automatically failing over to a hot spare processor without affecting the operating system or applications. Id.

It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id.

With regard to claim 10, **Claim 10 recites subject matter that is substantially similar to Claim 2** and is therefore rendered obvious for the same reasons as Claim 2 above.

With regard to claim 11, **Claim 11 recites subject matter that is substantially similar to Claim 2** and is therefore rendered obvious for the same reasons as Claim 2 above.

With regard to claim 12, teaching **The method of claim 9 further comprising: [12.1] determining at least one processor that is coupled to a common communication bus with said one of said plurality processors assigned to the role of boot processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the processors assigned the roles of boot processor and spare processor as being coupled to a common communication bus. For example, Kennedy taught that "the system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. Based on this disclosure from Kennedy, a POSITA would have understood that any alternate processor installed in one of the slots of the backplane bus would have been coupled to a common communication bus with the boot processor. See Wolfe Decl. (Ex. 1003) ¶ 218. As a

result, any form of identifying such an alternate processor would constitute a determination of an alternate processor coupled to a common communication bus with the boot processor. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught this claim limitation. Id.

With regard to claim 12, teaching [12.2] **selecting said another of said plurality of processors to be assigned the role of spare processor from said determined at least one processor**, Kennedy as modified by Nguyen and Fox taught assigning the role of spare to a certain processor for at least the reasons provided above in connection with claim 1. Moreover, Kennedy as modified by Nguyen and Fox taught determining that the spare processor is coupled to the common communication bus for at least the same reasons provided above in connection with this claim. Accordingly, Kennedy as modified by Nguyen and Fox taught selecting a processor coupled to the common communication bus to be assigned the role of spare. See Wolfe Decl. (Ex. 1003) ¶ 219.

With regard to claim 13, teaching **The method of claim 9 further comprising: during system boot-up, system firmware identifying the spare processor and holding said spare processor in reserve as a hot spare**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing a BIOS responsible for identifying a spare processor and holding the same in reserve as a hot spare. As explained above in connection with claim 1, Kennedy as modified by Nguyen and Fox taught that the processor assigned the role of spare is held in reserve as a hot spare processor until the loss of lockstep is detected for the boot processor. Moreover, according to Kennedy, "central boot ROM 150 comprises a read-only-

memory with the basic input/output system (BIOS) instruction set for an INTEL 80486 or 80386 microprocessor. Accordingly, in the present embodiment, at least one CPU connected to the 64-bit bus is, or emulates, an INTEL 80486 or 80386 microprocessor in order to boot the system." Kennedy (Ex. 1005) at 4:25-33. "[A] central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53-55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 220. In view of the above citations and this common knowledge, a POSITA would have understood that Kennedy's BIOS constitutes firmware responsible for identifying a spare processor and holding the same in reserve as a hot spare. Id. Accordingly, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 13 obvious to a POSITA. Id.

With regard to claim 14, teaching **The method of claim 9 further comprising: during system boot-up, system firmware booting the spare processor as an active spare**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing a BIOS responsible for booting the spare processor as an active spare. As explained above in connection with claim 6, Kennedy as modified by Nguyen and Fox taught that that the spare is active before being transferred the role of boot processor. Moreover, according to Kennedy, "central boot ROM 150 comprises a read-only-memory with the basic input/output system (BIOS) instruction set for an INTEL 80486 or 80386 microprocessor. Accordingly, in the present embodiment, at

least one CPU connected to the 64-bit bus is, or emulates, an INTEL 80486 or 80386 microprocessor in order to boot the system." Kennedy (Ex. 1005) at 4:25-33. "[A] central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53- 55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 221. In view of the above citations and this common knowledge, a POSITA would have understood that Kennedy's BIOS constitutes firmware responsible for booting the spare processor as an active spare. Id. Accordingly, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 14 obvious to a POSITA. Id.

With regard to claim 15, teaching **The method of claim 9 wherein system firmware performs said changing operation**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing a BIOS included in an IOSM responsible for changing the role of boot processor to the spare processor. As explained above in connection with claim 9, Kennedy as modified by Nguyen and Fox taught performing the changing operation. Moreover, Kennedy taught that each CPU includes a slot selection (sometimes referred to as "SS") logic for selecting which CPU is designated as the boot processor. "In general, the slot select logic 148 on the default boot CPU allows this CPU to boot." Kennedy (Ex. 1005) at 10:29-31, FIG. 3. Kennedy's input/output services module (""IOSM") is responsible for changing the role of boot processor to the spare processor in conjunction with the CPUs' slot selection logic. See id. at 6:36-38, FIG. 3. For example, "[a]ny other slot selection circuits on

CPUs installed in the system have a corresponding slot selection line from the IOSM 108." Id. As demonstrated by FIG. 3 of Kennedy reproduced below, Kennedy's IOSM includes the BIOS.

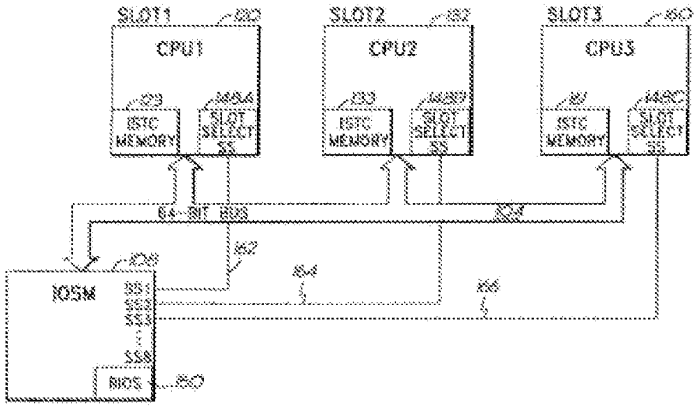


FIG. 3

The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 222. In view of the above citations and this common knowledge, a POSITA would have understood that that Kennedy's BIOS, which constitutes firmware, is responsible for changing the role of boot processor to the spare processor in conjunction with the CPUs' slot selection logic. Id. Accordingly, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 15 obvious to a POSITA. Id.

With regard to claim 16, teaching [16.pre**] A system comprising:**, Kennedy disclosed a system. For example, Kennedy taught "multiprocessor computer systems." Kennedy (Ex. 1005) at 1:14-15.

With regard to claim 16, teaching [16.1] **a plurality of lockstep processor modules that each includes a master processor and a slave processor that is in**

lockstep with said master processor, Kennedy as modified by Nguyen disclosed, or at least rendered obvious, this limitation by disclosing multicore processors that each include master and slave execution cores in lockstep. For example, Kennedy taught "multiprocessor computer systems." Kennedy (Ex. 1005) at 1:14-15. Kennedy's disclosure "relates to a method and apparatus to automatically control the determination of which processor in the system will boot the system." Id. at 1:15-17. Kennedy further disclosed that, "[i]f the boot processor fails, control automatically shifts to an alternate processor." Id. at 1:17-19. Even if Kennedy did not explicitly disclose that each of its processors is a processor module that includes master and slave processors in lockstep, Kennedy as modified by Nguyen would have incorporated processor modules that each include master and slave execution cores in lockstep. For example, Nguyen "relates to microprocessors and, in particular, to mechanisms for handling errors in FRC- enabled processors." Nguyen (Ex. 1008) ¶[[0002]. Nguyen also taught "a mechanism for handling recoverable and non-recoverable errors in multi-core processors." Id. ¶ [0058]. "Processor 110 includes first and second execution cores 120(a), 120(b)." Id. ¶ [0020]. "For one embodiment, one of execution cores may be designated as the master core and the other as the slave core when processor 10 is operating in FRC mode." Id. ¶ [0047]. "Execution cores 120 operate in lock step." Id. at ¶ [0024]. "FRC-hardware compares results generated by each core, and if a discrepancy is detected, the FRC system passes control to an error-handling routine." Id. ¶ [0006]. "However, for FRC-enabled processors, such errors are likely to be manifested as FRC errors, since they take the execution cores out of lock-step." Id. ¶ [0008]. In view of the above citations, Kennedy as modified by Nguyen taught multicore

processor modules that each include master and slave execution cores in lockstep. See Wolfe Decl. (Ex. 1003) ¶ 223. A POSITA would have been motivated to combine these disclosures from Kennedy and Nguyen to improve the spatial efficiency of Kennedy's system by consolidating the amount of space occupied by the processors. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen in this manner, and thus this combination of Kennedy and Nguyen would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen taught a plurality of lockstep processor modules that each include a master processor and a slave processor in lockstep with each other. Id.

With regard to claim 16, teaching [16.2] **a first communication bus with which at least a first two of said plurality of lockstep processor modules are communicatively coupled**, Kennedy as modified by Nguyen disclosed, or at least rendered obvious, this limitation by disclosing the processors assigned the roles of boot processor and spare processor as being coupled to a common communication bus. For example, Kennedy taught that "the system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. In view of these citations, Kennedy as modified by Nguyen taught that the boot processor and the alternate processor are installed in slots on the backplane bus. See Wolfe Decl. (Ex. 1003) ¶ 224. In other words, Kennedy's backplane bus is communicatively coupled to the boot processor and alternate processor. See Id. Accordingly, Kennedy as modified by Nguyen taught a first

communication bus with which at least a first two of the plurality of lockstep processor modules are communicatively coupled. Id.

With regard to claim 16, teaching [16.3] **a second communication bus with which at least a second two of said plurality of lockstep processor modules are communicatively coupled**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the processors as being interconnected with each other via system buses, processor-to-processor buses, and control buses. As explained above in connection with this claim, Kennedy taught a backplane bus that is communicatively coupled to the boot processor and alternate processor. Even if Kennedy did not explicitly disclose a second communication bus with which at least a second two of the lockstep processor modules are communicatively coupled, Kennedy as modified by Nguyen and Fox would have rendered this claim limitation obvious to a POSITA. For example, Fox taught that "[a] system bus 110 connects the memory controller 107 and other components to IA-32 processors 120, 122, via bridges 112, and processor buses 114. The IA-32 processors 120, 122 and the bridges 112 are grouped at nodes 111." Fox (Ex. 1006) at 5:9-12, FIG. 1. "In one alternate implementation, processors 120, 122 are interconnected to each other via processor-to-processor buses 203." Id. at 5:41-43. "Processors 120, 122 are also connected to bridge (bus controller) 207 via individual control buses 205." Id. at 5:43-45. In view of these citations, Kennedy as modified by Nguyen and Fox taught a plurality of processors that are interconnected to each over via two or more different buses (e.g., system bus 110, processor-to-processor buses 203, and individual control buses 205). See Wolfe Decl. (Ex. 1003) ¶ 225. As further demonstrated by FIG. 1 of Fox reproduced

below, Fox's system includes various processors that are communicatively coupled to one another via multiple communication buses.

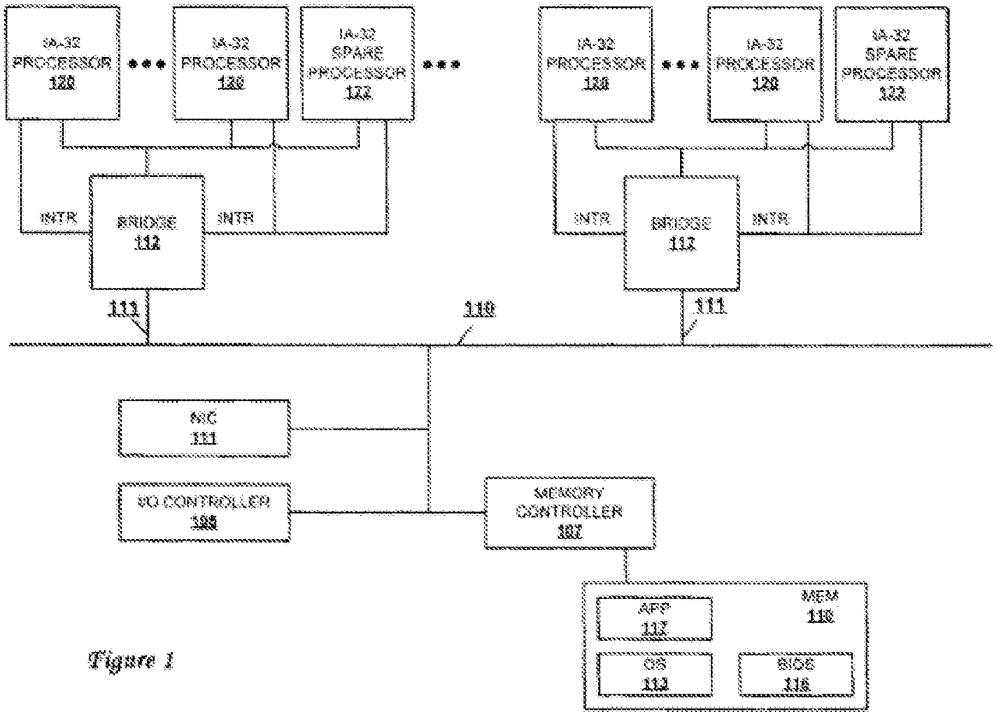


Figure 1

A POSITA would have been motivated to combine these disclosures from Kennedy and Nguyen and Fox to provide multiple communication channels and/or mechanisms for communicating with the processors. See Wolfe Decl. (Ex. 1003) ¶ 225. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught a second communication bus with which at least a second two of the plurality of lockstep processor modules are communicatively coupled. Id.

With regard to claim 16, teaching [16.4] **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, wherein said spare processor is communicatively coupled to the same one of said first and second communication buses as said boot processor,** Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing memories that hold configuration information about processors that are assigned the roles of boot processor and spare processor and communicatively coupled to the same communication bus. For example, Kennedy taught that "[t]he CPU ISTC memories hold configuration information." Kennedy (Ex. 1005) at 7:33-35. Kennedy also taught that, "[a]s previously explained, the ISTC memory associated with each CPU contains...configuration information about the CPU. Thus, the boot CPU determines how many CPUs are present on the bus." Id. at 9:16-21. Kennedy further taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. Kennedy's "system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. In view of the above citations, Kennedy as modified by Nguyen and Fox taught memories that store CPU configuration information that includes the roles of boot processor and alternate processor, which are assigned to

processors coupled to the same communication bus. See Wolfe Decl. (Ex. 1003) ¶ 226. Accordingly, Kennedy as modified by Nguyen and Fox taught data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of Spare processor to another one of the processors, wherein the boot processor and the spare processor are coupled to the same communication bus. Id.

With regard to claim 17, **Claim 17 recites subject matter that is substantially similar to claim 15**, which depends from claim 9, and is therefore rendered obvious for the same reasons as claim 15 in view of claim 9 above.

With regard to claim 18, **Claim 18 recites subject matter that is substantially similar to claim 15**, which depends from claim 9, and is therefore rendered obvious for the same reasons as claim 15 in view of claim 9 above.

With regard to claim 21, teaching **The system of claim 20 wherein said user interface presents for selection as said spare processor at least one processor that is communicatively to the same one of said first and second communication buses as said boot processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by implicitly disclosing the selection of a processor type that does not boot the system. For example, Kennedy as modified by Nguyen and Fox taught a spare processor that is communicatively coupled to the same communication bus as the boot processor for at least the same reasons provided above in connection with claim 16. Kennedy as modified by Nguyen and Fox also rendered obvious user interfaces that facilitate selecting the spare processor for at least the same reasons provided above in connection with claim 4. Accordingly, for at least the same reasons provided above in connection with claims 4 and 16, Kennedy as modified by

Nguyen and Fox would have rendered the subject matter of claim 21 obvious to a POSITA. See Wolfe Decl. (Ex. 1003) ¶ 227.).

With regard to claim 22, **Claim 22 recites subject matter that is substantially similar to claim 13** and is therefore rendered obvious for the same reasons as claim 13 above.

With regard to claim 23, **Claim 23 recites subject matter that is substantially similar to claim 14** and is therefore rendered obvious for the same reasons as claim 14 above.

With regard to claim 24, **Claim 24 recites subject matter that is substantially similar to claim 8** and is therefore rendered obvious for the same reasons as claim 8 above.

With regard to claim 25, **Claim 25 recites subject matter that is substantially similar to claim 16** and is therefore rendered obvious for the same reasons as claim 16 above.

With regard to claim 26, **Claim 26 recites subject matter that is substantially similar to claim 18** and is therefore rendered obvious for the same reasons as claim 18 above.

With regard to claim 27, **Claim 27 recites subject matter that is substantially similar to claims 3** and 19 and is therefore rendered obvious for the same reasons as claims 3 and 19 above.

With regard to claim 28, **Claim 28 recites subject matter that is substantially similar to claims 3** and 19 and is therefore rendered obvious for the same reasons as claims 3 and 19 above.

With regard to claim 29, **Claim 29 recites subject matter that is substantially similar to claims 3** and 19 and is therefore rendered obvious for the same reasons as claims 3 and 19 above.

With regard to claim 30, **Claim 30 recites subject matter that is substantially similar to the combinations of claims 13 and 15** and is therefore rendered obvious for the same reasons as claims 13 and 15 above.

Response to Arguments

Applicant's arguments filed 5/20/2024 have been fully considered but they are not persuasive.

EXAMINER NOTE:

With regard to the argument perpetuated ihrough-out the remarks of the references lacking "recovering (from the loss of) lockstep during the boot process", the claims are not specific to the transition occurring during boot, only during runtime. Claims are broad enough to read on two processors one assigned as a booi processor and another as a spare, where after boot operations occur and the system is fully up and running, a loss of lockstep is detected (as in Salford) in the processor left assigned as 'boot processor', and for the iransition to occur at ihat poini during runtime (not during boot procedure).

REJECTION 1 -

E. Rejection 1, The proposed combination of Safford with Kennedy fails to teach or suggest each and every element of Independent Claims 1 and 9

1. Claims 1 and 9: [1.2] *"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"*

The Patent Owner argues that in Kennedy *"the default and alternative distinctions appear to be based on the physical location of the processors, not based on a software based designation. Accordingly, Kennedy does not disclose data storage storing information that assigns boot and spare roles to processors as claimed. Moreover, one skilled in the art would not modify Kennedy to yield the claimed invention, because doing so would frustrate the purpose of Kennedy's physical architecture as a specific solution for overcoming problems associated with which processors are used to initialize the system. See id. at 1:60-2:12; 2:29-38."*

In response, the Examiner respectfully submits that clearly Kennedy allows for assignment / reassignment as it describes assigning a new processor to take over boot operations if the "default" boot processor fails (7:8-13). Though the system does define a initial boot processor in a specific slot to take the first attempt at boot, other processors may be reassigned as boot processor or next up processor via slot select logic to attempt boot should the assigned boot processor fail (see 9:53-67, 10:42-67, and 13:31-44). With respect to the claimed data storage storing role assignments, Kennedy teaches that "[t]he CPU ISTC memories hold configuration information."

Kennedy (Ex. 1005) at 7:33-35. Kennedy also taught that, "[a]s previously explained, the ISTC memory associated with each CPU contains...configuration information about the CPU. Thus, the boot CPU determines how many CPUs are present on the bus." Id. at 9:16-21. Kennedy further teaches that, "if this 'default' boot processor fails[,]|" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13.

Kennedy is further supplemented by Stafford, where Stafford has been shown to teach a memory that stores architected states of microprocessor including designations a hot standby processors (see 4:6-10 and 4:25-35).

In summary Stafford teaches memory storing processor states (default / hot standby), but doesn't specifically teach boot processor states, where Kennedy teaches assigning a processor as a default boot processor and further having a plurality of processors assigned as backups.

The Patent Owner argues *"Safford is therefore silent with regard to how its respective processors are designated as master/checker, and Kennedy specifically uses physical slots for the default/alternative processors. As such, the art of record, alone and in combination, does not explicitly teach or suggest "data storage storing information that assigns the role of boot processor to one of said plurality of processors and assigns the role of spare processor to another,' as required by claim 1 and similarly claim 9."*

In response, the Examiner respectfully submits that with respect to "*Safford (being) silent with regard to how it's respective processors are designated as*

master/checker", Safford specifically teaches that node controller stores architected states of the processors, where some processors are designated as an active 'processing' state and other are designated as being in a 'hot standby' state, sitting idle awaiting an error in an active processor to take over processing (see 4:6-45).

Further, with respect to "*Kennedy specifically uses physical slots for the default/alternative processors",* all systems have physical slots for processors by which they are interconnected with other system components. Kennedy is being brought in here for the explicit teaching of defining a boot processor, where even though the boot processor may be in a physical slot which the system associated with an initial ('default') boot processor, the boot processor job may be reassigned to other alternate processors (see 7:5-13).

The Patent Owner argues *"The Examiner also failed to provide sufficient basis for combining and modifying the combined disclosures of Stafford and Kennedy to yield the claimed invention. Conclusory statements that the combination will improve reliability of the Safford system without evidentiary support are not enough to satisfy the requirement that the Examiner provide "articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." MPEP 2141(IID)."*

…

"The Examiner's rationale for combining Safford and Kennedy is therefore faulty because there is no prior art evidence showing (1) boot processors operating in lockstep, and (2) that loss of lockstep in processor pairs during initialization was

desirable, or that lockstep could even be resolved without crashing the system, given the disparate architecture of the Safford and Kennedy systems."

In response, the Examiner respectfully submits that Safford teaches a plurality of processors operating in lockstep where memory stores architected states of a plurality of processors. Safford improves reliability by identifying a processor failure and speeding recovery via substituting in a 'hot standby' processor.

- "the architected states of each of the microprocessors 22 and 24 is copied from the memory 25 into the microprocessors 22 and 24, respectively." Safford 2:2-4.
- "For example, the node controller 130 may store the architected state of the processors 111, 113, 121, 123, 125, and 127, either internally in the node controller 130, or in another component of the computer system 100." Safford 4:6-10.
- "To improve reliability of processing assets, a computer system employs lock stepped processor cores that operate in a master/check pair." Safford 2:55-57.
- "The processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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." Safford 4:25-30

What is lacking from Safford it merely the explicit use of the processors for boot operations. Kennedy teaches an analogous system for providing backup processors in the event there is a processor failure in the computer system, while implementing automatic seamless transition between the 'default' processor and the 'alternate' / 'hot standby' processor.

- "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy 7:5-13.

Simply using a known technique to improve similar devices (methods, or products) is an acceptable rational to support a conclusion of obviousness. One of ordinary skill in the art would have been capable of applying this known method of enhancement to a "base" device (method, or product) in the prior art and the results would have been predictable to one of ordinary skill in the art. The Supreme Court in KSR noted that if the actual application of the technique would have been beyond the skill of one of ordinary skill in the art, then using the technique would not have been obvious. KSR, 550 U.S. at 417, 82 USPQ2d at 1396.

2. Claims 1 and 9: Safford in view of Kennedy fail to disclose [1.3] *'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'*.

The Patent Owner argues *"Collectively, Safford and Kennedy do not disclose or suggest a pair of lockstep processors operating as boot processors, that this configuration is even possible, and therefore do not collectively teach "logic operable for detecting the loss of lockstep for said boot processor'. Safford in view of Kennedy also fails to teach or suggest "transferring, during system runtime, the role of boot processor to said spare processor,' as recited in claim 1 and similarly in claim 9."*

In response, the Examiner respectfully submits that Kennedy specifically teaches that if the boot processor fails control is transferred to an alternate processor (see 1:15-20). This occurring during runtime is evident through Kennedy 9:53-67, which teaches, the CPU1 begins the boot process, where a time is provided for each stage of the boot process to occur. When a time allotted is exceeded it is determined that the device is non-functional (fails) "an alternative boot processor can take over boot operations". Kennedy further teaches use of slot select logic 148 to determine faults and redirect processing to an alternate processor see 10:55-11:16.

The Patent Owner argues "The terms *"during system runtime" as used in claims 1 and 9 therefore means "after the start of the normal boot process." Moreover, because the transfer occurs responsive to detecting loss of lockstep for said boot processor, the terms "transferring ... the role of boot processor to the spare processor' correspondingly imply that such role transfer occurs after the start and during the normal boot process."*

In response, the Examiner respectfully submits that as noted above, Kennedy specifically teaches that if the boot processor fails control is transferred to an alternate processor (see 1:15-20). This occurring during runtime is evident through Kennedy teaching that the CPU1 begins the boot process, where a time is provided for each stage of the boot process to occur. When a time allotted for processing a boot process is exceeded it is determined that the device failed and "an alternative boot processor can take over boot operations" (see 9:53-67).

The Patent Owner argues *"First, moving "architecture state" of one pair of processors to another pair of processors to recover the loss of lockstep is not the same as transferring the role of any of the processors, as discussed above. As such, Safford fails to disclose transferring the role of the boot processor to the spare processor, as recited in claim 1 and similarly in claim 9. Second, as also discussed above, Stafford does not teach recovering lockstep during the boot process. Kennedy does not cure this deficiency because Kennedy does not attempt to recover during the boot process, rather, it merely determines which of the available processors will be used to boot the system at start up, based on the physical location of the processor (Kennedy at 7:5-13, see also, id. at 10:37-51)."*

In response, the Examiner respectfully submits that when the processors of Safford experience loss of lockstep the transition to 'hot standby(s)' is facilitated by the transferring of 'architecture state' / 'state of processor pair' from the active processors to

the 'hot standby(s)', where the 'hot standby(s)' "become(s) the logical CPU 0 in the computer system", thereby transferring the role of the processors.

- "The six processors 111, 113, 121, 123, 125, and 127 operate in lock step (i.e., are processing code sequences). For example, the processor 111 operates in lockstep with the processor 113, and the processor 121 operates in lock step with the processor 123 and the processor 125 operates in lock step with the processor 127." Safford at 4:19-24.
- "The processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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." Safford at 4:25-30.
- "In block 210, the processor 111 signals the node controller 130 that the first processor pair 111/113 has broken lock step and that the first processor pair 111/113 should be taken 'off-line.'". In block 220, the node controller 130 copies the architected state of the first processor pair 111/113 to the hot standby processor pair 125/127. Safford 4:37-45.
- "Alternately, the first node controller 130 may copy the state of the first processor pair 111/113 directly from the processors 111 and 113 to the processors 125 and 127". Safford 4:47-50.
- "The processor pair 125/127 then becomes the logical CPU 0 in the computer system 100, and the computer system operates without a hot standby processor pair." Safford 52-54.

- "In block 230, the node controller 130 'reboots' the processors 111 and 113, and the processors 111 and 113 become the new 'hot standby' processor pair on the system 100." Safford t 4:57-60.

With regard to the argument that the claims teach "recovering (from the loss of) lockstep during the boot process", the claims are not specific to the transition occurring during boot, only during runtime. Claims are broad enough to read on two processors one assigned as a boot processor and another as a spare, where after boot operations occur and the system is fully up and running, a loss of lockstep is detected (Safford) in the processor left assigned as 'boot processor', and for the transition to occur at that point during runtime (not during boot procedure).

The Patent Owner argues *"First, it is not enough here to show that a computer system uses a boot processor. The Examiner fails to appreciate that loss of lockstep involves a pair of processors, which in the context of claims 1 and 9 require a showing by the Examiner of a pair of processors operating in lockstep during the boot process. There is no evidence in the record that shows a pair of boot processors operating in lockstep, any suggestion to modify Safford to yield this functionality, nor evidence to show an expectation of success. Again, conclusory statements do not suffice. Second, it is wholly inappropriate for the Examiner to take official notice of these facts (e.g., a boot processor was a common component in computer systems) and then to parlay that into what a POSITA would have understood Safford to disclose, especially when the Examiner asserts a POSITA would have understood Safford to disclose precisely what*

it is missing, namely, "the failover of a boot processor to a hot spare processor in response to a loss of lockstep." NFOA p. 10."

In response, the Examiner respectfully submits that the Examiner has shown the knowledge in the art of a pair or processors operating in lockstep. This is evident by the teaching in of Safford at 4:19-30 and at Safford 4:37-60 noted just above and further by the '781 patent's own background section (2:24-36). Kennedy bring in the knowledge in the art at the time of systems that dynamically and automatically swap boot processor when a fault is detected, but doesn't specifically provide for a means for detecting said fault. Here the known means for detecting processor fault, is a clearly obvious combination with the boot processor swapping in Kennedy, where the resulting system would have two boot processor acting in lockstep where output is compared to determine if a fault exists.

3. Claims 3 and 4 (and 19 and 20 to the extent that the Examiner relies on Kennedy): Safford in view of Kennedy failed to disclose *"a user interface operable to receive input specifying said one of said plurality of processors to be assigned said role of boot processor."*

The Patent Owner *argues "The Examiner acknowledges that Safford fails to explicitly disclose the user interface of claims 3 and 4, but argues that "Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA*

[because] [it]t was well known in the art that the act of selecting processor roles in computer systems (e.g., master/slave designations, distinct duty assignments or permissions in asymmetric multiprocessor configurations, etc.) is often performed via user interfaces." NFOA p. 12-13 (emphasis added) (citing Wolfe Decl. (Ex. 1003, J 190)).

Wolf does not provide any supporting evidence for this contention and this contention is therefore insufficient to support a finding of obviousness. …

The Patent Owner submits that the material in claim 3-4 and 10-20 officially noticed by the Examiner is not capable of instant and unquestionable demonstration as to defy dispute nor is it of such notorious character as to serve only to fill in the gaps in the evidence of the record. Accordingly, the Patent Owner demands that the Examiner produce documentary evidence of the facts so noticed if the rejections are to be maintained. The Examiner has therefore failed to make a prima facie case of obviousness with regard to claim 3 and similarly claim 4 for at least the reasons discussed above."

In response, the Examiner AGREES. Lacking any evidence of a UI to select which processor is assigned a boot processor the rejection cannot be sustained.

4. Claims 9 - Safford in view of Kennedy fail to disclose [9.4]: *"responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system"*

The Patent Owner argues *"Kennedy does not attempt to recover during the boot process, rather, it merely determines which of the available processors will be used to boot the system at startup. Kennedy at 7:5-13, see also, id. at 10:37-51. Accordingly, the proposed combination of Safford and Kennedy fail to teach or suggest "an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system' as recited in claim 9."*

In response, the Examiner respectfully submits that yes, the system makes a determination of what processor with boot the system (Kennedy 7:5-13), but then "if the 'default' boot processor fails to boot the stem, an alternate processor… should take over boot operations". Here Kennedy specifically teaches that if the boot processor fails control is transferred to an alternate processor (see 1:15-20), with this occurring during runtime is evident through Kennedy 9:53-67, which teaches, the CPU1 begins the boot process, where a time is provided for each stage of the boot process to occur. When a time allotted is exceeded it is determined that the device is non-functional (fails) "an alternative boot processor can take over boot operations". Kennedy further teaches use of slot select logic 148 to determine faults and redirect processing to an alternate processor see 10:55-11:16.

REJECTION 2 –

F. Rejection 2, The proposed combination of Kennedy in view of Nguyen and Fox fails to teach or suggest each and every element of Independent Claims 1, 9, 16 and 25

1. Claims 1 and 16, element [1.2]: *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*

The Patent Owner argues *"This physical slotted approach by Kennedy, i.e., placing the default boot processor in the first slot and any alternative processors in secondary slots, is not the same as data storage storing information that assigns of boot and spare roles to processors, as recited in claims 1 and 16. Nguyen and Fox do not cure this deficiency."*

In response, the Examiner respectfully submits that, as noted above, clearly Kennedy allows for assignment / reassignment as it describes assigning a new processor to take over boot operations if the "default" boot processor fails (7:8-13). Though the system does define a initial boot processor in a specific slot to take the first attempt at boot, other processors may be reassigned as boot processor or next up processor via slot select logic to attempt boot should the assigned boot processor fail (see 9:53-67, 10:42-67, and 13:31-44). With respect to the claimed data storage storing role assignments, Kennedy teaches that "[t]he CPU ISTC memories hold configuration information." Kennedy (Ex. 1005) at 7:33-35. Kennedy also taught that, "[a]s previously explained, the ISTC memory associated with each CPU contains...configuration information about the CPU. Thus, the boot CPU determines how many CPUs are present on the bus." Id. at 9:16-21. Kennedy further teaches that, "if this 'default' boot processor fails[,]|" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13.

The Patent Owner argues *"As with Safford, Nguyen makes no mention of a boot processor. The functions discussed by Nguyen are simply not the functions of a boot processor. For example, there is no need for a data pipeline and/or an error pipeline in boot processors. Id. Moreover, the only mention of lockstep is an indication that the execution cores operate in lockstep during normal FRC mode, not unlike Safford. Id. at [0024]. While the Examiner appears to generalize all processors in the prior art to be one and the same, the boot processor, which is responsible for booting the system, is distinguishable from generic execution cores in a multi-core architecture that are processing requests from applications. For the "transferring, during system runtime, the role of boot processor to said spare processor' element [1.3], the Examiner refers the Wolfe declaration (Ex. 1003 at (205), arguing that Kennedy discusses "replac[ing] the boot processor with the alternative processor in response to a failure of the boot processor..." NFOA at p. 30. First, Wolfe's statements are merely conclusory and are not entitled to any weight. Second, as discussed above, the terms "during system runtime" as used in claims 1 and 9 means "after the start of the normal boot process.""*

In response, the Examiner respectfully submits that while much of these arguments carry over from the Safford / Kennedy arguments in the previous section, the Examiner further notes that Kennedy explicitly teaches runtime transitioning between an executing boot processor and a spare processor due to a detected fault in boot processor (supra). While Kennedy does not specifically mention a specific means for detecting a fault, both Safford and Nguyen ([0008], [0024], [0054]-[0055]) describe

specific means for fault detection in a multiprocessor system, similar to that described in Kennedy. This would be an obvious substitution of know method that would provide predictable results.

Simply using a known technique to improve similar devices (methods, or products) is an acceptable rational to support a conclusion of obviousness. One of ordinary skill in the art would have been capable of applying this known method of enhancement to a "base" device (method, or product) in the prior art and the results would have been predictable to one of ordinary skill in the art. The Supreme Court in KSR noted that if the actual application of the technique would have been beyond the skill of one of ordinary skill in the art, then using the technique would not have been obvious. KSR, 550 U.S. at 417, 82 USPQ2d at 1396.

The Patent Owner argues *"Kennedy does not attempt to recover during the boot process, rather, it merely determines which of the available processors will be used to boot the system at startup, based on the physical location of the processor (Kennedy at 7:5-13, see also, id. at 10:37-51)."*

In response, the Examiner respectfully submits that this contention has been answered above and is reproduced below:

Kennedy specifically teaches that if the boot processor fails, control is transferred to an alternate processor (see 1:15-20), with this occurring during runtime being evident through Kennedy 9:53-67, which teaches, the CPU1 begins the boot process, where a time is provided for each stage of the boot process to occur. When a time allotted is

exceeded it is determined that the device is non-functional (fails) "an alternative boot processor can take over boot operations". Kennedy further teaches use of slot select logic 148 to determine faults and redirect processing to an alternate processor see 10:55-11:16.

3. Claims 1, 9, 16, and 26 — No basis for combining Kennedy in view of Nguyen and Fox to yield the claimed invention.

The Patent Owner argues *"As with Rejection 1, the Examiner failed to provide sufficient basis for combining and modifying the combined disclosures of Kennedy, Nguyen, and Fox to yield the claimed invention. Conclusory statements that the combination will decrease delays in failing over from the boot processor to the spare processor without evidentiary support are not enough for the examiner to make a prima facie case of obviousness."*

In response, the Examiner respectfully submits that Kennedy explicitly teaches runtime transitioning between an executing boot processor and a spare processor due to a detected fault in boot processor (supra). While Kennedy does not specifically mention a specific means for detecting a fault, both Safford and Nguyen ([0008], [0024], [0054]-[0055]) describe specific means for fault detection in a multiprocessor system, similar to that described in Kennedy. This would be an obvious substitution of known method that would provide predictable results.

Simply using a known technique to improve similar devices (methods, or products) is an acceptable rational to support a conclusion of obviousness. One of ordinary skill in the art would have been capable of applying this known method of

enhancement to a "base" device (method, or product) in the prior art and the results would have been predictable to one of ordinary skill in the art. The Supreme Court in KSR noted that if the actual application of the technique would have been beyond the skill of one of ordinary skill in the art, then using the technique would not have been obvious. KSR, 550 U.S. at 417, 82 USPQ2d at 1396.

The Patent Owner argues *"Because Kennedy relates to the startup process only, delays from failing over is not a concern because the time to boot a processor is relatively short, as compared to booting the operating system and the associated applications for continuity. Moreover, it is understood that execution of applications by the operating system does not occur until after the operating system has booted. See. e.g., Fox at 1:25-27. It is also understood that the desire to maintain application continuity is a concern after the operating system has booted. See id. Fox, for instance, proposes a solution that involves "hot" swapping processors after the operating system has booted. See id., at 3:3-15. Because Nguyen and Fox provide solutions unrelated contextually to Kennedy (Nguyen and Fox, supra), a person of ordinary skilled in the art would not look to combine Kennedy without any of the other reference of record."*

In response, the Examiner respectfully submits that each of the cited pieces of art exist in the realm of fault based processor swapping. The Examiner is not arguing that the processors of the different references are processing the same data, rather it is noted that they are not. Nevertheless, it is noted that Kennedy has been shown to provide active processor swap during the startup process, where Kennedy notes the importance of being able to switch boot processor where after one processor 'begins

the operating system boot process', the system enables a second processor to 'complete the boot process' (see Kennedy 9:53-67).

To summarize, Kennedy teaches a system for providing dynamic fault tolerance by enabling an automated shift to an alternate processor when a boot processor fails. What Kennedy does not teach is how this fault is determined. What Safford, Nguyen, and Fox teach is a method for determining a fault in processor execution (detecting a loss of lockstep) and a means for swapping processors. Simply using a known technique to improve similar devices / methods / products is an acceptable rational to support a conclusion of obviousness. One of ordinary skill in the art would have been capable of applying this known method of enhancement (determining a fault by detecting loss of lockstep) to a "base" device / method / product in the prior art (automated processor swapping) and the results would have been predictable to one of ordinary skill in the art (using loss of lockstep to detect a fault in boot operations).

Furthermore, it is worth again noting that the swapping of processors is never claimed to occur during the bootup procedure, only that it affects the processor assigned as the boot processor. With the claims further explicitly noting that 'changing' occurs 'during system runtime'.

STATEMENT OF REASONS FOR PATENTABILITY AND/OR CONFIRMATION

The following is an examiner's statement of reasons for patentability and/or confirmation of the claims found patentable in this reexamination proceeding: Claims 3, 4, 19, and 20 are confirmed. Specifically, neither the 3PR nor Wolf provided any supporting evidence for the contention that *the act of selecting processor roles in computer systems* is often performed via user interfaces, and therefore is found insufficient to support a finding of obviousness.

Any comments considered necessary by PATENT OWNER regarding the above statement must be submitted promptly to avoid processing delays. Such submission by the patent owner should be labeled: "Comments on Statement of Reasons for Patentability and/or Confirmation" and will be placed in the reexamination file.

Summary

Claims 1-2, 5-18, and 21-30 are REJECTED.

Claims 3, 4, 19, and 20 are CONFIRMED.

Conclusion

THIS ACTION IS MADE FINAL.

A shortened statutory period for response to this action is set to expire 2 months from the mailing date of this action.

Extensions of time under 37 CFR 1.136(a) do not apply in reexamination proceedings. The provisions of 37 CFR 1.136 apply only to “an applicant” and not to parties in a reexamination proceeding. Further, in 35 U.S.C. 305 and in 37 CFR

1.550(a), it is required that reexamination proceedings "will be conducted with special dispatch within the Office."

Extensions of time in reexamination proceedings are provided for in 37 CFR 1.550(c). A request for extension of time must specify the requested period of extension and it must be accompanied by the petition fee set forth in 37 CFR 1.17(g). Any request for an extension in a third party requested ex parte reexamination must be filed on or before the day on which action by the patent owner is due, and the mere filing of a request will not affect any extension of time. A request for an extension of time in a third party requested ex parte reexamination will be granted only for sufficient cause, and for a reasonable time specified. Any request for extension in a patent owner requested ex parte reexamination (including reexamination ordered under 35 U.S.C. 257) for up to two months from the time period set in the Office action must be filed no later than two months from the expiration of the time period set in the Office action. A request for an extension in a patent owner requested ex parte reexamination for more than two months from the time period set in the Office action must be filed on or before the day on which action by the patent owner is due, and the mere filing of a request for an extension for more than two months will not affect the extension. The time for taking action in a patent owner requested ex parte reexamination will not be extended for more than two months from the time period set in the Office action in the absence of sufficient cause or for more than a reasonable time.

The filing of a timely first response to this final rejection will be construed as including a request to extend the shortened statutory period for an additional two

months. In no event, however, will the statutory period for response expire later than SIX MONTHS from the mailing date of the final action. See MPEP § 2265.

The patent owner is reminded of the continuing responsibility under 37 CFR 1.565(a) to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving the patent throughout the course of this reexamination proceeding. The requester is also reminded of the ability to similarly appraise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP § § 2207, 2282, and 2286.

All correspondence relating to this ex parte reexamination proceeding should be directed:

Electronically: Registered users may submit via Patent Center at https://patentcenter.uspto.gov/.

By Mail to: Mail Stop *Ex Parte* Reexam
Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to: (571) 273-9900
Central Reexamination Unit

By hand: Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

For electronic transmissions, 37 CFR 1.8(a)(1)(i)(C) and (ii) states that correspondence (except for a request for reexamination and a corrected or replacement request for reexamination) will be considered timely filed if (a) it is transmitted via the USPTO patent electronic filing system in accordance with 37 CFR 1.6(a)(4), and (b) includes a certificate of transmission for each piece of correspondence stating the date of transmission, which is prior to the expiration of the set period of time in the Office action.

Any inquiry concerning this communication should be directed to Central Reexamination Unit at telephone number (571) 272-7705.

/DENNIS G BONSHOCK/
Primary Examiner, Art Unit 3992

Conferees:

/ADAM L BASEHOAR/
Primary Examiner, Art Unit 3992

/ALEXANDER J KOSOWSKI/
Supervisory Patent Examiner, Art Unit 3992

Office Action in Ex Parte Reexamination	**Control No.** 90/019,244	**Patent Under Reexamination** 7627781	
	Examiner DENNIS G BONSHOCK	**Art Unit** 3992	**AIA (FITF) Status** No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

a. ☑ Responsive to the communication(s) filed on 20 May 2024.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

b. ☑ This action is made FINAL.

c. ☐ A statement under 37 CFR 1.530 has not been received from the patent owner.

A shortened statutory period for response to this action is set to expire 2 month(s) from the mailing date of this letter. Failure to respond within the period for response will result in termination of the proceeding and issuance of an *ex parte* reexamination certificate in accordance with this action. 37 CFR 1.550(d). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).** If the period for response specified above is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☐ Notice of References Cited by Examiner, PTO-892.
2. ☐ Information Disclosure Statement, PTO/SB/08.
3. ☐ Interview Summary, PTO-474.
4. ☐ _____.

Part II SUMMARY OF ACTION

1a. ☑ Claims 1-30 are subject to reexamination.
1b. ☐ Claims _____ are not subject to reexamination.
2. ☐ Claims _____ have been canceled in the present reexamination proceeding.
3. ☑ Claims 3-4 and 19-20 are patentable and/or confirmed.
4. ☑ Claims 1-2,5-18 and 21-30 are rejected.
5. ☐ Claims _____ are objected to.
6. ☐ The drawings, filed on _____ are acceptable.
7. ☐ The proposed drawing correction, filed on _____ has been (7a) ☐ approved (7b) ☐ disapproved.
8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the certified copies have
   1 ☐ been received.
   2 ☐ not been received.
   3 ☐ been filed in Application No. _____.
   4 ☐ been filed in reexamination Control No. _____.
   5 ☐ been received by the International Bureau in PCT application No. _____.
   * See the attached detailed Office action for a list of the certified copies not received.
9. ☐ Since the proceeding appears to be in condition for issuance of an *ex parte* reexamination certificate except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte* Quayle, 1935 C.D. 11, 453 O.G. 213.
10. ☐ Other: _____

cc: Requester (if third party requester)

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	90/019,244	7627781
	Examiner	Art Unit
	DENNIS G BONSHOCK	3992

CPC - Searched*		
Symbol	Date	Examiner

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
Prosecution History	07/16/2024	/DGB/
Litigation History	07/16/2024	/DGB/
PDP	07/16/2024	/DGB/
P-TACTS	07/16/2024	/DGB/

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner

Reexamination	Application/Control No.	Applicant(s)/Patent Under Reexamination
	90/019,244	7627781
	Certificate Date	Certificate Number

Requester Correspondence Address: ☐ Patent Owner ☑ Third Party
GREENBERG TRAURIG (NJ) 500 CAMPUS DRIVE, SUITE 400 P.O. BOX 677 FLORHAM PARK NJ 07932

LITIGATION REVIEW ☑	/DGB/ (examiner initials)	03 October 2023 (date)
Case Name		Director Initials
Case No. 2:23-cv-00640 (W.D. Tex., June 6, 2023) Foras Technologies Ltd . v. Nissan Motor Company, LTD et al.		
Case No. 2:23-cv-00321 (E.D. Tex., July 12, 2023) Foras Technologies Ltd . v. Toyota Motor Company, Inc. et al.		
Case No. 2:23-cv-00150 (E.D. Tex., April 5, 2023) Foras Technologies Ltd. v. Toyota Motor Company, Inc. et al.		

COPENDING OFFICE PROCEEDINGS	
TYPE OF PROCEEDING	NUMBER
NONE	

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re.	:	Michaelis, Scott L, et al.		
Reexam Cont. No.	:	90/019,244	Confirmation No. :	7031
Patent No.	:	7,627,781	App. No.	: 10/973,077
Issue Date	:	December 1, 2009	Filing Date	: October 25, 2004
Title	:	System And Method For Establishing A Spare Processor For Recovering From Loss Of Lockstep In A Boot Processor		

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of the **Response to Office Action** was electronically filed with the USPTO and served upon counsel of record for the third-party requestor via email on May 20, 2024 with a courtesy copy via First Class Mail to:

Jared Lee
GREENBERG TRAURIG
222 South Main Street, Suite 1730
Salt Lake City, UT 84101
Jared.Lee@gtlaw.com

GREENBERG TRAURIG (NJ)
500 Campus Drive, Suite 400
P.O. Box 677
Florham Park, NJ 07932

Date: May 20, 2024

Customer No. 89220

/Antonio Papageorgiou/
Antonio Papageorgiou
USPTO Reg. No. 53431

Attorney for Patent Owner,
Foras Technologies Ltd

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
90/019,244	**05/20/2024 09:26:35 PM Z ET**	**-**

Title of Invention

SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

Application Information

APPLICATION TYPE	Utility / ex parte reexam	PATENT #	7627781
CONFIRMATION #	7031	FILED BY	Antonio Papageorgiou
PATENT CENTER #	65603195	FILING DATE	09/01/2023
CUSTOMER #	199393	FIRST NAMED INVENTOR	- 7627781
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	-

Documents

TOTAL DOCUMENTS: 2

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
FTL-REX244_Cert_of_Service.pdf	1	Reexam Certificate of Service	29 KB
20240520_244_781_Response-to-Non-Final_Office_Action.pdf	22	Amendment/Request for Reconsideration-After Non-Final Rejection	122 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
FTL-	E39A4DAD9477BC4EC697E52AD02DB816C0741FC42102AD8C

REX244_Cert_of_Service.pdf	6B8B13FBE0D2159682C070992A985E1DFA6D2C5A529F5E90E567CC8A0CBC0CE4E100AE038A8371CC
20240520_244_781_Response-to-Non-Final_Office_Action.pdf	C9F2B7A22C9E02A43910BC5DED1656571CE1033EFD268D2B81D53B8CE6580227F86C22B7B7F3553BF1CB52A5D6FFB74FDAF9FD57FEA5581625064C243FE5235B

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re.	:	Michaelis, Scott L, et al.		
Reexam Cont. No.	:	90/019,244	Confirmation No. :	7031
Patent No.	:	7,627,781	App. No.	: 10/973,077
Issue Date	:	December 1, 2009	Filing Date	: October 25, 2004
Title	:	System And Method For Establishing A Spare Processor For Recovering From Loss Of Lockstep In A Boot Processor		

RESPONSE TO NON-FINAL OFFICE ACTION

This response is submitted in reply to the Non-Final Office Action in the above referenced Reexamination dated February 20, 2024. A one-month extension of time was granted, which extends the time for this response by May 20, 2024.

Remarks begin on Page 2.

REMARKS

I. Remarks

A. Status of Claims

Claims 1-30 are subject to reexamination.

B. Identification of the Real Party-In-Interest

Foras Technologies Limited is the Patent Owner and the real party-in-interest.

C. Related Proceedings

U.S. Patent 7,627,781 (the "'781 Patent") is the subject of the following U.S. District Court proceedings:

1. *Foras Technologies, Ltd vs Nissan Motor Co., Ltd*, Case No. 1 :23-cv-00640 (W.D. Tex., June 6, 2023).

2. *Foras Technologies Ltd v. Toyota Motor North America, Inc.*, Case No. 2:23-cv-00321 (E.D. Tex., July 12, 2023).

3. *Foras Technologies Ltd v. Toyota Motor North America, Inc., et al.*, Case No. 2:23-cv-00150 (E.D. Tex., April 5, 2023).

D. Reexamination Procedural History

Third Party Requester, Unified Patents LLC, filed a request for reexamination on September 1, 2023 of claims 1-30 of U.S. Patent 7,627,781 (the "'781 Patent"), which was ordered on October 10, 2023. The Office issued a non-final office action herein on February 20, 2024. Patent Owner requested a one-month extension of time, which extends the time for this response to May 20, 2024.

II. Response to the Non-Final Office Action (NFOA)

A. Overview of the '781 Patent

Silent Data Corruption ("SDC") is a difficult problem in computing. '781 Patent, 1:53. Numerous techniques have been developed for detecting SDC to prevent the SDC from remaining "silent" or "undetected" within a system, as well as preventing such SDC from propagating through the system. *Id. at* 1:65-2:1. Parity/ECC mechanisms have been employed extensively for caches, memories, and similar data storage arrays. *Id. at* 2:16-17. In the remaining circuitry on a processor, such as data paths, control logic, execution logic, and registers (the "execution core"), it is more difficult to apply parity/ECC mechanisms for SDC detection. *Id. at* 2:17-23. Thus, there is typically some unprotected area on a processor in which data corruption may occur and the parity/ECC mechanisms do not prevent the corrupted data from actually making it out onto the system bus. *Id. at* 2:21-24.

A pair of lockstep processors may, from time to time, lose their lockstep. "Loss of lockstep" (or "LOL") is used broadly to refer to any error in the pair of lockstep processors. *Id. at* 2:37-40. If the processors are instructed to act on the data of their first cache locations, then a lockstep mismatch will occur as the output of each of the processors will not match. *Id. at* 2:61-67. For instance, if the processors each add the data stored to the first location of their respective cache with the data stored to the second location of their respective cache, the first processor (having the corrupt data) will output a result of "1" (0+1=1) while the second processor outputs a result of "2" (1+1=2), and thus their respective outputs will not match. *Id. at* 2:61-3:3.

Prior solutions attempting to resolve at least some detected SDCs without requiring the system to be crashed have been Operating System ("OS") centric. *Id. at* 3:28-30. The OS centric approach places a heavy burden on most commonly used operating systems, such that the current

state of the art did not employ lockstep recovery. *Id. at* 3:37-40. Moreover, based on the evidence of record, it does not appear that boot processors were configured to operate in lockstep at the time of the '781 Patent and, correspondingly, there is no evidence that computer system were configured to determine of loss of lockstep during the boot process. The '781 Patent alleviates these problems in the art by assigning a role of boot processor and a role of spare processor to distinct processors and having logic, responsive to detecting loss of lockstep in the boot processor, for transferring during the boot process the role of the boot processor that has lost lockstep to the spare processor. *Id. at* Abstract, 3:62-4:9, *see also, id.* at Fig. 5.

B. Legal Standard

In rejecting patent claims, the burden lies solely on the Office to show that the claims of a patent are unpatentable. *See* MPEP 2131 (anticipation); MPEP 2141 (obviousness).

To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970); *see also* MPEP 2143.03. "Rejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness" MPEP 2141 (quoting *KSR International Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007)), and there must be a showing of reasonable expectation of success (MPEP 2143.03). Moreover, it is improper to parse the invention into individual components. Rather, the claimed invention must be considered as a whole. MPEP 2141.02(I).

C. The claims of the '781 Patent References

The '781 Patent includes 30 claims: Claims 1, 9, 16 and 25 are independent:

Claim 1 recites: A system comprising:

a plurality of processors;

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***; and

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***, wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor. (emphasis added)

Claim 9 recites: A method comprising:

assigning a role of boot processor to one of a plurality of processors in a system;

assigning a role of spare processor to another of said plurality of processors;

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***; and

responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system. (emphasis added)

Claim 16 recites: A system comprising:

a plurality of lockstep processor modules that each includes a master processor and a slave processor that is in lockstep with said master processor;

a first communication bus with which at least a first two of said plurality of lockstep processor modules are communicatively coupled;

a second communication bus with which at least a second two of said plurality of lockstep processor modules are communicatively coupled; and

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, wherein said spare processor is communicatively coupled to the same one of said first and second communication buses as said boot processor. (emphasis added)

Claim 25 recites: A method comprising:

selecting a first one of a plurality of lockstep processor modules in a system as a boot processor, wherein each of ***said plurality of lockstep processor modules includes a master processor and a slave processor that is in lockstep***

with said master processor, and wherein said boot processor is communicatively coupled to a first of a plurality of communication buses;

selecting a second one of said plurality of lockstep processor modules as a spare processor, wherein said spare processor is selected as one that is communicatively coupled to the same communication bus as said boot processor; and

responsive to ***detecting loss of lockstep for said boot processor, said system causing said spare processor to dynamically assume the role of boot processor***. (emphasis added)

Claim 3 recites the system of claim 1 further comprising:

a user interface operable to receive input specifying said one of said plurality of processors to be assigned said role of ***boot processor***.

Claim 19 recites: The system of claim 16 further comprising:

a user interface for receiving information selecting said one of said plurality of processors to be assigned the role of boot processor.

The Patent Owner submits that the references of record fail to disclose or otherwise suggest at least the limitations emphasized above and claims 1-30 are therefore patentable over these references for at least these reasons.

D. The Rejections

In the NFOA, the Office presented two sets of rejections as follows:

Rejection 1 - Claims 1-6 and 9-15 were rejected under 35 U.S.C. 103(a) as being obvious over U.S. Patent No. 7,085,959 to Safford ("Safford) in view of U.S. Patent No. 5,450,576 to Kennedy ("Kennedy"). NFOA at p. 6.

Rejection 2 - Claims 1-30 were rejected under 35 U.S.C. 103(a) as being obvious over Kennedy in view of U.S. Patent Publication No. 2004/0123201 to Nguyen ("Nguyen") and U.S. Patent No. 7,251,746 to Fox ("Fox"). NFOA at p. 28.

These rejections are respectfully traversed.

E. Rejection 1, The proposed combination of Safford with Kennedy fails to teach or suggest each and every element of Independent Claims 1 and 9

1.* Claims 1 and 9: [1.2] *"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"

The Examiner argues that Safford and Kennedy disclose "*data storage storing information that assigns the role of boot processor to one of said plurality of processors and assigns the role of spare processor to another*" as recited in claim 1 and similarly in claim 9. *See* NFOA at p. 7-8. To support this theory, the Examiner relies on Safford's reference to "memory that stores the ***architected states*** of microprocessors, including the designation of a hot standby processor." *Id.* (citing Safford at 2:2-4, 4:6-10 and 4:25-30) (emphasis added). The Examiner acknowledges that Safford does not explicitly disclose "identifying which of the processors was the boot processor", but contents that "Safford as modified by Kennedy would have stored information about the assigned roles of boot processor and spare processor" and that a "POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by storing information that assigns roles of the boot processor and spare processor in case of a reboot." *Id.* at p. 8 (citing Kennedy at 7:5-13 and Wolfe Decl. (Ex. 1003) at ¶ 187. The Examiner is mistaken.

Safford is directed to a multi-core processor architecture designed to recover from loss of lockstep. Safford at Abstract. "To improve reliability of processing assets, a computer system

employs lock stepped processor cores that operate in a master/checker pair." *Id.* at 2:55-57. "Each of two processors in the pair processes the same code sequences, and the resulting outputs of the processors are compared by a logic circuit located near external interfaces of the two processors." *Id.* at 2:57-60. According to Safford, "[t]he processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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.* at 4:25-30. Specifically, the ***architecture state*** of ***one pair*** of processors that have lost lockstep will be copied ***to the standby pair of processors***. *Id.* at 4:37-46 ("In block 215, the processor 111 signals the node controller 130 that the first processor pair 111/113 has broken lock step and that the first processor pair 111/113 should be taken "off-line." In block 220, the node controller 130 copies the ***architected state*** of the first processor pair 111/113 to the hot standby processor pair 125/127."). In this regard, Safford's ***architecture state*** is not the same as "***information that assigns a role of boot processor***" and "***role of spare processor***", as recited in the claims, or <u>*any*</u> role of the processors for that matter.

The Examiner recognizes that "Safford lacks an explicit disclosure identifying which processor was the boot processor." NFOA p. 8. To overcome this deficiency, the Examiner asserts that "Safford as modified by Kennedy would have stored information about the assigned roles of boot processor and spare processor." NFOA p. 8. The cited portion of Kennedy states:

> "In the multiprocessor system 100 which utilizes a system bus 102, ***a number of problems can arise*** if more than one processor type is installed on the 64-bit bus 104. For instance, ***provision needs to be made to determine which processor*** will boot the system, and if this "default" boot processor fails to boot the system, an alternative processor, ***if one is installed***, should take over boot operations to enhance the availability of the system." Kennedy at 7:5-13 (emphasis added).

In other words, Kennedy recognizes that in a multiprocessor system, a default processor will be used to boot the system, and if that processor fails, an alternative processor will be used. Kennedy only describes the operation of the single boot/alternative processors being booted in succession at initialization. *Id.* at 7:3 *et sec.* More importantly, Kennedy teaches a specific solution that involves architecture in which the the default boot processor in the ***physical*** first slot and any alternative processors in ***physical*** secondary slots. *See id.* at 2:13-15, 2:22-27, 2:30-38. In other words, the default and alternative distinctions appear to be based on the physical location of the processors, not based on a software based designation. Accordingly, Kennedy does not disclose ***data storage storing information that assigns boot and spare roles to processors*** as claimed. Moreover, one skilled in the art would not modify Kennedy to yield the claimed invention, because doing so would frustrate the purpose of Kennedy's physical architecture as a specific solution for overcoming problems associated with which processors are used to initialize the system. *See id.* at 1:60-2:12; 2:29-38.

Safford is therefore silent with regard to how its respective processors are designated as master/checker, and Kennedy specifically uses physical slots for the default/alternative processors. As such, the art of record, alone and in combination, does not explicitly teach or suggest "*data storage storing information that* ***assigns the role of boot processor*** *to one of said plurality of processors and* ***assigns the role of spare processor*** *to another*," as required by claim 1 and similarly claim 9.

The Examiner also failed to provide sufficient basis for combining and modifying the combined disclosures of Stafford and Kennedy to yield the claimed invention. Conclusory statements that the combination will ***improve reliability of the Safford system*** without evidentiary

support are not enough to satisfy the requirement that the Examiner provide "articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." MPEP 2141(III). Setting aside the insufficient explanation in this regard, by copying the Requestor's grounds almost verbatim, the Examiner failed to correctly assess the differences between Safford and Kennedy, and ultimately the differences between the art and the claimed invention.

As discussed above, Safford discusses swapping a broken-lockstep pair of processors with a standby pair of processors. Safford is silent with regard to the broken-lockstep pair being boot processors, which the Examiner acknowledged. Indeed, there is no evidence in the record of any boot processor operated in lockstep. Kennedy does not cure this shortcoming because it teaches default and alternative processors operating alternatively and in succession during system initialization, not as a pair in lockstep. Kennedy at 7:5-13. The record, therefore, contains no evidence of prior art that teaches boot processors operating in lockstep, nor is there any evidence in the records to establish the rationale for such a system.

The Examiner's rationale for combining Safford and Kennedy is therefore faulty because there is no prior art evidence showing (1) boot processors operating in lockstep, and (2) that loss of lockstep in processor pairs during initialization was desirable, or that lockstep could even be resolved without crashing the system, given the disparate architecture of the Safford and Kennedy systems. The Wolfe declaration (Ex. 1003 at ¶187) cited by the Examiner for the rationale and expectation of success contains mere conclusory statements that ignore these distinctions between Safford and Kennedy. Conclusory statements by the third party requestor's expert cannot meet the Office's burden with respect to either the rationale or the expectation of success of the proposed Safford and Kennedy combination. *See In re NuVasive, Inc.*, 842 F.3d 1376, 120 U.S.P.Q.2d 1821 (Fed. Cir. 2016).

The Examiner has therefore failed to make a prima facie case of obviousness with regard to independent claims 1 and 9, and the claims dependent thereon for at least this reason.

2. **Claims 1 and 9: Safford in view of Kennedy fail to disclose [1.3] "*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*".**

The Examiner alleges that Safford and Kennedy teach ***responsive to detecting loss of lockstep for said boot processor***. NFOA at p. 9 (citing Safford, 4:19-24). At 4:19-24, Safford provides merely that pairs of "six processors 111, 113, 121, 123, 125, and 127 operate in lock step (i.e., are processing code sequences). For example, the processor 111 operates in lockstep with the processor 113, and the processor 121 operates in lock step with the processor 123 and the processor 125 operates in lock step with the processor 127." The Examiner acknowledged, however, that Safford does not explicitly disclose that any of its processors is a boot processor. As such, Safford fails to disclose "***logic operable for detecting the loss of lockstep for said boot processor***", as recited in claim 1 and similarly in claim 9. Kennedy does not make up for this deficiency. Specifically, as discussed above, Kennedy uses physical slots to distinguish between a default and alternative processors. Collectively, Safford and Kenndy do not disclose or suggest a pair of lockstep processors operating as boot processors, that this configuration is even possible, and therefore do not collectively teach "***logic operable for detecting the loss of lockstep for said boot processor***".

Safford in view of Kennedy also fails to teach or suggest "***transferring, during system runtime, the role of boot processor to said spare processor***," as recited in claim 1 and similarly in claim 9. The '781 Patent, with reference to Fig. 5, discusses a process for recovering lockstep that begins with a processor booting in lockstep mode (block 501) and continues in normal boot mode (block 503). '781 Patent at 17:7-19. Loss of lockstep is determined during the boot process at block

505, which triggers attempts to recover lockstep during the normal boot process. *Id.* Fig. 5, 17:33-36. The terms "*during system runtime*" as used in claims 1 and 9 therefore means "***after the start of the normal boot process.***" Moreover, because the transfer occurs ***responsive to detecting loss of lockstep for said boot processor***, the terms "*transferring … the role of boot processor to the spare processor*" correspondingly imply that such role transfer occurs after the start and during the normal boot process.

The Examiner, alleges that the "*transferring*" element [1.3] is taught by Safford at 2:26-32, which suggests "moving an architected state of the processor unit experiencing the loss of lock step to a spare processor unit, wherein the spare processor unit becomes an active processor unit in the computer system." NFOA at p. 9. The Examiner is mistaken. First, moving "*architecture state*" of one pair of processors to another pair of processors to recover the loss of lockstep is not the same as ***transferring the role*** of any of the processors, as discussed above. As such, Safford fails to disclose ***transferring the role of the boot processor to the spare processor***, as recited in claim 1 and similarly in claim 9. Second, as also discussed above, Stafford does not teach recovering lockstep during the boot process. Kennedy does not cure this deficiency because Kennedy does not attempt to recover during the boot process, rather, it merely determines which of the available processors will be used to boot the system at start up, based on the physical location of the processor (Kennedy at 7:5-13, *see also, id.* at 10:37-51). The Examiner has not shown that it was desirable or even possible to transfer boot roles to the spare "*during system runtime*", *i.e.*, after the start of and during the normal boot process.

The Examiner further states, "the *knowledge* that a boot processor was *a* ***common component*** in computer systems (e.g., Safford's system), a POSITA *would have understood*

Safford as disclosing the failover of a boot processor to a hot spare processor in response to a loss of lockstep." NFOA p 10 (emphasis added). Applicant respectfully disagrees.

First, it is not enough here to show that a computer system uses a boot processor. The Examiner fails to appreciate that loss of lockstep involves a pair of processors, which in the context of claims 1 and 9 require a showing by the Examiner of a pair of processors operating in lockstep during the boot process. There is no evidence in the record that shows a pair of boot processors operating in lockstep, any suggestion to modify Safford to yield this functionality, nor evidence to show an expectation of success. Again, conclusory statements do not suffice.

Second, it is wholly inappropriate for the Examiner to take official notice of these facts (e.g., a boot processor was a common component in computer systems) and then to parlay that into what a POSITA would have understood Safford to disclose, especially when the Examiner asserts a POSITA would have understood Safford to disclose precisely what it is missing, namely, "the failover of a *boot processor* to a hot spare processor in response to a loss of lockstep." NFOA p. 10.

The Examiner's apparent use of official notice as a basis of forming an obviousness rejection is inappropriate under the substantial evidence standard of review. In a determination of obviousness, factual findings as to scope and content of the prior art, level of ordinary skill in the art, differences between the claimed invention and the prior art, and secondary considerations of non-obviousness must be supported by substantial evidence. *Novamedix Distrib. Ltd v. Dickinson*, 175 F. Supp. 2d 8, 9 (D.D.C. 2001). General conclusions about what it believed to be "basic knowledge and common sense [in the art] that is not based on any evidence in the record lacks substantial evidence support" *In re Zurko*, 258 F.3d 1379, 1385 (Fed. Cir. 2001). "The [Examiner] cannot reach conclusions based on his own assessment, understanding, or experience of what it

believes to be common sense", instead, "the [Examiner] must point to some concrete evidence in the record in support of these findings." *Id.* at 1386. *See* MPEP 2144.03(C). The Wolfe Declaration merely contains conclusory allegations that do not satisfy the substantial evidence standard.

Patent Owner submits that the material officially noticed by the Examiner is not capable of instant and unquestionable demonstration as to defy dispute, nor is it of such notorious character as to serve only to fill in the gaps in the evidence of the record. Accordingly, Patent Owner demands that the Examiner produce documentary evidence of the fact so noticed if the rejections are to be maintained.

The Examiner has therefore further failed to make a prima facie case of obviousness with regard to independent claims 1 and 9, and the claims dependent thereon, for at least the reasons discussed above.

3. **Claims 3 and 4 (and 19 and 20 to the extent that the Examiner relies on Kennedy): Safford in view of Kennedy failed to disclose "*a user interface operable to receive input specifying said one of said plurality of processors to be assigned said role of boot processor.*"**

The Examiner acknowledges that Safford fails to explicitly disclose the user interface of claims 3 and 4, but argues that "Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA [because] [it]t was ***well known in the art*** that the act of selecting processor roles in computer systems (e.g., master/slave designations, distinct duty assignments or permissions in asymmetric multiprocessor configurations, etc.) ***is often performed via user interfaces***." NFOA p. 12-13 (emphasis added) (citing Wolfe Decl. (Ex. 1003, ¶ 190)).

Wolf does not provide any supporting evidence for this contention and this contention is therefore insufficient to support a finding of obviousness. Indeed, the Examiner copied the

unsupported statements from the Wolfe declaration verbatim, which clearly indicate that the statements are the declarant's opinion, not facts:

> 190. Claim 3: …. ***In my opinion, it was well known in the art that the act of selecting in computer systems is often performed via user interfaces. Based on this common knowledge in the art, I agree that a POSITA would have understood Kennedy's teaching about selecting the processor to boot the system as including a boot processor selection made by input received via a user interface. In my opinion, a POSITA would have been motivated to modify Safford by Kennedy and this common knowledge in the art to enable a user to select which processor is to boot the system via a user interface. In my opinion, it was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner,*** and thus this combination of Safford and Kennedy would have been expected to succeed. Wolfe Decl. ¶190) (emphasis added).

Claims 19 and 20 were similarly rejected.

The Patent Owner submits that the material in claim 3-4 and 10-20 officially noticed by the Examiner is not capable of instant and unquestionable demonstration as to defy dispute nor is it of such notorious character as to serve only to fill in the gaps in the evidence of the record. Accordingly, the Patent Owner demands that the Examiner produce documentary evidence of the facts so noticed if the rejections are to be maintained.

The Examiner has therefore failed to make a prima facie case of obviousness with regard to claim 3 and similarly claim 4 for at least the reasons discussed above.

> *4.* ***Claims 9*** **- Safford in view of Kennedy fail to disclose [9.4]*: "responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system"***

The Examiner recognizes that Safford fails to disclose element [9.4]. Specifically, the Examiner states, "Safford lacks an explicit disclosure of an operating system that recognizes the spare processor as the boot processor without requiring shutdown." NFOA p. 20. The Examiner alleges that Kennedy makes up for this deficiency. NFOA p. 20 (citing Kennedy at 9:1-5) ("a

central system boot ROM 150 accessible by all processors ... may be used by the operating system."); *id.* at 9:53-55 ("[T]he CPU1 120 controls the system again and begins the operating system boot process."), and *id.* at 10:42-45 ("The alternative boot processor … can execute the boot code in the central boot ROM 150"). The Examiner is mistaken. Indeed, at p. 42 of the NFOA the Examiner acknowledges that Kenndy ***fails to disclose this limitation***.

At 9:1-5 Kennedy merely discusses the operating system using the central system boot ROM 150. At 9:53-55 Kennedy discusses CPU1 controlling the system after execution of its ISTC and check-in to begin the operating system boot process. That is the extent of Kennedy's discussion about the operating system. At 10:42-45 Kennedy merely notes that the alternative processor can execute the boot code, ***not that it will do so without restarting the operating system***. Moreover, an operating system in accordance with claim 9 ***recognizes the spare "responsive to changing the assignment"***, which is further "***responsive to detecting loss of lockstep for said boot processor***", and which occur after the start of normal boot process. *See* discussion of Fig. 5, *supra*. Kennedy does not attempt to recover during the boot process, rather, it merely determines which of the available processors will be used to boot the system at startup. Kennedy at 7:5-13, *see also*, *id.* at 10:37-51. Accordingly, the proposed combination of Safford and Kennedy fail to teach or suggest ***"an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system***" as recited in claim 9.

F. Rejection 2, The proposed combination of Kennedy in view of Nguyen and Fox fails to teach or suggest each and every element of Independent Claims 1, 9, 16 and 25

1. **Claims 1 and 16, element [1.2]:** ***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***

The Examiner asserts that Kennedy discloses 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*." NFOA at 28, The portions of Kennedy that the Examiner relied on are as follows:

> The CPU ISTC memories hold configuration information and the ISTC specific to the associated CPU, and the I/O board ISTC memories hold the peripheral configuration information and the ISTC for the associated I/O board. Kennedy at 7:33-35.
>
> As previously explained, the ISTC memory associated with each CPU contains the ISTC code as well as configuration information about the CPU. Thus, ***the boot CPU determines how many CPUs are present*** on the bus 104 by reading the memory location assigned to the ISTC memory for each slot (e.g. D100,0000 hex for slot 1, D200,0000 hex for slot 2, etc). *Id.* at 9:16-21) (emphasis added).
>
> For instance, ***provision needs to be made*** to determine which processor will boot the system, and if this "default' boot processor fails to boot the system, an alternative processor, ***if one is installed***, should take over boot operations to enhance the availability of the system. *Id.* at 7:9-13 (emphasis added).
>
> The architecture provides a method and apparatus to interface multiple processors to the backplane system bus such that upon start-up of the computer system, one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor. *Id.* at 7:33-35.

As can be gleaned from at least the above, Kennedy is directed to "a multiprocessor interconnection architecture with a backplane system bus for use with multiple processors." *Id.* at 2:13-15. According to Kennedy, each CPU, memory, and I/O module for the system maintains a copy of the initialization data and self-test code portion of its own boot code, and potentially its

entire boot code, which can be transferred to memory for execution for the purpose of ***initialization and self-testing of the associated module***." *Id.* at 2:22-27 (emphasis added). The Kennedy system includes a backplane bus with connector slots capable of receiving multiple central processing units, memory modules, and other input/output modules. *Id.* at 2:30-34. The system includes a default boot central processing unit ***installed in a first slot*** on the backplane bus, and may also have one or more alternate boot central processing units installed ***in a second slot*** on the backplane bus. *Id.* at 2:35-38. This physical slotted approach by Kennedy, *i.e.*, placing the default boot processor in the first slot and any alternative processors in secondary slots, is not the same as ***data storage storing information that assigns of boot and spare roles to processors***, as recited in claims 1 and 16. Nguyen and Fox do not cure this deficiency. Moreover, one skilled in the art would not modify Kennedy to yield the claimed invention, because doing so would frustrate the purpose of Kennedy's physical architecture as a solution with regard to overcoming problems associated with which processors boot the system. *See* Kennedy at 1:60-2:12; 2:29-38).

Accordingly, the Examiner has not made a prima facie case that claims 1 and 16, and the claims dependent thereon, are unpatentable over the cited references for at least this reason.

2. **Claims 1, 9, 17, and 25, element [1.3]:** ***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***

The Examiner acknowledges that Kennedy fails to teach or suggest "***detecting loss of lockstep for said boot processor.***" NFOA p. 30 (emphasis added). To overcome this deficiency, the Examiner relies on Nguyen for this limitation missing in Kennedy.

Nguyen is directed to multi-core processors operating in functional redundancy checking mode. Nguyen, Abstract. Each replicated execution core may include one or more caches, register files and supporting resources in addition to the basic execution units (integer, floating point, load

store, etc.). Nguyen at [0006]. Further, each execution core 120 includes a data pipeline 124 and an error pipeline 128 that feed into FRC checker 130 and error detector 140, respectively. *Id.* Data pipeline 124 represents the logic that operates on various types of data assist moves through processor 110 toward FRC checker 130. *Id.* Data processed by data pipeline 124 may include result operands, Status flags, addresses, instructions and the like that are generated and staged through processor 110 during code execution. *Id.*

As with Safford, Nguyen makes no mention of a boot processor. The functions discussed by Nguyen are simply not the functions of *a boot processor*. For example, there is no need for a data pipeline and/or an error pipeline in boot processors. *Id.* Moreover, the only mention of lockstep is an indication that the execution cores operate in lockstep during normal FRC mode, not unlike Safford. *Id.* at [0024]. While the Examiner appears to generalize all processors in the prior art to be one and the same, the boot processor, which is responsible for booting the system, is distinguishable from generic execution cores in a multi-core architecture that are processing requests from applications.

For the "*transferring, during system runtime, the role of boot processor to said spare processor*" element [1.3], the Examiner refers the Wolfe declaration (Ex. 1003 at ¶205), arguing that Kenndy discusses "replac[ing] the boot processor with the alternative processor in response to a failure of the boot processor…" NFOA at p. 30. First, Wolfe's statements are merely conclusory and are not entitled to any weight. Second, as discussed above, the terms "*during system runtime*" as used in claims 1 and 9 means "***after the start of the normal boot process.***" Moreover, because the transfer occurs ***responsive to detecting loss of lockstep for said boot processor***, the terms "*transferring … the role of boot processor to the spare processor*" correspondingly imply that such role transfer occurs after the start and during the normal boot

process. Kennedy does not attempt to recover during the boot process, rather, it merely determines which of the available processors will be used to boot the system at ***startup***, based on the physical location of the processor (Kennedy at 7:5-13, *see also, id.* at 10:37-51). Accordingly, Kennedy in view of Nguyen does not disclose transferring the role of the processor after startup and during the normal boot process, in accordance with the claims. Moreover, the Examiner has not shown that it was desirable or even possible to transfer boot roles to the spare "*during system runtime*", *i.e.*, after the start of and during the normal boot process, particularly with regard to the slotted architecture taught by Kennedy. Accordingly, the Examiner has failed to provide sufficient basis for combing and modifying Kennedy and Nguyen to support a finding of obviousness.

Claims 9, 17, and 25 recite similar elements. Accordingly, Kennedy in view of Nguyen and Fox do not disclose all of the elements of claim 1, 9, 17, and 25.

3. ***Claims 1, 9, 16, and 26*** **– No basis for combining Kennedy in view of Nguyen and Fox to yield the claimed invention.**

As with Rejection 1, the Examiner failed to provide sufficient basis for combining and modifying the combined disclosures of Kennedy, Nguyen, and Fox to yield the claimed invention. Conclusory statements that the combination will ***decrease delays in failing over from the boot processor to the spare processor*** without evidentiary support are not enough for the examiner to make a *prima facie* case of obviousness. Notwithstanding, Examiner here too has failed to correctly assess the differences between Kennedy, Nguyen, and Fox and ultimately the differences between the art and the claimed invention.

Kennedy teaches default and alternative processors operating alternatively and in succession at the time that the system is ***initialized***, *i.e.*, before the operating system boot process.

Kennedy at 7:4-13; 8:5-8 ("When a circuit board is initializing, the board specific ISTC is transferred from the associated ISTC memory to the memory 130 by the boot processor and assembled into executable code."), 9:53-55 ("Once each CPU in the system has executed its ISTC and has checked in, the CPU1 120 controls the system again and begins the operating system boot process.") Because Kennedy relates to the startup process only, *delays from failing over* is not a concern because the time to boot a processor is relatively short, as compared to booting the operating system and the associated applications for continuity. Moreover, it is understood that execution of applications by the operating system does not occur until after the operating system has booted. *See. e.g.*, Fox at 1:25-27. It is also understood that the desire to maintain application continuity is a concern after the operating system has booted. *See id.* Fox, for instance, proposes a solution that involves "hot" swapping processors after the operating system has booted. *See id.*, at 3:3-15. Because Nguyen and Fox provide solutions unrelated contextually to Kennedy (Nguyeen and Fox, *supra*), a person of ordinary skilled in the art would not look to combine Kennedy without any of the other reference of record.

Moreover, as also discussed above, there is no prior art evidence showing (1) boot processors operating in lockstep, and (2) that loss of lockstep in boot processor pairs could be resolved without crashing the system, with the combined Kennedy, Nguyen, and Fox.

The Examiner has therefore failed to make a *prima facie* case of obviousness with regard to claims 1-30 for at least the reasons discussed above.

III. Conclusion

The other dependent claims are patentable over the references of record for the same reasons noted above, as well as additional reasons. The Patent Owner reserves the right to argue additional distinctions in its discretion.

In view of the foregoing, the Patent Owner respectfully requests that the Examiner certify the patentability of claims 1-30 of the '781 Patent.

Date: May 20, 2024

Respectfully submitted,

/Antonio Papageorgiou/
Antonio Papageorgiou
Reg. No. 53,431

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/019,244	09/01/2023	7627781		7031

199393 7590 05/01/2024

Atlantic IP
c/o Lombard Geliebter LLP
1325 Sixth Aveneue
28th Floor
New York, NY 10019

EXAMINER	
BONSHOCK, DENNIS G	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
05/01/2024	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patents and Trademark Office
P.O.Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

5/1/2024

THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS

GREENBERG TRAURIG (NJ)
500 CAMPUS DRIVE, SUITE 400
P.O. BOX 677
FLORHAM PARK, NEW JERSEY 07932

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. : 90019244
PATENT NO. : 7627781
ART UNIT : 3992

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above-identified ex parte reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the ex parte reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Decision on Petition for Extension of Time in Reexamination	Application No.: 90/019,244	Applicant(s): 7,627,781	
	Examiner BONSHOCK, DENNIS	Art Unit 3992	

1. THIS IS A DECISION ON THE PETITION FILED 04/22/2024.

2. THIS DECISION IS ISSUED PURSUANT TO:
   - A. ☒ 37 CFR 1.550(c) – The time for taking any action by a patent owner in a third party requested *ex parte* reexamination proceeding will be extended only for sufficient cause and for a reasonable time specified.
   - B. ☐ 37 CFR 1.550(c) – The time for taking action by a patent owner in a patent owner requested *ex parte* reexamination proceeding will only be extended for more than two months for sufficient cause and for a reasonable time specified.
   - C. ☐ 37 CFR 1.956 – The time for taking any action by a patent owner in an *inter partes* reexamination proceeding will be extended only for sufficient cause and for a reasonable time specified.

   The petition is before the Central Reexamination Unit for consideration.

3. FORMAL MATTERS: Patent Owner requests that the time period for filing a response to the non-final Office action mailed on 02/20/2024, which set a one (1) month shortened statutory period for response, be extended by one (1) month.
   - A. Petition fee per 37 CFR § 1.17(g)):
     - i. ☐ Petition includes authorization to debit a deposit account.
     - ii. ☒ Petition includes authorization to charge a credit card account.
     - iii. ☐ Other: _____.
   - B. ☒ Proper certificate of service was provided. (Not required in reexamination where patent owner is requester).
   - C. ☒ Petition was timely filed.
   - D. ☒ Petition properly signed.

4. DECISION (See MPEP 2265 and 2665)
   - A. ☒ Granted or ☐ Granted-in-part for _____. (See 37 CFR 1.550(c) and 37 CFR 1.956).
     - i. ☐ Other/comment: _____.
   - B. ☐ Dismissed because:
     - i. ☐ Formal matters (See unchecked box(es) (A, B, C and/or D) in section 3 above).
     - ii. ☐ Petitioner failed to provide a factual accounting of reasonably diligent behavior by all those responsible for preparing a response to the outstanding Office action within the statutory time period.
     - iii. ☐ Petitioner failed to explain why, in spite of the action taken thus far, the requested additional time is needed.
     - iv. ☐ The statements provided fail to establish sufficient cause to warrant extension of the time for taking action (See attached).
     - v. ☐ The petition is moot.
     - vi. ☐ Other/comment: _____.

5. CONCLUSION:

Direct telephone inquiries to either QAS Stephen J. Stein at 571-272-1544 or SPRS Andrew J. Fischer at 571-272-6779.	/Andrew J. Fischer/, Supervisory Patent Reexamination Specialist, Central Reexamination Unit

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

FEE ADDRESS CHANGE

APPLICATION #	FILING OR 371(C) DATE	REQUEST ID
90/019,244	09/01/2023	206518

The following fields have been changed to Customer Number 199393 on 04/26/2024 via Patent Center in view of the certification copied below that authorized the change.

- Maintenance Fee Address

The address for Customer Number 199393 is:

199393 - Atlantic IP
c/o Lombard Geliebter LLP
1325 Sixth Aveneue
28th Floor
New York, NY 10019

I certify, in accordance with 37 CFR 1.4(d)(4), that I am :

An attorney or Agent of Record registered to practice before the Patent and Trademark Office who has been given power of attorney in this application

Signature	**Name**	**Registration #**
/Antonio PAPAGEORGIOU/	Antonio Papageorgiou	53431

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
90/019,244	**04/24/2024 10:53:54 AM Z ET**	**-**

Title of Invention

SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

Application Information

APPLICATION TYPE	Utility / ex parte reexam	PATENT #	7627781
CONFIRMATION #	7031	FILED BY	Laura Alto
PATENT CENTER #	65222008	FILING DATE	09/01/2023
CUSTOMER #	89220	FIRST NAMED INVENTOR	- 7627781
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Antonio Papageorgiou

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
FTL-REX244_Cert_of_Service_04-24-2024.pdf	1	Reexam Certificate of Service	68 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
FTL-REX244_Cert_of_Service_04-24-2024.pdf	40E5DFB722D3E46802CEDFA6A85C52AEDB337BF4BC87278F4A67D7508DA0FE56AE8784F640E5A0414AE089276A3AA5CD652B46B9FF199AE16065D158804FE977

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of	:	Michaelis, Scott L, et al.	
Application No.	:	90/019,244	Confirmation No. : 7031
Patent No.	:	7,627,781	
Filing Date	:	October 25, 2004	Issue Date: December 1, 2009
Title	:	SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR	

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of the foregoing Request for an Extension of Time Pursuant to 37 C.F.R. 1.550(c); Declaration of Brett Cooper in Support of Petition to for Extension of Time to Respond to a Reexamination Office Action Pursuant to 37 CFR 1.550(c), Power of Attorney and Assignee showing of ownership per 37 CFR 3.73 was electronically filed on April 22, 2024 and served upon counsel of record via First Class Mail on April 24, 2024 to:

GREENBERG TRAURIG (NJ)
500 CAMPUS DRIVE, SUITE 400
P.O. BOX677
FLORHAM PARK NJ 07932

Date: April 24, 2024

Customer No. 89220

/Antonio Papageorgiou/
Antonio Papageorgiou
USPTO Reg. No. 53431

Attorney for Patent Owner, Foras Technologies Ltd

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
90/019,244	09/01/2023	7627781	

CONFIRMATION NO. 7031

89220
Antonio Papageorgiou
Lombard Geliebter LLP
230 PARK AVENUE 4TH FL WEST
New York, NY 10169

POA ACCEPTANCE LETTER

OC000000071516912

Date Mailed: 04/24/2024

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 04/22/2024.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/pflewis/

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
90/019,244	09/01/2023	7627781	

CONFIRMATION NO. 7031

POWER OF ATTORNEY NOTICE

OC000000071516833

56436
Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528

Date Mailed: 04/24/2024

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 04/22/2024.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/pflewis/

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
90/019,244	**04/22/2024 06:22:59 PM Z ET**	**-**

Title of Invention

SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

Application Information

APPLICATION TYPE	Utility / ex parte reexam	PATENT #	7627781
CONFIRMATION #	7031	FILED BY	Laura Alto
PATENT CENTER #	65196989	FILING DATE	09/01/2023
CUSTOMER #	56436	FIRST NAMED INVENTOR	- 7627781
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Antonio Papageorgiou

Documents

TOTAL DOCUMENTS: 4

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
FTL-REX244_POA_04-22-2024.pdf	2	Power of Attorney	753 KB
FTL-REX244_Statement of Assignee_04-22-2024.pdf	3	Assignee showing of ownership per 37 CFR 3.73	225 KB
FTL-REX244_Petition for Extension Reexamination_04-22-2024.pdf	2	Reexam Request for Extension of Time	80 KB
FTL-REX244_Declaration_in_support_extension_04-22-	14	Oath or Declaration filed	7396 KB

2024.pdf

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
FTL-REX244_POA_04-22-2024.pdf	062D5011891F2F9FD71E071C5273C4AABBFEE3884E6187A0B7DE1D670C3E28E50FC8233EE89F109CAC37FE19E7FB63A6D00193DF653F70177AD13B31BA8CE52C
FTL-REX244_Statement of Assignee_04-22-2024.pdf	CEB8AA98E6656050BE4C198542814417C5B7C9E6AE8A1B796CDFADD07D1C04EA66FBC32CE863CE6496942A35ED12249582A0D871468E275DA7F655C5E9E55A30
FTL-REX244_Petition for Extension Reexamination_04-22-2024.pdf	2C0798BADF6AAC9BAC33413D99EDEFA52FFB6E17CC13066AA5615F7ACBDCC184B4226DDA80380F4E398A0349E2F3C4FDD277A1948A0A158178DA23071AE2D494
FTL-REX244_Declaration_in_support_extension_04-22-2024.pdf	67B56ED82EB4DF35E11F5C7383B1DF3DAE5F30265A9D009C6BB0AA4E179588DAF835D45453A613ECC9D39D4A3B5D73AA31F96E53FDE1B9971107009C31AF2225

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventors	:	Michaelis, Scott L, et al.		
Application No.	:	90/019,244	Confirmation No. :	7031
Patent No.	:	7,627,781		
Filing Date	:	October 25, 2004	Issue Date:	December 1, 2009
Title	:	System And Method For Establishing A Spare Processor For Recovering From Loss Of Lockstep In A Boot Processor		

Request for an Extension of Time Pursuant to 37 C.F.R. 1.550(c)

A non-final office action was issued in the above referenced reexamination of the on February 20, 2024, with a two-month response period making a reply from the patent owner due on April 22, 2024.

The patent owner is requesting a one month extension to file a reply to the non-final office action pursuant to 37 C.F.R. 1.550(c). The patent owner submits that this extension is reasonable.

The fee pursuant to 1.17(g) accompanies this petition.

As noted in the declaration in support of this petition, the patent owner has sufficient cause for the Office to grant its request. Specifically, the third party requester, Unified Patents LLC, served the request for reexamination using the customer number associated with the previous assignor, Hewlett Packard Enterprises. Although the previous assignor would forward notice to the patent owner, this time it did not. As a result, the patent owner nor its representatives received notice of the non-final office action until the due date for a response, April 22, 2024. Specifically, Mr. Brett Cooper only became aware of the reexamination in connection with a pre-trial

conference that was held today. (See Cooper Decl. at ¶ 7). Because of the late notice of the Office Action, FORAS does not have the ability to meaningfully reply to the Office Action.

Accordingly, grant of this request is respectfully requested.

Date: April 22, 2024

Respectfully submitted,

/Antonio Papageorgiou/
Antonio Papageorgiou

Reg. No. 53,431

PTO/AIA/96 (11-23)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

First Named Inventor/Patent Owner: Foras Technologies Ltd.

Application No./Patent No.: 7,627,781 Filed/Issue Date: December 1, 2009

Titled: SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

Foras Technologies Ltd., a corporation

(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. ☑ The assignee of the entire right, title, and interest.

2. ☐ An assignee of less than the entire right, title, and interest (check applicable box):

   ☐ The extent (by percentage) of its ownership interest is _____%. Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.

   ☐ There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. ☐ The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. ☐ The recipient, via a court proceeding or the like (*e.g.*, bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached*.

B. ☑ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Scott L. Michaelis & Anurupa Rajkumari To: Hewlett-Packard Development Company, L.P.

   The document was recorded in the United States Patent and Trademark Office at Reel 015941, Frame 0381, or for which a copy thereof is attached*.

2. From: Hewlett-Packard Development Company, L.P. To: Hewlett Packard Enterprise Development LP

   The document was recorded in the United States Patent and Trademark Office at Reel 037079, Frame 0001, or for which a copy thereof is attached*.

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-0031. Public burden for this form is estimated to average 12 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/AIA/96 (11-23)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

3. From: Hewlett Packard Enterprise Development LP, Hewlett Packard Enterprise Company To: Sonraí Memory, Ltd.

The document was recorded in the United States Patent and Trademark Office at
Reel 052567, Frame 0734, or for which a copy thereof is attached*.

4. From: Sonraí Memory, Ltd. To: Foras Technologies Ltd.

The document was recorded in the United States Patent and Trademark Office at
Reel 058992, Frame 0571, or for which a copy thereof is attached*.

5. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached*.

6. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached*.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

☑ *As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

/Antonio Papageorgiou/	April 22, 2024
Signature	Date
Antonio Papageorgiou	53431
Printed or Typed Name	Title or Registration Number

[Page 2 of 2]

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. The United States Patent and Trademark Office (USPTO) collects the information in this record under authority of 35 U.S.C. 2. The USPTO's system of records is used to manage all applicant and owner information including name, citizenship, residence, post office address, and other information with respect to inventors and their legal representatives pertaining to the applicant's/owner's activities in connection with the invention for which a patent is sought or has been granted. The applicable Privacy Act System of Records Notice for the information collected in this form is COMMERCE/PAT-TM-7 Patent Application Files, available in the Federal Register at 78 FR 19243 (March 29, 2013). https://www.govinfo.gov/content/pkg/FR-2013-03-29/pdf/2013-07341.pdf

Routine uses of the information in this record may include disclosure to:

1) law enforcement, in the event that the system of records indicates a violation or potential violation of law;

2) a federal, state, local, or international agency, in response to its request;

3) a contractor of the USPTO having need for the information in order to perform a contract;

4) the Department of Justice for determination of whether the Freedom of Information Act (FOIA) requires disclosure of the record;

5) a Member of Congress submitting a request involving an individual to whom the record pertains, when the individual has requested the Member's assistance with respect to the subject matter of the record;

6) a court, magistrate, or administrative tribunal, in the course of presenting evidence, including disclosures to opposing counsel in the course of settlement negotiations;

7) the Administrator, General Services Administration (GSA), or their designee, during an inspection of records conducted by GSA under authority of 44 U.S.C. 2904 and 2906, in accordance with the GSA regulations and any other relevant (i.e., GSA or Commerce) directive, where such disclosure shall not be used to make determinations about individuals;

8) another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c));

9) the Office of Personnel Management (OPM) for personnel research purposes; and

10) the Office of Management and Budget (OMB) for legislative coordination and clearance.

If you do not furnish the information requested on this form, the USPTO may not be able to process and/or examine your submission, which may result in termination of proceedings, abandonment of the application, and/or expiration of the patent.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
90/019,244	04/22/2024 06:22:59 PM Z ET	-

Title of Invention

SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

Application Information

APPLICATION TYPE	Utility / ex parte reexam	PATENT #	7627781
CONFIRMATION #	7031	FILED BY	Laura Alto
PATENT CENTER #	65196989	AUTHORIZED BY	Antonio Papageorgiou
CUSTOMER #	56436	FILING DATE	09/01/2023
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	- 7627781

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 1047	E20244LI23238913	Laura Alto

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
1463	PETITIONS REQUIRING THE PETITION FEE SET FORTH IN 37 CFR 1.17(G) (GROUP II)	220.00	1	220.00
			TOTAL AMOUNT:	$220.00

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/AIA/81B (07-13)
Approved for use through 09/30/2025. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are required to respond to a collection of information unless it displays a valid OMB control number

REEXAMINATION OR SUPPLEMENTAL EXAMINATION – PATENT OWNER POWER OF ATTORNEY OR REVOCATION OF POWER OF ATTORNEY WITH A NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS FOR REEXAMINATION OR SUPPLEMENTAL EXAMINATION AND PATENT

Control Number(s)	90/019,244
Filing Date(s)	September 1, 2023
First Named Inventor	Scott L. Michaelis
Title	System And Method For Estab sh ng A Spare Processor For Recovering
Patent Number	7,627,781
Examiner Name	Dennis G. Bonshock
Attorney Docket No(s)	FTL-REX244

I. Power of Attorney. This form may be used to change the Power of Attorney in a reexamination or supplemental examination proceeding (or multiple proceedings where merged). This form may also be used to change the Power of Attorney in the patent file; in such a case, a copy of this form will be placed in both the patent file and the reexamination or supplemental examination proceeding.

A. Revocation of Previous Power of Attorney. I hereby revoke all previous patent owner powers of attorney, if any, given:

■ in the above-identified reexamination or supplemental examination proceeding control number(s) (more than one may be changed only if the proceedings are merged).

■ in the file of the above-identified patent.

(check BOTH boxes if change in BOTH the patent file and the reexamination or supplemental examination proceeding is requested).

B. Designation of Power of Attorney.

☐ A Power of Attorney is submitted herewith.

OR

■ I hereby appoint Practitioner(s) associated with the Customer Number identified in the box at right as my/our attorney(s) or agent(s) to prosecute the proceeding(s)/patent identified above and selected in section I(A), and to transact all business in the United States Patent and Trademark Office connected therewith:

89220

OR

☐ I hereby appoint Practitioner(s) named below as my/our attorney(s) or agent(s) to prosecute the proceeding(s) identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

Practitioner(s) Name	Registration Number

Authorization for the Power of Attorney is provided by the signature on page 2 of this form.

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-0035. Public burden for this form is estimated to average 15 minutes hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **If filing this completed form by mail, send to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

II. Change of Correspondence Address

Please recognize or change the correspondence address for the above-identified reexamination or supplemental examination proceeding control number(s) (more than one may be changed only if they are merged proceedings) and for the file of the above-identified patent to be:

■ The address associated with the above-identified Customer Number.
OR
☐ The address associated with the Customer Number identified in the box at right:
OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

NOTE: THE CORRESPONDENCE ADDRESS FOR THE REEXAMINATION OR SUPPLEMENTAL EXAMINATION PROCEEDING CONTROL NUMBER(S) MUST BE THE SAME AS THAT FOR THE PATENT. SEE 37 CFR 1.33.

III. Authorization for Power of Attorney and (if selected) Change of Correspondence Address

I am the:
☐ Inventor, having ownership of the patent being reexamined.
OR
■ Patent owner.
Statement under 37 CFR 3.73(c) (Form PTO/AIA/96) submitted herewith or filed on _____.

Signature of Inventor or Patent Owner		Date	April 22,2024
Name	CIARAN O'GARA	Telephone	
Title and Company	Director , Foras Technologies Ltd.		

NOTE: Signatures of all the inventors or patent owners of the entire interest or their representative(s) are required. If more than one signature is required, submit multiple forms, check the box below, and identify the total number of forms submitted in the blank below.

■ A total of 1 forms are submitted. *If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventors	:	Michaelis, Scott L, et al.	
Application No.	:	90/019,244	Confirmation No. : 7031
Patent No.	:	7,627,781	
Filing Date	:	10/25/2004	Issue Date: 12/01/2009
Title	:	System And Method For Establishing A Spare Processor For Recovering From Loss Of Lockstep In A Boot Processor	

DECLARATION OF BRETT COOPER IN SUPPORT OF PETITION TO FOR EXTENSION OF TIME TO RESPOND TO A REEXAMINATION OFFICE ACTION PURSUANT TO 37 CFR 1.550(c)

I, the undersigned, hereby declare and state that:

1) U.S. Patent No. 7,627,781 (the "'781 Patent") is owned by FORAS TECHNOLOGIES LTD. ("FORAS") (*see* Frame/Reel No. 058992/0571) (Ex. A, Assignment 4), which is located at the following address:

THE HYDE BUILDING, SUITE 23
THE PARK, CARRICKMINES
DUBLIN 18, IRELAND

2) I am an attorney for FORAS in a litigation involving the '781 Patent, namely *Foras Technologies Ltd. v. Nissan Motor Company, Ltd, et al.*, 1:23-cv-640 (EDTX) (the "Action"). *See* Ex. B.

3) I am authorized to act on behalf of the assignee, FORAS, including to submit this statement on its behalf in support of the extension of time herein to respond to the Office Action dated February 20, 2024 (the "Office Action").

4) The request for reexamination of the '781 Patent was filed on September 1, 2023. The certificate of service by the third party requester (Unified Patents, LLC) indicates that the request and the accompanying appendices and exhibits were served by mail as follows:

FORAS TECHNOLOGIES LTD.
c/o 56436 - Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528

5) This customer number is associated with the previous assignor, Hewlett Packard Enterprise ("HP"). (*See* Ex. A, Assignment 3). On information and belief, the third party requester did not attempt to serve FORAS directly, including at the assignee address noted above.

6) HP is under no obligation to forward documents concerning the assigned patents, and on information and belief, HP did not alert FORAS and FORAS did not therefore receive any documents concerning the '781 Patent that were mailed to the HP address, including the request for reexamination and the Office Action in this reexamination.

7) FORAS first learned about this reexamination and the Office Action today, April 22, 2024, while getting prepared for today's initial pretrial conference with Judge Howell in the Western District of Texas. *See* Ex. B (D.I. 40). Specifically, a motion to stay proceedings in the Action (filed on April 10, 2024) listed the existence of this reexamination – the opposition to this motion to stay is not due until Friday, April 26, 2024. Once the existence of this reexamination was known, I checked the status on the USPTO Patent Center. This is when FORAS first learned that an Office Action and that a response thereto is due today, April 22, 2024.

8) Because of the late notice of the Office Action, FORAS does not have the ability to meaningfully reply to the Office Action.

9) Accordingly, FORAS respectfully requests a one-month extension of time, which would extend the deadline for a response to May 20, 2024, for it to reply to the Office Action.

10) I hereby declare that all statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and all statements made herein are made with the knowledge that whoever knowingly and

willfully makes any false, fictitious, or fraudulent statements or representations shall be subject to the penalties set forth under 18 U.S.C. 1001.

Date: April 22, 2024

Respectfully submitted,

/Brett Cooper/
Brett Cooper, Esq.
For Foras Technologies Ltd.

EXHIBIT A

UNITED STATES
PATENT AND TRADEMARK OFFICE

Assignment abstract of title for Application 10973077

Invention title/Inventor	Patent	Publication	Application	PCT	International registration
SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR Scott L. Michaelis, Anurupa Rajkumari	7627781 Dec 1, 2009	20060107112 May 18, 2006	10973077 Oct 25, 2004		

Assignments (4 total)

Assignment 4

Reel/frame	Execution date	Date recorded	Properties	Pages
058992/0571	Jan 19, 2022	Feb 11, 2022	5	5

Conveyance
ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors
SONRAÍ MEMORY, LTD.

Correspondent
VORYS, SATER, SEYMOUR AND PEASE LLP
1909 K STREET NW
SUITE 900
WASHINGTON, DC 20006-1152

Assignee
FORAS TECHNOLOGIES LTD.
THE HYDE BUILDING, SUITE 23,
THE PARK, CARRICKMINES,
DUBLIN 18
IRELAND

Assignment 3

Reel/frame	Execution date	Date recorded	Properties	Pages
052567/0734	Apr 23, 2020	May 5, 2020	17	6

Conveyance
ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors
HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP
HEWLETT PACKARD ENTERPRISE COMPANY

Correspondent
ADAM MULLOOLY
SUITE 23, THE HYDE BUILDING, THE PARK, CARRICKMINE
DUBLIN, DUBLIN 18 IRELAND

Assignee
SONRAÍ MEMORY, LTD.
SUITE 23, THE HYDE BUILDING, THE PARK, CARRICKMINE
DUBLIN DUBLIN 18
IRELAND

Assignment 2

Reel/frame	Execution date	Date recorded	Properties	Pages
037079/0001	Oct 27, 2015	Nov 9, 2015	11870	1112

Conveyance
ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors
HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.

Correspondent
HEWLETT PACKARD ENTERPRISE
3404 E. HARMONY ROAD MS 79
FORT COLLINS, CO 80528

Assignee
HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP
11445 COMPAQ CENTER DRIVE WEST
HOUSTON, TEXAS 77070

Assignment 1

Reel/frame	Execution date	Date recorded	Properties	Pages
015941/0381	Oct 20, 2004	Oct 25, 2004	1	3

Conveyance
ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors
MICHAELIS, SCOTT L.
RAJKUMARI, ANURUPA

Correspondent
HEWLETT-PACKARD COMPANY
RECORDS MANAGER
INTELLECTUAL PROPERTY ADMINISTRATION
P.O. BOX 272400
FORT COLLINS, CO 80527-2400

Assignee
HEWLETT-PACKARD DEVELOPMENT COMPANY, LP
20555 SH 249
HOUSTON, TEXAS 77070

EXHIBIT B

PATENT

U.S. District Court [LIVE]
Western District of Texas (Austin)
CIVIL DOCKET FOR CASE #: 1:23-cv-00640-RP

Foras Technologies Ltd. v. Nissan Motor Company, Ltd. et al
Assigned to: Judge Robert Pitman
Related Case: 6:23-cv-00386-RP
Cause: 35:271 Patent Infringement

Date Filed: 06/06/2023
Jury Demand: Plaintiff
Nature of Suit: 830 Patent
Jurisdiction: Federal Question

Plaintiff

Foras Technologies Ltd.

represented by **Jonathan Randy Yim**
BC Law Group, P.C.
200 Madison Avenue
24th Floor
New York, NY 10016
516-359-9668
Email: jyim@bclgpc.com
LEAD ATTORNEY
ATTORNEY TO BE NOTICED

Ashley Marie Ratycz
BC Law Group, P.C.
200 Madison Avenue, 24th Floor
New York, NY 10016
212-951-0100
Email: aratycz@bclgpc.com
ATTORNEY TO BE NOTICED

Drew B Hollander
BC Law Group, P.C.
200 Madison Avenue
24th Floor
New York, NY 10016
212-951-0100
Email: dhollander@bclgpc.com
ATTORNEY TO BE NOTICED

Seth Raymond Hasenour
BC Law Group, P.C.
607 Terrell Hill Drive
Ste Unit B
Austin, TX 78704
516-359-9668
Email: shasenour@bclgpc.com
ATTORNEY TO BE NOTICED

Brett E Cooper
BC LAW GROUP, P.C.
200 Madison Avenue, 24th Floor
New York, NY 10016

516-359-9668
Email: bcooper@b-clg.com
ATTORNEY TO BE NOTICED

V.

Defendant

Nissan Motor Company, Ltd. represented by

Binal J. Patel
Banner & Witcoff, LTD
71 S. Wacker Drive, Ste 3600
Chicago, IL 60606
312-463-5463
Fax: 312-463-5001
Email: bpatel@bannerwitcoff.com
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

John R. Hutchins
Banner & Witcoff, Ltd.
1100 13th St. Nw, Suite 1200
Washington, DC 20005
202-824-3000
Fax: 202-824-3001
Email: jhutchins@bannerwitcoff.com
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

Paige Arnette Amstutz
Scott, Douglass & McConnico, LLP
303 Colorado Street
Suite 2400
Austin, TX 78701
(512) 495-6300
Fax: 512-495-6399
Email: pamstutz@scottdoug.com
LEAD ATTORNEY
ATTORNEY TO BE NOTICED

Paul Thomas Qualey
Banner & Witcoff, LTD
1100 13th St., NW, Suite 1200
Washington D.C., DC 20005
202-824-3000
Fax: 202-824-3001
Email: pqualey@bannerwitcoff.com
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

Wesley W. Jones
Banner & Witcoff, Ltd.

1100 13th St. Nw, Suite 1200
Washington, DC 20005
202-824-3000
Fax: 202-824-3001
Email: wjones@bannerwitcoff.com
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

Defendant

ZF Friedrichshafen AG represented by **Binal J. Patel**
(See above for address)
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

John R. Hutchins
(See above for address)
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

Paige Arnette Amstutz
(See above for address)
LEAD ATTORNEY
ATTORNEY TO BE NOTICED

Paul Thomas Qualey
(See above for address)
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

Wesley W. Jones
(See above for address)
LEAD ATTORNEY
PRO HAC VICE
ATTORNEY TO BE NOTICED

Date Filed	#	Docket Text
06/06/2023	1	COMPLAINT (Filing fee $ 402 receipt number ATXWDC-17522685), filed by Foras Technologies Ltd.. (Attachments: # 1 Exhibit 1, # 2 Exhibit 2, # 3 Exhibit 3, # 4 Exhibit 4, # 5 Exhibit 5, # 6 Exhibit 6, # 7 Exhibit 7, # 8 Exhibit 8, # 9 Exhibit 9, # 10 Exhibit 10, # 11 Exhibit 11, # 12 Civil Cover Sheet)(Cooper, Brett) (Entered: 06/06/2023)
06/06/2023	2	Notice of Filing of Patent/Trademark Form (AO 120). AO 120 forwarded to the Director of the U.S. Patent and Trademark Office. (Cooper, Brett) (Entered: 06/06/2023)
06/06/2023	3	RULE 7 DISCLOSURE STATEMENT filed by Foras Technologies Ltd.. (Cooper, Brett) (Entered: 06/06/2023)
06/06/2023	4	NOTICE *Notice of Related Cases* by Foras Technologies Ltd. (Cooper, Brett) (Entered: 06/06/2023)

06/06/2023	5	REQUEST FOR ISSUANCE OF SUMMONS by Foras Technologies Ltd.. *as to Nissan Motor Company, Ltd.* (Cooper, Brett) (Entered: 06/06/2023)
06/06/2023	6	REQUEST FOR ISSUANCE OF SUMMONS by Foras Technologies Ltd.. *as to ZF Friedrichshafen AG* (Cooper, Brett) (Entered: 06/06/2023)
06/06/2023		Case assigned to Judge Robert Pitman. CM WILL NOW REFLECT THE JUDGE INITIALS AS PART OF THE CASE NUMBER. PLEASE APPEND THESE JUDGE INITIALS TO THE CASE NUMBER ON EACH DOCUMENT THAT YOU FILE IN THIS CASE. (pg) (Entered: 06/09/2023)
06/06/2023		If ordered by the court, all referrals and consents in this case will be assigned to Magistrate Judge Howell (pg) (Entered: 06/09/2023)
06/09/2023	7	Summons Issued as to Nissan Motor Company, Ltd. (pg) (Entered: 06/09/2023)
06/09/2023	8	Summons Issued as to ZF Friedrichshafen AG. (pg) (Entered: 06/09/2023)
06/09/2023	9	Report on Patent/Trademark sent to U.S. Patent and Trademark Office with Complaint. (pg) (Entered: 06/09/2023)
09/12/2023	10	MOTION *for Order Appointing International Process Server* by Foras Technologies Ltd.. (Attachments: # 1 Proposed Order)(Cooper, Brett) (Entered: 09/12/2023)
09/27/2023	11	ORDER GRANTING 10 MOTION for Order Appointing International Process Server. Signed by Judge Robert Pitman. (cc3) (Entered: 09/27/2023)
12/04/2023	12	Unopposed MOTION for Extension of Time to File Response/Reply as to 1 Complaint, *ZF Friedrichshafen AG's Unopposed Motion for Extension of Time to Respond to the Complaint* by ZF Friedrichshafen AG. (Attachments: # 1 Proposed Order Proposed Order) (Amstutz, Paige) (Entered: 12/04/2023)
12/05/2023		Text Order GRANTING 12 Unopposed Motion for Extension of Time to File Response/Reply entered by Judge Robert Pitman. IT IS ORDERED that ZF FRIEDRICHSHAFEN AG shall answer or otherwise respond to the complaint on or before January 8, 2024. (This is a text-only entry generated by the court. There is no document associated with this entry.) (sh) (Entered: 12/05/2023)
12/05/2023		Reset Answer Deadline: ZF Friedrichshafen AG answer due 1/8/2024. (cc3) (Entered: 12/05/2023)
12/06/2023	13	SUMMONS Returned Executed by Foras Technologies Ltd.. ZF Friedrichshafen AG served on 11/13/2023, answer due 1/8/2024. (Cooper, Brett) (Entered: 12/06/2023)
12/28/2023	14	Unopposed MOTION for Extension of Time to File Answer re 1 Complaint, by Nissan Motor Company, Ltd.. (Attachments: # 1 Proposed Order to Unopposed Motion)(Amstutz, Paige) (Entered: 12/28/2023)
01/02/2024		Text Order GRANTING 14 Unopposed Motion for Extension of Time to Answer entered by Judge Robert Pitman. IT IS ORDERED that Nissan shall answer or otherwise respond to the complaint on or before February 9, 2024. (This is a text-only entry generated by the court. There is no document associated with this entry.) (sh) (Entered: 01/02/2024)
01/02/2024		Reset Answer Deadline: Nissan Motor Company, Ltd. answer due 2/9/2024. (cc3) (Entered: 01/02/2024)
01/08/2024	15	MOTION to Dismiss for Lack of Jurisdiction *Defendant ZF Friedrichshafen AG's Motion to Dismiss the Complaint Pursuant to Fed. R. Civ. P. 12(b)(2)* by ZF Friedrichshafen AG. (Attachments: # 1 Affidavit Declaration of David Council ISO ZF Friedrichshafen AG's Motion to Dismiss Plaintiff's Complaint)(Amstutz, Paige) (Entered: 01/08/2024)

01/10/2024	16	RULE 7 DISCLOSURE STATEMENT filed by ZF Friedrichshafen AG. (Amstutz, Paige) (Entered: 01/10/2024)
01/16/2024	17	Unopposed MOTION for Extension of Time to File Response/Reply as to 15 MOTION to Dismiss for Lack of Jurisdiction *Defendant ZF Friedrichshafen AG's Motion to Dismiss the Complaint Pursuant to Fed. R. Civ. P. 12(b)(2)* by Foras Technologies Ltd.. (Attachments: # 1 Proposed Order)(Cooper, Brett) (Entered: 01/16/2024)
01/18/2024		Text Order GRANTING 17 Unopposed Motion for Extension of Time to File Response/Reply entered by Judge Robert Pitman. IT IS ORDERED that Plaintiff shall respond on or before January 29, 2024. (This is a text-only entry generated by the court. There is no document associated with this entry.) (sh) (Entered: 01/18/2024)
01/29/2024	18	Response in Opposition to Motion, filed by Foras Technologies Ltd., re 15 MOTION to Dismiss for Lack of Jurisdiction *Defendant ZF Friedrichshafen AG's Motion to Dismiss the Complaint Pursuant to Fed. R. Civ. P. 12(b)(2)* filed by Defendant ZF Friedrichshafen AG (Cooper, Brett) (Entered: 01/29/2024)
02/05/2024	19	RESPONSE in Support, filed by ZF Friedrichshafen AG, re 15 MOTION to Dismiss for Lack of Jurisdiction *Defendant ZF Friedrichshafen AG's Motion to Dismiss the Complaint Pursuant to Fed. R. Civ. P. 12(b)(2)* filed by Defendant ZF Friedrichshafen AG *Defendant ZF Friedrichshafen AG's Reply Brief In Support of its Motion to Dismiss the Complaint Pursuant to Fed. R. Civ. P. 12(b)(2)* (Amstutz, Paige) (Entered: 02/05/2024)
02/06/2024	20	SUMMONS Returned Executed by Foras Technologies Ltd.. Nissan Motor Company, Ltd. served on 12/20/2023, answer due 2/9/2024. (Cooper, Brett) (Entered: 02/06/2024)
02/09/2024	21	NOTICE *of Translation to English* by Foras Technologies Ltd. re 20 Summons Returned Executed (Cooper, Brett) (Entered: 02/09/2024)
02/09/2024	22	NOTICE *of Translation to English* by Foras Technologies Ltd. re 13 Summons Returned Executed (Cooper, Brett) (Entered: 02/09/2024)
02/09/2024	23	Motion to Dismiss for Lack of Jurisdiction *Defendant Nissan Motor Company, Ltd.'s Motion to Dismiss the Complaint Pursuant to Fed. R. Civ. P. 12(b)(2)* by Nissan Motor Company, Ltd.. (Attachments: # 1 Affidavit Declaration of Jacquelyn Trussell ISO Nissan Motor Company, Ltd.'s Motion to Dismiss Plaintiff's Complaint)(Amstutz, Paige) Modified on 2/12/2024 To edit text (bc). (Entered: 02/09/2024)
02/23/2024	24	Response in Opposition to Motion, filed by Foras Technologies Ltd., re 23 MOTION to Dismiss for Lack of Jurisdiction filed by Defendant Nissan Motor Company, Ltd. (Cooper, Brett) (Entered: 02/23/2024)
03/01/2024	25	RESPONSE in Support, filed by Nissan Motor Company, Ltd., re 23 MOTION to Dismiss for Lack of Jurisdiction filed by Defendant Nissan Motor Company, Ltd. *Defendant Nissan Motor Company Ltd.'s Reply in Support of Its Motion to Dismiss the Complaint Pursuant to Fed. R. Civ. P. 12(b)(2)* (Amstutz, Paige) (Entered: 03/01/2024)
03/13/2024	26	Order for parties to submit a Joint Proposed Scheduling Order; due by 3/27/2024. Signed by Judge Dustin M. Howell. (cc3) (Entered: 03/13/2024)
03/21/2024	27	RULE 7 DISCLOSURE STATEMENT filed by Nissan Motor Company, Ltd. identifying Corporate Parent None for Nissan Motor Company, Ltd.. (Amstutz, Paige) (Entered: 03/21/2024)
03/22/2024	28	CERTIFICATE OF SERVICE by Nissan Motor Company, Ltd. 27 Disclosure Statement (Rule 7) (Amstutz, Paige) (Entered: 03/22/2024)

03/22/2024	29	MOTION to Appear Pro Hac Vice by Paige Arnette Amstutz *for Wesley W. Jones* (Filing fee $ 100 receipt number ATXWDC-18562555) by on behalf of Nissan Motor Company, Ltd., ZF Friedrichshafen AG. (Attachments: # 1 Proposed Order)(Amstutz, Paige) (Entered: 03/22/2024)
03/22/2024	30	MOTION to Appear Pro Hac Vice by Paige Arnette Amstutz *for Binal J. Patel* (Filing fee $ 100 receipt number ATXWDC-18562636) by on behalf of Nissan Motor Company, Ltd., ZF Friedrichshafen AG. (Attachments: # 1 Proposed Order)(Amstutz, Paige) (Entered: 03/22/2024)
03/22/2024	31	MOTION to Appear Pro Hac Vice by Paige Arnette Amstutz *for John R. Hutchins* (Filing fee $ 100 receipt number ATXWDC-18562645) by on behalf of Nissan Motor Company, Ltd., ZF Friedrichshafen AG. (Attachments: # 1 Proposed Order)(Amstutz, Paige) (Entered: 03/22/2024)
03/27/2024	32	ORDER GRANTING 29 Motion to Appear Pro Hac Vice as to WESLEY W. JONES. Pursuant to our Administrative Policies and Procedures for Electronic Filing, the attorney hereby granted to practice pro hac vice in this case must register for electronic filing with our court within 10 days of this order. Registration is managed by the PACER Service Center Signed by Judge Robert Pitman. (cc3) (Entered: 03/27/2024)
03/27/2024	33	ORDER GRANTING 30 Motion to Appear Pro Hac Vice as to BINAL J. PATEL. Pursuant to our Administrative Policies and Procedures for Electronic Filing, the attorney hereby granted to practice pro hac vice in this case must register for electronic filing with our court within 10 days of this order. Registration is managed by the PACER Service Center Signed by Judge Robert Pitman. (cc3) (Entered: 03/27/2024)
03/27/2024	34	ORDER GRANTING 31 Motion to Appear Pro Hac Vice as to JOHN R. HUTCHINS. Pursuant to our Administrative Policies and Procedures for Electronic Filing, the attorney hereby granted to practice pro hac vice in this case must register for electronic filing with our court within 10 days of this order. Registration is managed by the PACER Service Center Signed by Judge Robert Pitman. (cc3) (Entered: 03/27/2024)
03/27/2024	35	NOTICE of Attorney Appearance by Ashley Marie Ratycz on behalf of Foras Technologies Ltd.. Attorney Ashley Marie Ratycz added to party Foras Technologies Ltd.(pty:pla) (Ratycz, Ashley) (Entered: 03/27/2024)
03/27/2024	36	Rule 26(f) Discovery Report/Case Management Plan - *Joint Rule 26 Report* by Foras Technologies Ltd.. (Cooper, Brett) (Entered: 03/27/2024)
03/27/2024	37	Joint MOTION for Entry of Scheduling Order Deadlines by Foras Technologies Ltd.. (Attachments: # 1 Exhibit A - Agreed Proposed Scheduling Order)(Cooper, Brett) (Entered: 03/27/2024)
04/02/2024	38	MOTION to Appear Pro Hac Vice by Paige Arnette Amstutz *Motion for Admission Pro Hac Vice Paul T. Qualey for Nissan Motor Company, Ltd. and ZF Friedrichshafen AG* (Filing fee $ 100 receipt number ATXWDC-18597104) by on behalf of Nissan Motor Company, Ltd., ZF Friedrichshafen AG. (Attachments: # 1 Proposed Order Proposed Order)(Amstutz, Paige) (Entered: 04/02/2024)
04/03/2024	39	ORDER GRANTING 38 Motion to Appear Pro Hac Vice as to PAUL T. QUALEY. Pursuant to our Administrative Policies and Procedures for Electronic Filing, the attorney hereby granted to practice pro hac vice in this case must register for electronic filing with our court within 10 days of this order. Registration is managed by the PACER Service Center Signed by Judge Robert Pitman. (cc3) (Entered: 04/03/2024)
04/09/2024	40	ORDER Setting Initial Pretrial Conference for 4/22/2024 01:30 PM before Judge Dustin M. Howell. Signed by Judge Dustin M. Howell. (bot2) (Entered: 04/09/2024)

04/10/2024	41	Opposed MOTION to Stay Case by Nissan Motor Company, Ltd., ZF Friedrichshafen AG. (Attachments: # 1 Affidavit of John R. Hutchins, # 2 Exhibit 1, # 3 Exhibit 2, # 4 Exhibit 3, # 5 Exhibit 4, # 6 Exhibit 5, # 7 Exhibit 6, # 8 Exhibit 7, # 9 Exhibit 8, # 10 Exhibit 9, # 11 Exhibit 10, # 12 Exhibit 11, # 13 Exhibit 12, # 14 Proposed Order Granting Motion to Stay)(Hutchins, John) (Entered: 04/10/2024)
04/15/2024	42	Unopposed MOTION for Extension of Time to File Response/Reply as to 41 Opposed MOTION to Stay Case by Foras Technologies Ltd.. (Attachments: # 1 Proposed Order) (Cooper, Brett) (Entered: 04/15/2024)
04/17/2024		Text Order GRANTING 42 Unopposed Motion for Extension of Time to File Response/Reply entered by Judge Robert Pitman. IT IS ORDERED that Plaintiff shall respond to the motion to stay on or before April 26, 2024. (This is a text-only entry generated by the court. There is no document associated with this entry.) (sh) (Entered: 04/17/2024)

PACER Service Center			
Transaction Receipt			
04/22/2024 13:10:37			
PACER Login:	apapageorgiou	**Client Code:**	foras
Description:	Docket Report	**Search Criteria:**	1:23-cv-00640-RP
Billable Pages:	6	**Cost:**	0.60

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/019,244	09/01/2023	7627781		7031

56436 7590 02/20/2024

Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528

EXAMINER	
BONSHOCK, DENNIS G	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
02/20/2024	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

GREENBERG TRAURIG (NJ)
500 CAMPUS DRIVE, SUITE 400
P.O. BOX 677
FLORHAM PARK NJ 07932

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/019,244* .

PATENT UNDER REEXAMINATION *7627781* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Office Action in Ex Parte Reexamination	**Control No.** 90/019,244	**Patent Under Reexamination** 7627781	
	Examiner DENNIS G BONSHOCK	**Art Unit** 3992	**AIA (FITF) Status** No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

a. ☐ Responsive to the communication(s) filed on _____.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

b. ☐ This action is made FINAL.

c. ☑ A statement under 37 CFR 1.530 has not been received from the patent owner.

A shortened statutory period for response to this action is set to expire 2 month(s) from the mailing date of this letter. Failure to respond within the period for response will result in termination of the proceeding and issuance of an *ex parte* reexamination certificate in accordance with this action. 37 CFR 1.550(d). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).** If the period for response specified above is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☐ Notice of References Cited by Examiner, PTO-892.
2. ☐ Information Disclosure Statement, PTO/SB/08.
3. ☐ Interview Summary, PTO-474.
4. ☐ _____.

Part II SUMMARY OF ACTION

1a. ☑ Claims 1-30 are subject to reexamination.
1b. ☐ Claims _____ are not subject to reexamination.
2. ☐ Claims _____ have been canceled in the present reexamination proceeding.
3. ☐ Claims _____ are patentable and/or confirmed.
4. ☑ Claims 1-30 are rejected.
5. ☐ Claims _____ are objected to.
6. ☐ The drawings, filed on _____ are acceptable.
7. ☐ The proposed drawing correction, filed on _____ has been (7a) ☐ approved (7b) ☐ disapproved.
8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the certified copies have
   1 ☐ been received.
   2 ☐ not been received.
   3 ☐ been filed in Application No. _____.
   4 ☐ been filed in reexamination Control No. _____.
   5 ☐ been received by the International Bureau in PCT application No. _____.
   * See the attached detailed Office action for a list of the certified copies not received.
9. ☐ Since the proceeding appears to be in condition for issuance of an *ex parte* reexamination certificate except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte* Quayle, 1935 C.D. 11, 453 O.G. 213.
10. ☐ Other: _____

cc: Requester (if third party requester)

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

Non-Final Office Action

This is a Non-Final action of claims 1-30 of United States Patent Number: 7,627,781 issued to Michaelis et al. (hereinafter the '781 Patent). Reexamination was ordered on 10/10/2023 responsive to the Request for *ex parte* reexamination filed 9/1/2023. Responsive to the Office ordering reexamination, the Patent Owner chose not to file a Patent Owner statement.

Extensions of time under 37 CFR 1.136(a) will not be permitted in these proceedings because the provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Additionally, 35 U.S.C. 305 requires that *ex parte* reexamination proceedings "will be conducted with special dispatch" (37 CFR 1.550(a)). Extensions of time in *ex parte* reexamination proceedings are provided for in 37 CFR 1.550(c).

References

- U.S. Patent No. 7,085,959 to Safford (hereinafter Safford)
- U.S. Patent No. 5,450,576 to Kennedy et al. (hereinafter Kennedy)
- U.S. Patent No. 7,251,746 to Fox (hereinafter Fox)

- U.S. Patent No. 7,328,371 to Kalyanasundharam et al. (hereinafter Kalyanasundharam)
- U.S. Patent Application Pub. No. 2004/0123201 to Nguyen et al. (hereinafter Nguyen)
- U.S. Patent No. 5,491,788 to Cepulis (hereinafter Cepulis)

Prosecution History

Original Examination:

The '781 Patent was filed as application 10/973,077 (hereinafter the '077 application) on October 25, 2004.

During the original prosecution of Michaelis patent, a Non-Final rejection was mailed on 9/20/2007. In this initial action the Office rejected the claims under obvious-type double patenting over claims 1-30 of US application 10/972,588 and further under 35 USC 102 over Safford et al. (US patent 7,085,959), while further rejecting other claims over a combination of Safford and Marshal (US patent 5,915,082).

On 12/12/2007, Patent Owner responded arguing specifically that: ***"In contrast, in Safford, transfer of the architected state of a processor is a transfer of information from one processor to another, as required for one processor to start processing instructions at the point that another processor was halted, while the system is running. Safford does not discuss transfer of system information (which processer is the boot processor) that remains permanent and survives a system reset. Safford discusses rebooting individual processors while a system is running (for example, column 4, lines 57-60, and claim3), but that is not the***

same as rebooting the entire system, which is initiated by the one designated boot processor after a system reset or power shutdown."

In the Office's 3/19/2008 Non-Final rejection Examiner Le maintained the Safford based rejections while further adding an additional grounds of rejection under 35 USC 103 over Fox et al. (US patent 7,251,749) in further view of Quach (US patent 6,625,749). There was some debate as to whether Safford was valid prior art, but Patent Owner (then Applicant) eventually admitted that Safford was eligible:

> NOTE: In applicant's reply to the previous office action, applicant asserted that Safford was not proper prior art under 35 U.S.C. § 103. However, that assertion was not correct, and is hereby retracted, with apologies for the mistake. Safford was published on January 8, 2004, before the filing of the present application on October 25, 2004.
>
> *File History* (Ex. 1002), p. 229

On 9/10/2008, Patent Owner filed a Terminal Disclaimer that was approved by the Office on 9/11/2008 and further amended the claims as follows:

1. *(currently amended) A system comprising:*

a plurality of processors;

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; and

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, wherein said another of said plurality of processors that is

assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said bool processor.

In the Office's 9/23/2008 Notice of Allowance, the claims were allowed apparently in view of the claim amendments of 9/10/2008, corresponding arguments presented, and discussion in an interview on 9/5/2008.

Litigation History

The '781 Patent is involved in the following litigation:

- Foras Technologies Ltd. v. Nissan Motor Company, LTD et al., Case No. 2:23-cv-00640 (W.D. Tex., June 6, 2023)
- Foras Technologies Ltd. v. Toyota Motor Company, Inc. et al., Case No. 2:23-cv-00321 (E.D. Tex., July 12, 2023)
- Foras Technologies Ltd. v. Toyota Motor Company, Inc. et al., Case No. 2:23-cv-00150 (E.D. Tex., April 5, 2023)

Claim Rejections

In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of

rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Rejection 1 – as proposed in Grounds 2 - Safford (Ex. 1004) in view of Kennedy (Ex. 1005) - (see Request pages 73-87 and in Appendix B)

Claims 1-6 and 9-15 are rejected under 35 U.S.C. 103(a) as being obvious over the combination of Safford (Ex. 1004) in view of Kennedy (Ex. 1005).

With regard to claim 1, teaching [pre] **A system comprising:,** Safford disclosed a computer system. For example, Safford disclosed that "a computer system employs lock stepped processor cores that operate in a master/check pair." Safford (Ex. 1004) at 2:55-57.;

With regard to claim 1, teaching [1.1] **a plurality of processors**, Safford disclosed a plurality of processors included in the computer system. For example, Safford taught that "a computer system employs lock stepped processor cores that operate in a master/check pair. Each of two processors in the pair processes the same

code sequences, and the resulting outputs of the processors are compared by a logic circuit located near external interfaces of the two processors." Safford (Ex. 1004) at 2:55-60. "For example, the node controller 130 may store the architected state of the processors 111, 113, 121, 123, 125, and 127, either internally in the node controller 130, or in another component of the computer system 100." Id. at 4:6-10. Accordingly, Safford disclosed a plurality of processors included in the computer system.

With regard to claim 1, teaching [1.2] **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**, Safford disclosed, or at least rendered obvious, this limitation by disclosing memory that stores the architected states of microprocessors, including the designation of a hot standby processor. For example, Safford taught that "the architected states of each of the microprocessors 22 and 24 is copied from the memory 25 into the microprocessors 22 and 24, respectively." Safford (Ex. 1004) at 2:2-4. "For example, the node controller 130 may store the architected state of the processors 111, 113, 121, 123, 125, and 127, either internally in the node controller 130, or in another component of the computer system 100." Id. at 4:6-10. "To improve reliability of processing assets, a computer system employs lock stepped processor cores that operate in a master/check pair." Id. at 2:55-57. "The processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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. at 4:25-30. Accordingly, Safford described assigning the roles of master and checker to certain processors and the role

of hot standby to another processor. See Wolfe Decl. (Ex. 1003) ¶187. Safford further described one pair of these processors "operating in a lock step mode" as initially "appear[ing] to [a] computer system...to be...a logical CPU 0" but later that logical CPU 0 designation is reassigned to another processor. See Safford (Ex. 1007) at 3:4-5, 4:52-53 ("The processor pair 125/127 then becomes the logical CPU 0 in the computer system 100'). A POSITA would have recognized Safford's logical CPU 0 as being the designation used by computing system 100 to identify its boot processor at least because, in multi-processor system architectures like Safford's, "logical CPU 0" was commonly used to indicate that a processor had been selected as a boot processor. See Wolfe Decl. (Ex. 1003) ¶ 187. Safford lacks an explicit disclosure identifying which processor was the boot processor, Safford as modified by Kennedy would have stored information about the assigned roles of boot processor and spare processor. See, e.g., Kennedy (Ex. 1005) at 7:5-13. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In view of the above citations, Safford taught data storage that stores the architected states, such as role assignments (e.g., master/checker), of paired processors, and Kennedy taught that such assignments could be the roles of boot processor and spare processor. See Wolfe Decl. (Ex. 1003) ¶ 187. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by storing information that assigns the roles of boot processor and spare processor in case of a reboot. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have

been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of spare processor to another one of the processors. Id.

With regard to claim 1, teaching [1.3] **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**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the failover of a processor to a hot standby processor in response to a loss of lockstep. For example, Safford taught that, "[u]sing the impending loss of lock step signal, the computer system 100 may continue operating (processing) using one of the processors 111 and 113 that did not experience an error." Safford (Ex. 1004) at 3:27-30; "The six processors 111, 113, 121, 123, 125, and 127 operate in lock step (i.e., are processing code sequences). For example, the processor 111 operates in lockstep with the processor 113, and the processor 121 operates in lock step with the processor 123 and the processor 125 operates in lock step with the processor 127." Id. at 4:19-24. "The processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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. at 4:25-30. "In block 210, the processor 111 signals the node controller 130 that the first processor pair 111/113 has broken lock step and that the first processor pair 111/113 should be taken 'off-line.'" Id. at 4:37-40. Accordingly, Safford described replacing a failed processor with a hot spare processor in response to a loss

of lockstep. See Wolfe Decl. (Ex. 1003) ¶ 188. The '781 Patent notes that a firmware component known as a processor abstraction layer (PAL) "copies all the needed information from memory into the internal registers of the hot spare processor so that the hot spare processor becomes a new active processor, transforming the role of the hot spare to boot processor." Ex. 1001 at 17:57-61. Safford disclosed this limitation through an identical process for transferring the role of boot processor to the spare processor. See Safford (Ex. 1004) at 32:29-32. For example, Safford taught "moving an architected state of the processor unit experiencing the loss of lock step to the spare processor unit, wherein the spare processor unit becomes an active processor unit in the computer system." Id. In view of these citations and the knowledge that a boot processor was a common component in computer systems (e.g., Safford's system), a POSITA would have understood Safford as disclosing the failover of a boot processor to a hot spare processor in response to a loss of lockstep. See Wolfe Decl. (Ex. 1003) ¶188. However, even if Safford did not explicitly identify the failed processor as having been the boot processor, Safford as modified by Kennedy would have performed boot processor failover. See, e.g., Kennedy (Ex. 1005) at 7:5-13. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Id. In view of the above citations, Safford taught logic that causes its system to replace a failed processor with a hot spare processor in response to a loss of lockstep, and Kennedy taught logic that causes its system to fail over the boot processor to an alternate processor. See Wolfe Decl. (Ex. 1003) ¶ 188. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by

providing fault tolerance and high availability in the event that the boot processor fails before a reboot occurs. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught logic operable for transferring the role of boot processor to the spare processor during system runtime in response to detecting loss of lockstep for the boot processor. Id.

With regard to claim 1, teaching [1.4] **wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the failover to a hot standby processor that is sitting idle until the loss of lockstep. For example, Safford taught that "[t]he processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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." Safford (Ex. 1004). at 4:25-30. Accordingly, Safford as modified by Kennedy taught that the processor assigned the role of spare is held in reserve until a loss of lockstep is detected.

With regard to claim 2, teaching **The system of claim 1 wherein said data storage is non-volatile data storage**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing memory that stores the architected states of microprocessors. For example, as explained above in connection with claim 1,

Safford as modified by Kennedy disclosed data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of spare processor to another one of the processors. Even if Safford did not explicitly identify the data storage as being non-volatile data storage, Safford as modified by Kennedy would have included non-volatile data storage that stores information about the assignments of the boot processor and the spare processor. For example, "[t]he CPU1 120 has a corresponding non-volatile memory 123 which contains configuration information and CPU specific initialize and self-test code (ISTC) for the CPU1 120[.]" Kennedy (Ex. 1005), 5:15-18. From these disclosures, a POSITA would have understood that the data storage that stores the information assigning the roles of boot processor and spare processor included non-volatile memory. See Wolfe Decl. (Ex. 1003) ¶ 189. Accordingly, Safford as modified by Kennedy would have rendered the subject matter of claim 2 obvious to a POSITA.

With regard to claim 3, teaching **The system of claim 1 further comprising: a user interface operable to receive input specifying said one of said plurality of processors to be assigned said role of boot processor,** Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the selection of a processor type to boot the system. In other words, even if Safford did not explicitly disclose a user interface that receives input specifying the processor assigned the role of boot processor, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA. For example, Kennedy taught selecting which processor among multiple processors is to boot the system. See Kennedy (Ex. 1005) at 4:33-34. Specifically, "other processor types may be selected to boot the system with a

corresponding change in the boot ROM 150, as well known in the art." Id. It was well known in the art that the act of selecting processor roles in computer systems (e.g., master/slave designations, distinct duty assignments or permissions in asymmetric multiprocessor configurations, etc.) is often performed via user interfaces. See Wolfe Decl. (Ex. 1003) 4 190. Based on this common knowledge in the art, a POSITA would have understood Kennedy's disclosure about selecting the processor to boot the system as including a boot processor selection made by input received via a user interface. Id. A POSITA would have been motivated to modify Safford by Kennedy and this common knowledge in the art to enable a user to select which processor is to boot the system via a user interface. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy and this common knowledge in the art would have rendered the subject matter of claim 3 obvious to a POSITA. Id.

With regard to claim 4, teaching **The system of claim 1 further comprising: a user interface operable to receive input specifying said another of said plurality of processors to be assigned said role of spare processor**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by implicitly disclosing the selection of a processor type that does not boot the system. In other words, even if Safford did not explicitly disclose a user interface that receives input specifying the processor assigned the role of spare processor, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA. For example, Kennedy taught selecting which processor among multiple processors is to boot the system. See

Kennedy (Ex. 1005) at 4:33-34. Specifically, "other processor types may be selected to boot the system with a corresponding change in the boot ROM 150, as well known in the art." Id. at 4:33-34. It was well known in the art that the act of selecting processor roles in computer systems (e.g., master/slave designations, distinct duty assignments or permissions in asymmetric multiprocessor configurations, etc.) is often performed via user interfaces. See Wolfe Decl. (Ex. 1003) ¶ 191. Based on this common knowledge in the art, a POSITA would have understood Kennedy's disclosure of selecting the processor to boot the system as including a spare processor selection (e.g., one of the alternate processors not selected as the boot processor) made by input received via a user interface. Id. A POSITA would have been motivated to modify Safford by Kennedy and this common knowledge in the art to enable a user to select, via a user interface, which processor is to serve as a spare to the boot processor. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy and this common knowledge in the art would have rendered the subject matter of claim 4 obvious to a POSITA. Id.

With regard to claim 5, teaching **The system of claim 1 wherein each of said plurality of processors is a processor module that includes a master processor and a slave processor**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the selection of a processor type to boot the system. For example, Safford taught that "a computer system employs lock stepped processor cores that operate in a master/check pair." Safford (Ex. 1004) at 2:55-57. "The processors operate as pairs when in lock step (i.e., the processors 111 and 113

are a pair; the processors 121 and 123 are a second pair ; and the processors 125 and 127 are a third pair). From the node controller's perspective, each pair of processors appears as a single (logical) processor." Id. at 3:57-63. Accordingly, Safford as modified by Kennedy taught a plurality of processor modules that each include a master processor and a slave processor. See Wolfe Decl. (Ex. 1003) ¶ 192.)

With regard to claim 6, teaching **The system of claim 1 wherein said another of said plurality of processors that is assigned said role of spare processor is an active spare before being transferred said role of boot processor**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a spare processor being active before a role transfer. For example, Safford taught that "the processor 111 signals the node controller 130 that the first processor pair 111/113 has broken lock step and that the first processor pair 111/113 should be taken 'off-line.'" Safford (Ex. 1004) at 4:37-40. "In block 230, the node controller 130 'reboots' the processors 111 and 113, and the processors 111 and 113 become the new 'hot standby' processor pair on the system 100." Id. at 4:57-60. Safford also taught that "the spare processor unit becomes an active processor unit in the computer system" Id., claim 1. In view of the above citations, a POSITA would have understood Safford as disclosing that a spare processor can be active before a role transfer. See Wolfe Decl. (Ex. 1003) ¶ 193. Moreover, a POSITA would have understood that the role at issue could be the boot processor assignment in view of Safford as modified by Kennedy for at least the same reasons provided above in connection with claim 1. Id. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by providing high availability through active

and/or inactive spare processors. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught that the spare processor is an active spare before being transferred the role of boot processor. Id.

With regard to claim 9, teaching [9.pre] **A method comprising:**, Safford disclosed a method. "An apparatus, operating on an advanced multi-core processor architecture, and a corresponding method, are used to enhance recovery from loss of lock step in a multi-processor computer system." Safford (Ex. 1004), Abstract

With regard to claim 9, teaching [9.1] **assigning a role of boot processor to one of a plurality of processors in a system**, Safford disclosed, or at least rendered obvious, this limitation by disclosing the architected states of microprocessors that operate as a master/checker pair as well as the designation of the logical CPU 0. For example, Safford taught that "the architected states of each of the microprocessors 22 and 24 is copied from the memory 25 into the microprocessors 22 and 24, respectively." Safford (Ex. 1004) at 2:2-4. "For example, the node controller 130 may store the architected state of the processors 111, 113, 121, 123, 125, and 127, either internally in the node controller 130, or in another component of the computer system 100." Id. at 4:6-10. "To improve reliability of processing assets, a computer system employs lock stepped processor cores that operate in a master/check pair." Id. at 2:55-57. Accordingly, Safford described assigning the roles of master and checker to certain processors. See Wolfe Decl. (Ex. 1003) ¶ 194. Safford further described one pair of these processors "operating in a lock step mode" as initially "appear[ing] to [a] computer

system...to be...a logical CPU 0" but later that logical CPU 0 designation is reassigned to another processor. See Safford (Ex. 1007) at 3:4-5, 4:52-53 ("The processor pair 125/127 then becomes the logical CPU 0 in the computer system 100"). A POSITA would have recognized Safford's logical CPU 0 as being the designation used by computing system 100 to identify its boot processor at least because, in multi-processor system architectures like Safford's, "logical CPU 0" was commonly used to indicate that a processor had been selected as a boot processor. See Wolfe Decl. (Ex. 1003) ¶ 194. However, even if Safford is interpreted as not explicitly identifying which processor was the boot processor, Safford as modified by Kennedy would have assigned the role of boot processor. See, e.g., Kennedy (Ex. 1005) at 7:5-13. For example, Kennedy taught that, "if this 'default' boot processor fails[,|" then an alternate processor "should take over boot operations to enhance availability of the system." Id. at 7:9-13. In view of the above citations, Safford taught assigning roles (e.g., master/checker) to paired processors, and Kennedy taught that one such assignment could be the role of boot processor. See Wolfe Decl. (Ex. 1003) ¶ 194. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by assigning the role of boot processor to a certain processor in case of a reboot. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught assigning a role of boot processor to one of a plurality of processors in a system. Id.

With regard to claim 9, teaching [9.2] **assigning a role spare processor to another of said plurality of processors**, Safford as modified by Kennedy disclosed, or

at least rendered obvious, this limitation by disclosing the designation of a hot standby processor. For example, Safford taught that "[t]he processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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." Safford (Ex. 1004) at 4:25-30. Accordingly, Safford as modified by Kennedy taught assigning a role of spare processor to another one of the plurality of processors in the system. Id.

With regard to claim 9, teaching [9.3] **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**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing failing over a processor to a hot standby processor in response to a loss of lockstep. For example, Safford taught that, "[u]sing the impending loss of lock step signal, the computer system 100 may continue operating (processing) using one of the processors 111 and 113 that did not experience an error." Safford (Ex. 1004) at 3:27-30. "The processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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. at 4:25-30. "In block 210, the processor 111 signals the node controller 130 that the first processor pair 111/113 has broken lock step and that the first processor pair 111/113 should be taken 'off-line.'" Id. at 4:37-40. Accordingly, Safford described replacing a failed processor with a hot spare processor in response to a loss of lockstep. See Id. at 3:27-30, 4:37-54. The '781 Patent notes that a firmware

component known as a processor abstraction layer (PAL) "copies all the needed information from memory into the internal registers of the hot spare processor so that the hot spare processor becomes a new active processor, transforming the role of the hot spare to boot processor." Ex. 1001 at 17:57-61. Safford disclosed this claim limitation through an identical process for transferring the role of boot processor to the spare processor. Safford (Ex. 1004) at 32:29-32. For example, Safford taught "moving an architected state of the processor unit experiencing the loss of lock step to the spare processor unit, wherein the spare processor unit becomes an active processor unit in the computer system." Id. In view of this disclosure and the knowledge that a boot processor was a common component in computer systems (e.g., Safford's system), a POSITA would have understood Safford as disclosing the failover of a boot processor to a hot spare processor in response to a loss of lockstep. See Wolfe Decl. (Ex. 1003) ¶ 195. However, even if Safford is not interpreted as explicitly identifying the failed processor as having been the boot processor, Safford as modified by Kennedy would have performed boot processor failover. See, e.g., Kennedy (Ex. 1005) at 7:5-13. In view of the above citations, Safford taught replacing a failed processor with a hot spare processor in response to a loss of lockstep, and Kennedy taught failing over the boot processor to an alternate processor. See Wolfe Decl. (Ex. 1003) ¶ 195. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by providing high availability in the event that the boot processor fails before a reboot occurs. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id.

Accordingly, Safford as modified by Kennedy taught changing, during system runtime, the assigned role of boot processor to the spare processor in response to loss of lockstep for the boot processor. Id.

With regard to claim 9, teaching [9.4] **responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the improved availability of processor assets to avoid a computer system crash and by disclosing a central boot ROM that is used by the operating system and stores the boot code executed in conjunction with the operating system to boot the system. For example, in focusing on "improv[ing] availability of the processor assets of the computer system", Safford notes that "loss of lock step, if not promptly corrected, may cause the computer system 18 to crash." Safford (Ex. 1004) at 3:21-22, 1:45-47. Safford lacks an explicit disclosure of an operating system that recognizes the spare processor as the boot processor without requiring shutdown, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA. For example, according to Kennedy, "a central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53-55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. A POSITA would have understood that, because Kennedy's central boot ROM 150 is used by the operating system and stores the boot code executed in conjunction with the operating

system to boot the system, the operating system would recognize the spare processor as the boot processor after failover without requiring shutdown of the operating system, especially if Kennedy is to achieve its objective of providing fault tolerance and high availability. See Kennedy (Ex. 1005) at 9:1-5, 9:53-55, 10:42-45; see also Wolfe Decl. (Ex. 1003) ¶ 196. This conclusion of a POSITA's understanding is also supported by Safford's objective of avoiding system crashes. See Safford (Ex. 1004) at 3:21-22, 1:45-47. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to improve the reliability of Safford's system by providing fault tolerance and high availability in the event that the boot processor fails before a reboot occurs. See Wolfe Decl. (Ex. 1003) ¶ 196. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy would have rendered obvious to a POSITA an operating system that recognizes the spare processor as the boot processor without requiring shutdown. Id.

With regard to claim 10, **Claim 10 recites subject matter that is substantially similar to claim 2** and is therefore rendered obvious for the same reasons as claim 2 above.

With regard to claim 11, **Claim 11 recites subject matter that is substantially similar to claim 2** and is therefore rendered obvious for the same reasons as claim 2 above.

With regard to claim 12, teaching **The method of claim 9 further comprising:** [12.1] **determining at least one processor that is coupled to a common communication bus with said one of said plurality processors assigned to the**

role of boot processor, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a determinization that the alternate processor is installed in one of the slots of a backplane bus and that the boot processor is installed in another one of the slots of the backplane bus. Safford does not explicitly disclose determining a processor that is coupled to a common communication bus with the boot processor, however Safford as modified by Kennedy would have determined that the alternate processor and the boot processor are coupled to the same backplane bus. For example, according to Kennedy, "the system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. In view of these citations, Kennedy taught that the boot processor and the alternate boot processors are installed in specific slots along the same backplane bus. Based on this disclosure from Kennedy, a POSITA would have understood that any alternate processor installed in one of the slots of the backplane bus would have been coupled to a common communication bus with the boot processor. See Wolfe Decl. (Ex. 1003) ¶ 197. As a result, any form of identifying such an alternate processor would constitute a determination of an alternate processor coupled to a common communication bus with the boot processor. Id. A POSITA would have been motivated to combine these disclosures from Kennedy with Safford to improve the communication capabilities of Safford's system for fault tolerance and high availability. Id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this

combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught this claim limitation. Id.

With regard to claim 12, teaching [12.2] **selecting said another of said plurality of processors to be assigned the role of spare processor from said determined at least one processor**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the selection of one or more alternate (e.g., spare) boot processors by installing the alternate boot processor into a specific slot corresponding to the alternate boot role on the backplane bus. Even if Safford did not explicitly disclose selecting the processor coupled to the common communication bus as the spare processor, Safford as modified by Kennedy would have selected the alternate processor installed in the same backplane bus as the boot processor to be the spare. For example, according to Kennedy, "the system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. A POSITA would have been motivated to combine these disclosures from Kennedy with Safford to improve the reliability of Safford's system by providing fault tolerance and high availability in the event that the boot processor fails before a reboot occurs. See Wolfe Decl. (Ex. 1003) ¶ 198. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy taught this claim limitation. Id.

With regard to claim 13, teaching **The method of claim 9 further comprising: during system boot-up, system firmware identifying the spare processor and holding said spare processor in reserve as a hot spare**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a BIOS responsible for identifying a spare processor and holding the same in reserve as a hot spare. Safford taught identifying a spare processor and holding the spare processor in reserve as a hot spare for at least the same reasons provided above in connection with claim 1. Even if Safford did not explicitly disclose doing so via system firmware, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA. For example, according to Kennedy, "central boot ROM 150 comprises a read-only-memory with the basic input/output system (BIOS) instruction set for an INTEL 80486 or 80386 microprocessor. Accordingly, in the present embodiment, at least one CPU connected to the 64-bit bus is, or emulates, an INTEL 80486 or 80386 microprocessor in order to boot the system." Kennedy (Ex. 1005) at 4:25-33. "[A] central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53-55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 199. In view of the above citations and this common knowledge, a POSITA would have understood that Kennedy's BIOS constitutes firmware responsible for identifying a spare processor and holding the same in reserve as a hot spare. Id. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to achieve cost savings by

performing these tasks with firmware instead of using additional hardware. See id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy would have rendered the subject matter of claim 13 obvious to a POSITA. Id.

With regard to claim 14, teaching **The method of claim 9 further comprising: during system boot-up, system firmware booting the spare processor as an active spare**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a BIOS responsible for booting the spare processor as an active spare. Safford taught booting the spare processor as an active spare for at least the same reasons provided above in connection with claim 6. Even if Safford did not explicitly disclose doing so via system firmware, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA. For example, according to Kennedy, "central boot ROM 150 comprises a read-only-memory with the basic input/output system (BIOS) instruction set for an INTEL 80486 or 80386 microprocessor. Accordingly, in the present embodiment, at least one CPU connected to the 64-bit bus is, or emulates, an INTEL 80486 or 80386 microprocessor in order to boot the system." Kennedy (Ex. 1005) at 4:25-33. "[A] central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53-55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 200. In view of the above citations

and this common knowledge, a POSITA would have understood that Kennedy's BIOS constitutes firmware responsible for booting the spare processor as an active spare. Id. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to enable Safford's system to dynamically modify the activity of spare processors (e.g., between active spare and hot spare) to increase efficiency and/or throughput. See id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy would have rendered the subject matter of claim 14 obvious to a POSITA. Id.

With regard to claim 15, teaching **The method of claim 9 wherein system firmware performs said changing operation**, Safford as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a BIOS included in an IOSM responsible for changing the role of boot processor to the spare processor. Safford taught performing the changing operation for at least the same reasons provided above in connection with claim 9. Even if Safford did not explicitly disclose doing so via system firmware, Safford as modified by Kennedy would have rendered this claim limitation obvious to a POSITA. For example, Kennedy taught that each CPU includes a slot selection (sometimes referred to as "SS") logic for selecting which CPU is designated as the boot processor. "In general, the slot select logic 148 on the default boot CPU allows this CPU to boot." Kennedy (Ex. 1005) at 10:29-31, FIG. 3. Kennedy's input/output services module ("IOSM") is responsible for changing the role of boot processor to the spare processor in conjunction with the CPUs' slot selection logic. See Id. at 6:36-38, FIG. 3. For example, "[a]ny other slot selection circuits on CPUs installed

in the system have a corresponding slot selection line from the IOSM 108." Id. As demonstrated by FIG. 3 of Kennedy reproduced below, Kennedy's IOSM includes the BIOS.

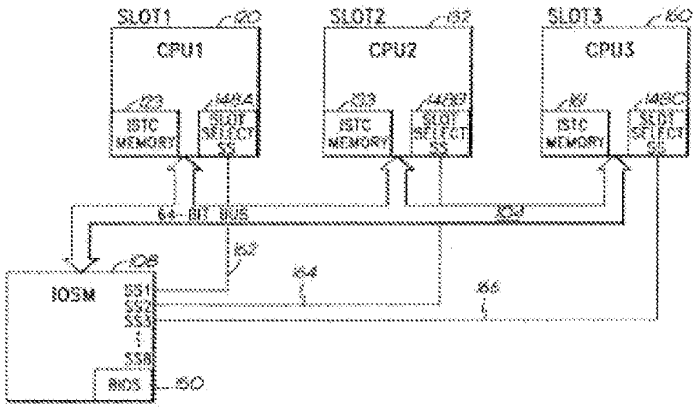


FIG. 3

The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 201. In view of the above citations and this common knowledge, a POSITA would have understood that that Kennedy's BIOS, which constitutes firmware, is responsible for changing the role of boot processor to the spare processor in conjunction with the CPUs' slot selection logic. Id. A POSITA would have been motivated to combine these disclosures from Safford and Kennedy to achieve cost savings by performing the change operation with firmware instead of using additional hardware. See id. It was well within the skill of a POSITA to modify Safford's system by Kennedy in this manner, and thus this combination of Safford and Kennedy would have been expected to succeed. Id. Accordingly, Safford as modified by Kennedy would have rendered the subject matter of claim 15 obvious to a POSITA.

Rejection 2 – as proposed in Ground 3 - Kennedy (Ex. 1005) in view of Nguyen (Ex. 1008) and Fox (Ex. 1006)) - (see Request pages 88-106 and in Appendix C)

Claims 1-30 are rejected under 35 U.S.C. 103(a) as being obvious over the combination of Kennedy (Ex. 1005) in view of Nguyen (Ex. 1008) and Fox (Ex. 1006).

With regard to claim 1, teaching [1.pre] **A system comprising:,** Kennedy disclosed a system. For example, Kennedy taught "multiprocessor computer systems." Kennedy (Ex. 1005) at 1:14-15.

With regard to claim 1, teaching [1.1] **a plurality of processors**, Kennedy disclosed this limitation by disclosing multiprocessor computer systems that enable a boot processor to fail over to an alternate processor. For example, Kennedy taught "multiprocessor computer systems." Kennedy (Ex. 1005) at 1:14-15. Kennedy's disclosure "relates to a method and apparatus to automatically control the determination of which processor in the system will boot the system." Id. at 1:15-17. Kennedy further disclosed that, "[i]f the boot processor fails, control automatically shifts to an alternate processor." Id. at 1:17-19. Accordingly, Kennedy taught a plurality of processors.

With regard to claim 1, teaching [1.2] **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**, Kennedy disclosed, or at least rendered obvious, this limitation by disclosing memories

that hold configuration information about processors that are assigned the roles of boot processor and spare processor. For example, Kennedy taught that "[t]he CPU ISTC memories hold configuration information." Kennedy (Ex. 1005) at 7:33-35. Kennedy also taught that, "[a]s previously explained, the ISTC memory associated with each CPU contains...configuration information about the CPU. Thus, the boot CPU determines how many CPUs are present on the bus." Id. at 9:16-21. Kennedy further taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. In view of the above citations, Kennedy taught memories that store CPU configuration information that includes the roles of boot processor and alternate processor in the system. See Wolfe Decl. (Ex. 1003) ¶ 204. Accordingly, Kennedy taught data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of spare processor to another one of the processors. Id.

With regard to claim 1, teaching [1.3] **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**, Kennedy as modified by Nguyen disclosed, or at least rendered obvious, this limitation by disclosing the transfer of the boot processor role to the alternate processor in response to a loss of lockstep. For example, Kennedy taught that, "if this 'default'? boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system."

Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. Even if Kennedy did not explicitly identify the boot processor's failure as being a loss of lockstep, Kennedy as modified by Nguyen would have detected a loss of lockstep. See, e.g., Nguyen (Ex. 1008) ¶ [0008], [0024], [0054]-[0055]. For example, Nguyen taught that "[e]xecution cores 120 operate in lock step." Nguyen (Ex. 1008) ¶ [0024]. "FRC- hardware compares results generated by each core, and if a discrepancy is detected, the FRC system passes control to an error-handling routine." Id. ¶ [0006]. "However, for FRC-enabled processors, such errors are likely to be manifested as FRC errors, since they take the execution cores out of lock-step." Id. ¶ [0008]. "[E]ach execution core may include a status bit that is set if an error is detected. The master execution core may check this bit to determine if it has also generated an error." Id. ¶ [0050]. Accordingly, Nguyen described detecting a loss of lockstep. In view of the above citations, Kennedy taught logic that causes its system to replace the boot processor with an alternate processor in response to a failure of the boot processor, and Nguyen taught detecting an error (i.e., a loss of lockstep). See Wolfe Decl. (Ex. 1003) ¶ 205. A POSITA would have been motivated to combine these disclosures from Kennedy and Nguyen to improve the reliability of Kennedy's system by providing fault tolerance and high availability in the event that the boot processor fails through a loss of lockstep. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen in this manner, and thus this combination of Kennedy and Nguyen would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen taught logic

operable for transferring the role of boot processor to the spare processor during system runtime in response to detecting loss of lockstep for the boot processor. Id.

With regard to claim 1, teaching [1.4] **wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the transfer of the boot processor role to the spare processor in response to detecting loss of lockstep and by disclosing the automatic failover to a hot spare processor. For example, Kennedy as modified by Nguyen taught a spare processor that assumes the role of boot processor in response to a loss of lockstep for at least the reasons provided above. Even if Kennedy did not explicitly identify the spare processor as having been a hot spare processor that is held in reserve, Kennedy as modified by Nguyen and Fox would have transferred the role of boot processor to a hot spare processor. See, e.g., Fox (Ex. 1006), Abstract. For example, according to Fox, "[t]he invention provides a substantially seamless transfer of executing processes from the failing processor to the replacement processor, without halting the execution of the operating system (OS) and/or applications on the overall MP. That is, a failing processor is able to automatically fail-over to a hot-spare processor without affecting the OS or executing applications." Fox (Ex. 1006) at 4:16-23. In view of the above citations, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, transferring the role of boot processor to a hot spare processor during system runtime in response to detecting loss of lockstep for the boot processor. See Wolfe Decl. (Ex. 1003) ¶ 206. A POSITA would have been motivated to

combine these disclosures from Kennedy, Nguyen, and Fox to decrease delays in failing over from the boot processor to the spare processor. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught that the processor assigned the role of spare is held in reserve as a hot spare processor until the loss of lockstep is detected for the boot processor. Id.

With regard to claim 2, teaching **The system of claim 1 wherein said data storage is non-volatile data storage**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing non-volatile memory that stores the configuration information in the system. For example, as explained above in connection with claim 1, Kennedy as modified by Nguyen and Fox taught data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of spare processor to another one of the processors. Kennedy further taught that "[t]he CPU1 120 has a corresponding non-volatile memory 123 which contains configuration information and CPU specific initialize and self-test code (ISTC) for the CPU1 120." Kennedy (Ex. 1005), 5:15-18. From these disclosures, a POSITA would have understood that the data storage that stores the information assigning the roles of boot processor and spare processor included non-volatile memory. See Wolfe Decl. (Ex. 1003) ¶ 207. Accordingly, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 2 obvious to a POSITA. Id.

With regard to claim 3, teaching **The system of claim 1 further comprising: a user interface operable to receive input specifying said one of said plurality of processors to be assigned said role of boot processor**, Kennedy as modified by Nguyen, Fox, and common knowledge in the art rendered obvious this limitation by disclosing the selection of a processor type to boot the system. In other words, even if Kennedy did not explicitly disclose a user interface that receives input specifying the processor assigned the role of boot processor, Kennedy as modified by Nguyen, Fox, and common knowledge in the art would have rendered this claim limitation obvious to a POSITA. For example, Kennedy taught selecting which processor among multiple processors is to boot the system. See Kennedy (Ex. 1005) at 4:33-34. Specifically, "other processor types may be selected to boot the system with a corresponding change in the boot ROM 150, as well known in the art." Id. It was well known in the art that the act of selecting processor roles in computer systems (e.g., master/slave designations, distinct duty assignments or permissions in asymmetric multiprocessor configurations, etc.) is often performed via user interfaces. See Wolfe Decl. (Ex. 1003) ¶ 208. Based on this common knowledge in the art, a POSITA would have understood Kennedy's disclosure of selecting the processor to boot the system as including a boot processor selection made by input received via a user interface. Id. A POSITA would have been motivated to modify Kennedy by Nguyen, Fox, and this common knowledge in the art to enable a user to select which processor is to boot the system via a user interface. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen, Fox, and the common knowledge in this manner, and thus this combination of Kennedy, Nguyen, Fox, and the common knowledge would have been expected to

succeed. Id. Accordingly, Kennedy as modified by Nguyen, Fox, and this common knowledge in the art would have rendered the subject matter of claim 3 obvious to a POSITA. Id.

With regard to claim 4, teaching **The system of claim 1 further comprising: a user interface operable to receive input specifying said another of said plurality of processors to be assigned said role of spare processor**, Kennedy as modified by Nguyen, Fox, and common knowledge in the art rendered obvious this limitation by implicitly disclosing the selection of a processor type that does not boot the system. In other words, even if Kennedy did not explicitly disclose a user interface that receives input specifying the processor assigned the role of spare processor, Kennedy as modified by Nguyen, Fox, and common knowledge in the art would have rendered this claim limitation obvious to a POSITA. For example, Kennedy taught selecting which processor among multiple processors is to boot the system. See Kennedy (Ex. 1005) at 4:33-34. Specifically, "other processor types may be selected to boot the system with a corresponding change in the boot ROM 150, as well known in the art." Id. at 4:33-34. It was well known in the art that the act of selecting processor roles in computer systems (e.g., master/slave designations, distinct duty assignments or permissions in asymmetric multiprocessor configurations, etc.) is often performed via user interfaces. See Wolfe Decl. (Ex. 1003) ¶ 209. Based on this common knowledge in the art, a POSITA would have understood Kennedy's disclosure of selecting the processor to boot the system as including a spare processor selection (e.g., one of the alternate processors not selected as the boot processor) made by input received via a user interface. Id. A POSITA would have been motivated to modify Kennedy by Nguyen, Fox,

and this common knowledge in the art to enable a user to select, via a user interface, which processor is to serve as a spare to the boot processor. Id. A POSITA would have been motivated to modify Kennedy by Nguyen, Fox, and this common knowledge in the art to enable a user to select, via a user interface, which processor is to serve as a spare to the boot processor. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen, Fox, and the common knowledge in this manner, and thus this combination of Kennedy, Nguyen, Fox, and the common knowledge would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen, Fox, and this common knowledge in the art would have rendered the subject matter of claim 4 obvious to a POSITA. Id.

With regard to claim 5, teaching **The system of claim 1 wherein each of said plurality of processors is a processor module that includes a master processor and a slave processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing multicore processors that each include master and slave execution cores. Even if Kennedy did not explicitly disclose that each of its processors is a processor module that includes master and slave processors, Kennedy as modified by Nguyen and Fox would have incorporated processor modules that each include master and slave execution cores. For example, Nguyen "relates to microprocessors and, in particular, to mechanisms for handling errors in FRC-enabled processors." Nguyen (Ex. 1008) ¶ [0002]. Nguyen also taught "a mechanism for handling recoverable and non-recoverable errors in multi-core processors." Id. ¶ [0058]. "Processor 110 includes first and second execution cores 120(a), 120(b)." Id. ¶ [0020]. "For one embodiment, one of execution cores may be designated as the master core

and the other as the slave core when processor 10 is operating in FRC mode." Id. ¶ [0047]. In view of the above citations, Kennedy as modified by Nguyen and Fox taught multicore processor modules that each include master and slave execution cores. See Wolfe Decl. (Ex. 1003) ¶ 210. A POSITA would have been motivated to combine these disclosures from Kennedy, Nguyen, and Fox to improve the spatial efficiency of Kennedy's system by consolidating the amount of space occupied by the processors. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught that each processor is a processor module including a master processor and a slave processor.

With regard to claim 6, teaching **The system of claim 1 wherein said another of said plurality of processors that is assigned said role of spare processor is an active spare before being transferred said role of boot processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the spare as active before the role of boot processor is transferred. Even if Kennedy did not explicitly disclose that the alternate processor is an active spare before being transferred the role of boot processor, Kennedy as modified by Nguyen and Fox would have included the alternate processor as an active spare before being transferred the role of boot processor. For example, Fox's BIOS "brings the standby processor out of the held-off state and into an active state." Fox (Ex. 1006) at 7:8-10. In view of this disclosure, Kennedy as modified by Nguyen and Fox taught that the alternate processor is an active spare before being transferred the role of boot processor. See Wolfe Decl.

(Ex. 1003) ¶ 211. A POSITA would have been motivated to combine these disclosures from Kennedy, Nguyen, and Fox to improve the usage and/or efficiency of the processors in the system. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught that the spare is active before being transferred the role of boot processor. Id.

With regard to claim 7, teaching **The system of claim 1 further comprising: an operating system, wherein, responsive to the logic transferring the role of boot processor to the spare processor, the operating system recognizes the spare processor as the boot processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the failover to a hot spare processor without affecting the OS or executing applications. Even if Kennedy did not explicitly disclose an operating system that recognizes the spare processor as the boot processor after the role transfer, Kennedy as modified by Nguyen and Fox would have rendered this claim limitation obvious to a POSITA. For example, Fox taught that "[t]he invention provides a substantially seamless transfer of executing processes from the failing processor to the replacement processor, without halting the execution of the operating system (OS) and/or applications on the overall MP. That is, a failing processor is able to automatically fail-over to a hot-spare processor without affecting the OS or executing applications." Fox (Ex. 1006) at 4:16-25. Fox also taught that, "once the spare processor has been successfully brought on line, the BIOS (table) is updated and the OS [and] bus controllers...are messaged to change their configurations to reflect a

processor change from the failing processor to the spare processor." Id. at 8:10-14. "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., Abstract, 3:28-31, 6:64-67. In view of the above citations, a POSITA would have understood that, because Fox's failing processor automatically fails over to a hot spare processor without affecting the operating system and then control is returned to the operating system which updates the "active" processor list, the operating system would recognize the spare processor as having assumed the role of the failing processor after the role transfer. See Wolfe Decl. (Ex. 1003) ¶ 212. Or at the very least, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 7 obvious to a POSITA due at least in part to Fox's failing processor automatically being replaced by a hot spare processor without affecting the operating system and then having control returned to the operating system which updates the "active" processor list. Id. It was well within the skill of a POSITA to modify Nguyen's system by Kennedy and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id.

With regard to claim 8, teaching **The system of claim 1 wherein said plurality of processors are Itanium Processor Family (IPF) processors**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing IPF processors. Even if Kennedy did not explicitly disclose that its processors are IPF processors, Kennedy as modified by Nguyen and Fox would have included IPF processors. For example, Nguyen taught that "[o]ne embodiment of the recovery mechanism employs code that is closely associated with the processor. For example,

the Itanium Processor Family of Intel Corporation employs a layer of firmware called processor abstraction layer (PAL), which provides an abstraction of the processor to the rest of the computing system." Nguyen (Ex. 1008) ¶ [0045]. In view of the above citations, Kennedy as modified by Nguyen and Fox taught that the processors would have been IPF processors. See Wolfe Decl. (Ex. 1003) 7 213. A POSITA would have been motivated to combine these disclosures from Kennedy, Nguyen, and Fox to improve the spatial efficiency of the system by consolidating the processors into the multicore processor package used by the IPF. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught this claim limitation.

With regard to claim 9, teaching [9.pre] **A method comprising:**, Kennedy disclosed a method. For example, Kennedy taught "a method and apparatus to automatically control the determination of which processor in the system will boot the system." Kennedy (Ex. 1005) at 1:15-17.

With regard to claim 1, teaching [9.1] **assigning a role of boot processor to one of a plurality of processors in a system**, Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a processor that is assigned the role of boot processor in a multiprocessor system. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at

2:17-21. Kennedy also taught "multiprocessor computer systems." Id. at 1:14-15. In view of the above citations, Kennedy taught assigning the role of boot processor to one of the processors in the system. See Wolfe Decl. (Ex. 1003) ¶ 214. Accordingly, Kennedy taught assigning the role of boot processor to one of the processors in the system. Id.

With regard to claim 9, teaching [9.2] **assigning a role spare processor to another of said plurality of processors**, Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a processor that is assigned the role of alternate processor in a multiprocessor system. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. Kennedy also taught "multiprocessor computer systems." Id. at 1:14-15. In view of the above citations, Kennedy taught assigning the role of alternate processor to one of the processors in the system. See Wolfe Decl. (Ex. 1003) ¶ 215. Accordingly, Kennedy taught assigning the role of spare processor to one of the processors in the system. Id.

With regard to claim 9, teaching [9.3] **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**, Kennedy as modified by Nguyen disclosed, or at least rendered obvious, this limitation by disclosing the replacement of the boot processor with the alternate processor in response to a loss of lockstep for the

boot processor. For example, Kennedy taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. Even if Kennedy did not explicitly identify the boot processor's failure as being a loss of lockstep, Kennedy as modified by Nguyen would have detected a loss of lockstep. See, e.g., Nguyen (Ex. 1008) ¶ [0008], [0024], [0054]-[0055]. For example, Nguyen taught that "[e]xecution cores 120 operate in lock step." Nguyen (Ex. 1008) ¶ [0024]. "FRC-hardware compares results generated by each core, and if a discrepancy is detected, the FRC system passes control to an error-handling routine." Id. ¶ [0006]. "However, for FRC-enabled processors, such errors are likely to be manifested as FRC errors, since they take the execution cores out of lock-step." Id. ¶ [0008]. "[E]ach execution core may include a status bit that is set if an error is detected. The master execution core may check this bit to determine if it has also generated an error." Id. ¶ [0050]. Accordingly, Nguyen described detecting a loss of lockstep. In view of the above citations, Kennedy taught replacing the boot processor with an alternate processor in response to a failure of the boot processor, and Nguyen taught detecting an error (i.e., a loss of lockstep). See Wolfe Decl. (Ex. 1003) ¶ 216. A POSITA would have been motivated to combine these disclosures from Kennedy and Nguyen to improve the reliability of Kennedy's system by providing fault tolerance and high availability in the event that the boot processor fails through a loss of lockstep. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen in this manner, and thus this combination of

Kennedy and Nguyen would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen taught changing the assignment of the role of boot processor to the spare processor during system runtime in response to detecting loss of lockstep for the boot processor. Id.

With regard to claim 9, teaching [9.4] **responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the failover to a hot spare processor without affecting the OS or executing applications. Even if Kennedy did not explicitly disclose an operating system that recognizes the spare processor as the boot processor after the role transfer, Kennedy as modified by Nguyen and Fox would have rendered this claim limitation obvious to a POSITA. For example, Fox taught that "[t]he invention provides a substantially seamless transfer of executing processes from the failing processor to the replacement processor, without halting the execution of the operating system (OS) and/or applications on the overall MP. That is, a failing processor is able to automatically fail-over to a hot-spare processor without affecting the OS or executing applications." Fox (Ex. 1006), 4:16-25. Fox also taught that, "once the spare processor has been successfully brought on line, the BIOS (table) is updated and the OS [and] bus controllers...are messaged to change their configurations to reflect a processor change from the failing processor to the spare processor." Id. 8:10-14. "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., Abstract, 3:28-31, 6:64-67. In view of these citations, a POSITA would have understood that, because Fox's failing processor automatically fails over to a hot spare processor without affecting the operating system and then control is returned to the operating system which updates the "active" processor list, the operating system would recognize the spare processor as having assumed the role of the failing processor after failover. See Wolfe Decl. (Ex. 1003) ¶ 217. Or at the very least, Kennedy as modified by Nguyen and Fox would have rendered this subject matter obvious to a POSITA due at least in part to Fox's failing processor automatically failing over to a hot spare processor without affecting the operating system or applications. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id.

With regard to claim 10, **Claim 10 recites subject matter that is substantially similar to Claim 2** and is therefore rendered obvious for the same reasons as Claim 2 above.

With regard to claim 11, **Claim 11 recites subject matter that is substantially similar to Claim 2** and is therefore rendered obvious for the same reasons as Claim 2 above.

With regard to claim 12, teaching **The method of claim 9 further comprising: [12.1] determining at least one processor that is coupled to a common communication bus with said one of said plurality processors assigned to the role of boot processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least

rendered obvious, this limitation by disclosing the processors assigned the roles of boot processor and spare processor as being coupled to a common communication bus. For example, Kennedy taught that "the system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. Based on this disclosure from Kennedy, a POSITA would have understood that any alternate processor installed in one of the slots of the backplane bus would have been coupled to a common communication bus with the boot processor. See Wolfe Decl. (Ex. 1003) ¶ 218. As a result, any form of identifying such an alternate processor would constitute a determination of an alternate processor coupled to a common communication bus with the boot processor. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught this claim limitation. Id.

With regard to claim 12, teaching [12.2] **selecting said another of said plurality of processors to be assigned the role of spare processor from said determined at least one processor**, Kennedy as modified by Nguyen and Fox taught assigning the role of spare to a certain processor for at least the reasons provided above in connection with claim 1. Moreover, Kennedy as modified by Nguyen and Fox taught determining that the spare processor is coupled to the common communication bus for at least the same reasons provided above in connection with this claim. Accordingly, Kennedy as modified by Nguyen and Fox taught selecting a processor coupled to the common communication bus to be assigned the role of spare. See Wolfe Decl. (Ex. 1003) ¶ 219.

With regard to claim 13, teaching **The method of claim 9 further comprising: during system boot-up, system firmware identifying the spare processor and holding said spare processor in reserve as a hot spare**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing a BIOS responsible for identifying a spare processor and holding the same in reserve as a hot spare. As explained above in connection with claim 1, Kennedy as modified by Nguyen and Fox taught that the processor assigned the role of spare is held in reserve as a hot spare processor until the loss of lockstep is detected for the boot processor. Moreover, according to Kennedy, "central boot ROM 150 comprises a read-only-memory with the basic input/output system (BIOS) instruction set for an INTEL 80486 or 80386 microprocessor. Accordingly, in the present embodiment, at least one CPU connected to the 64-bit bus is, or emulates, an INTEL 80486 or 80386 microprocessor in order to boot the system." Kennedy (Ex. 1005) at 4:25-33. "[A] central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53-55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 220. In view of the above citations and this common knowledge, a POSITA would have understood that Kennedy's BIOS constitutes firmware responsible for identifying a spare processor and holding the same in reserve as a hot spare. Id. Accordingly, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 13 obvious to a POSITA. Id.

With regard to claim 14, teaching **The method of claim 9 further comprising: during system boot-up, system firmware booting the spare processor as an active spare**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing a BIOS responsible for booting the spare processor as an active spare. As explained above in connection with claim 6, Kennedy as modified by Nguyen and Fox taught that that the spare is active before being transferred the role of boot processor. Moreover, according to Kennedy, "central boot ROM 150 comprises a read-only-memory with the basic input/output system (BIOS) instruction set for an INTEL 80486 or 80386 microprocessor. Accordingly, in the present embodiment, at least one CPU connected to the 64-bit bus is, or emulates, an INTEL 80486 or 80386 microprocessor in order to boot the system." Kennedy (Ex. 1005) at 4:25-33. "[A] central system boot ROM 150 accessible by all processors...may be used by the operating system." Kennedy (Ex. 1005) at 9:1-5. "[T]he CPU1 120 controls the system again and begins the operating system boot process." Id. at 9:53- 55. "The alternate boot processors...can execute the boot code in the central boot ROM 150". Id. at 10:42-45. The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 221. In view of the above citations and this common knowledge, a POSITA would have understood that Kennedy's BIOS constitutes firmware responsible for booting the spare processor as an active spare. Id. Accordingly, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 14 obvious to a POSITA. Id.

With regard to claim 15, teaching **The method of claim 9 wherein system firmware performs said changing operation**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing a BIOS included

in an IOSM responsible for changing the role of boot processor to the spare processor. As explained above in connection with claim 9, Kennedy as modified by Nguyen and Fox taught performing the changing operation. Moreover, Kennedy taught that each CPU includes a slot selection (sometimes referred to as "SS") logic for selecting which CPU is designated as the boot processor. "In general, the slot select logic 148 on the default boot CPU allows this CPU to boot." Kennedy (Ex. 1005) at 10:29-31, FIG. 3. Kennedy's input/output services module (""IOSM") is responsible for changing the role of boot processor to the spare processor in conjunction with the CPUs' slot selection logic. See id. at 6:36-38, FIG. 3. For example, "[a]ny other slot selection circuits on CPUs installed in the system have a corresponding slot selection line from the IOSM 108." Id. As demonstrated by FIG. 3 of Kennedy reproduced below, Kennedy's IOSM includes the BIOS.

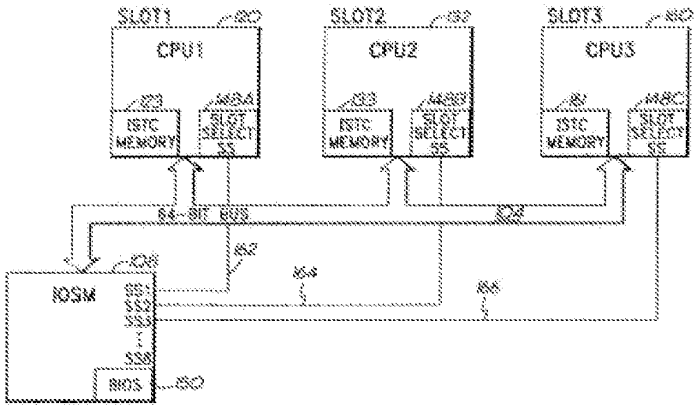


FIG. 3

The BIOS was a very well-known example of firmware. Wolfe Decl. (Ex. 1003) ¶ 222. In view of the above citations and this common knowledge, a POSITA would have

understood that that Kennedy's BIOS, which constitutes firmware, is responsible for changing the role of boot processor to the spare processor in conjunction with the CPUs' slot selection logic. Id. Accordingly, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 15 obvious to a POSITA. Id.

With regard to claim 16, teaching [16.pre**] A system comprising:**, Kennedy disclosed a system. For example, Kennedy taught "multiprocessor computer systems." Kennedy (Ex. 1005) at 1:14-15.

With regard to claim 16, teaching [16.1] **a plurality of lockstep processor modules that each includes a master processor and a slave processor that is in lockstep with said master processor**, Kennedy as modified by Nguyen disclosed, or at least rendered obvious, this limitation by disclosing multicore processors that each include master and slave execution cores in lockstep. For example, Kennedy taught "multiprocessor computer systems." Kennedy (Ex. 1005) at 1:14-15. Kennedy's disclosure "relates to a method and apparatus to automatically control the determination of which processor in the system will boot the system." Id. at 1:15-17. Kennedy further disclosed that, "[i]f the boot processor fails, control automatically shifts to an alternate processor." Id. at 1:17-19. Even if Kennedy did not explicitly disclose that each of its processors is a processor module that includes master and slave processors in lockstep, Kennedy as modified by Nguyen would have incorporated processor modules that each include master and slave execution cores in lockstep. For example, Nguyen "relates to microprocessors and, in particular, to mechanisms for handling errors in FRC- enabled processors." Nguyen (Ex. 1008) ¶[[0002]. Nguyen also taught "a mechanism for handling recoverable and non-recoverable errors in multi-core

processors." Id. ¶ [0058]. "Processor 110 includes first and second execution cores 120(a), 120(b)." Id. ¶ [0020]. "For one embodiment, one of execution cores may be designated as the master core and the other as the slave core when processor 10 is operating in FRC mode." Id. ¶ [0047]. "Execution cores 120 operate in lock step." Id. at ¶ [0024]. "FRC-hardware compares results generated by each core, and if a discrepancy is detected, the FRC system passes control to an error-handling routine." Id. ¶ [0006]. "However, for FRC-enabled processors, such errors are likely to be manifested as FRC errors, since they take the execution cores out of lock-step." Id. ¶ [0008]. In view of the above citations, Kennedy as modified by Nguyen taught multicore processor modules that each include master and slave execution cores in lockstep. See Wolfe Decl. (Ex. 1003) ¶ 223. A POSITA would have been motivated to combine these disclosures from Kennedy and Nguyen to improve the spatial efficiency of Kennedy's system by consolidating the amount of space occupied by the processors. Id. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen in this manner, and thus this combination of Kennedy and Nguyen would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen taught a plurality of lockstep processor modules that each include a master processor and a slave processor in lockstep with each other. Id.

With regard to claim 16, teaching [16.2] **a first communication bus with which at least a first two of said plurality of lockstep processor modules are communicatively coupled**, Kennedy as modified by Nguyen disclosed, or at least rendered obvious, this limitation by disclosing the processors assigned the roles of boot processor and spare processor as being coupled to a common communication bus. For

example, Kennedy taught that "the system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. In view of these citations, Kennedy as modified by Nguyen taught that the boot processor and the alternate processor are installed in slots on the backplane bus. See Wolfe Decl. (Ex. 1003) ¶ 224. In other words, Kennedy's backplane bus is communicatively coupled to the boot processor and alternate processor. See Id. Accordingly, Kennedy as modified by Nguyen taught a first communication bus with which at least a first two of the plurality of lockstep processor modules are communicatively coupled. Id.

With regard to claim 16, teaching [16.3] **a second communication bus with which at least a second two of said plurality of lockstep processor modules are communicatively coupled**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the processors as being interconnected with each other via system buses, processor-to-processor buses, and control buses. As explained above in connection with this claim, Kennedy taught a backplane bus that is communicatively coupled to the boot processor and alternate processor. Even if Kennedy did not explicitly disclose a second communication bus with which at least a second two of the lockstep processor modules are communicatively coupled, Kennedy as modified by Nguyen and Fox would have rendered this claim limitation obvious to a POSITA. For example, Fox taught that "[a] system bus 110 connects the memory controller 107 and other components to IA-32 processors 120, 122, via bridges 112, and processor buses 114. The IA-32 processors 120, 122 and the

bridges 112 are grouped at nodes 111." Fox (Ex. 1006) at 5:9-12, FIG. 1. "In one alternate implementation, processors 120, 122 are interconnected to each other via processor-to-processor buses 203." Id. at 5:41-43. "Processors 120, 122 are also connected to bridge (bus controller) 207 via individual control buses 205." Id. at 5:43-45. In view of these citations, Kennedy as modified by Nguyen and Fox taught a plurality of processors that are interconnected to each over via two or more different buses (e.g., system bus 110, processor-to-processor buses 203, and individual control buses 205). See Wolfe Decl. (Ex. 1003) ¶ 225. As further demonstrated by FIG. 1 of Fox reproduced below, Fox's system includes various processors that are communicatively coupled to one another via multiple communication buses.

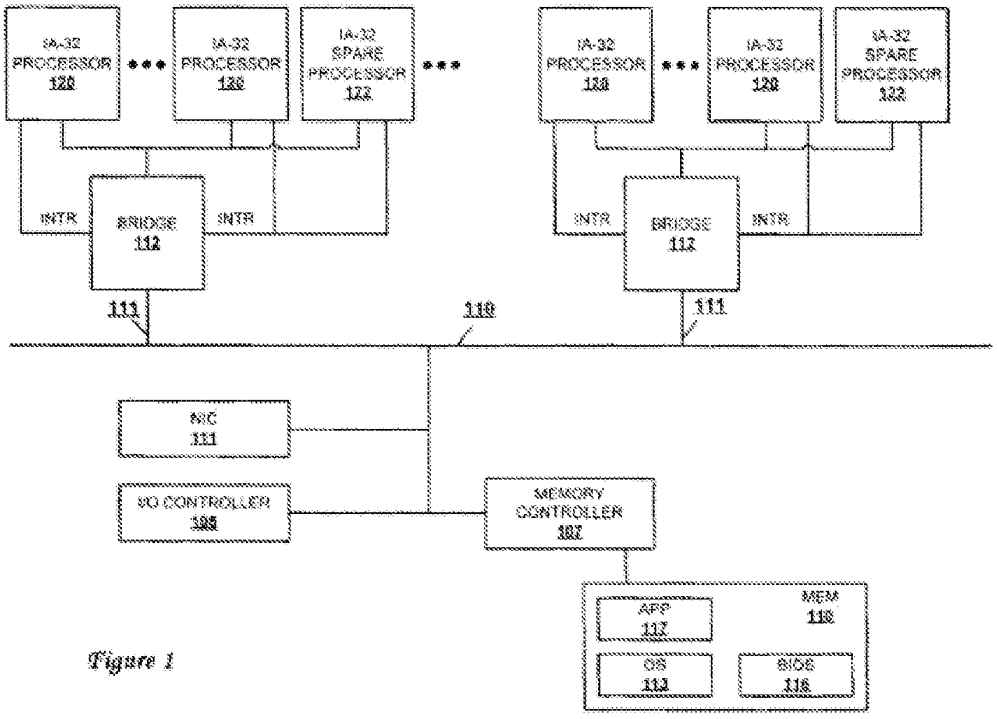


Figure 1

A POSITA would have been motivated to combine these disclosures from Kennedy and Nguyen and Fox to provide multiple communication channels and/or mechanisms for communicating with the processors. See Wolfe Decl. (Ex. 1003) ¶ 225. It was well within the skill of a POSITA to modify Kennedy's system by Nguyen and Fox in this manner, and thus this combination of Kennedy, Nguyen, and Fox would have been expected to succeed. Id. Accordingly, Kennedy as modified by Nguyen and Fox taught a second communication bus with which at least a second two of the plurality of lockstep processor modules are communicatively coupled. Id.

With regard to claim 16, teaching [16.4] **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, wherein said spare processor is communicatively coupled to the same one of said first and second communication buses as said boot processor,** Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing memories that hold configuration information about processors that are assigned the roles of boot processor and spare processor and communicatively coupled to the same communication bus. For example, Kennedy taught that "[t]he CPU ISTC memories hold configuration information." Kennedy (Ex. 1005) at 7:33-35. Kennedy also taught that, "[a]s previously explained, the ISTC memory associated with each CPU contains...configuration information about the CPU. Thus, the boot CPU determines how many CPUs are present on the bus." Id. at 9:16-21. Kennedy further taught that, "if this 'default' boot processor fails[,]" then an alternate processor "should take over boot operations to enhance availability of the system." Kennedy (Ex. 1005) at 7:9-13. In

addition, Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21. Kennedy's "system has a centrally accessible memory, a default boot central processing unit installed in a first slot on the backplane bus, and may also have one or more alternate boot central processing units installed in a second slot on the backplane bus." Kennedy (Ex. 1005) at 2:34-38. In view of the above citations, Kennedy as modified by Nguyen and Fox taught memories that store CPU configuration information that includes the roles of boot processor and alternate processor, which are assigned to processors coupled to the same communication bus. See Wolfe Decl. (Ex. 1003) ¶ 226. Accordingly, Kennedy as modified by Nguyen and Fox taught data storage storing information that assigns the role of boot processor to one of the processors and assigns the role of Spare processor to another one of the processors, wherein the boot processor and the spare processor are coupled to the same communication bus. Id.

With regard to claim 17, **Claim 17 recites subject matter that is substantially similar to claim 15**, which depends from claim 9, and is therefore rendered obvious for the same reasons as claim 15 in view of claim 9 above.

With regard to claim 18, **Claim 18 recites subject matter that is substantially similar to claim 15**, which depends from claim 9, and is therefore rendered obvious for the same reasons as claim 15 in view of claim 9 above.

With regard to claim 19, **Claim 19 recites subject matter that is substantially similar to claim 3** and is therefore rendered obvious for the same reasons as claim 3 above.

With regard to claim 20, **Claim 20 recites subject matter that is substantially similar to claim 4** and is therefore rendered obvious for the same reasons as claim 4 above.

With regard to claim 21, teaching **The system of claim 20 wherein said user interface presents for selection as said spare processor at least one processor that is communicatively to the same one of said first and second communication buses as said boot processor**, Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by implicitly disclosing the selection of a processor type that does not boot the system. For example, Kennedy as modified by Nguyen and Fox taught a spare processor that is communicatively coupled to the same communication bus as the boot processor for at least the same reasons provided above in connection with claim 16. Kennedy as modified by Nguyen and Fox also rendered obvious user interfaces that facilitate selecting the spare processor for at least the same reasons provided above in connection with claim 4. Accordingly, for at least the same reasons provided above in connection with claims 4 and 16, Kennedy as modified by Nguyen and Fox would have rendered the subject matter of claim 21 obvious to a POSITA. See Wolfe Decl. (Ex. 1003) ¶ 227.).

With regard to claim 22, **Claim 22 recites subject matter that is substantially similar to claim 13** and is therefore rendered obvious for the same reasons as claim 13 above.

With regard to claim 23, **Claim 23 recites subject matter that is substantially similar to claim 14** and is therefore rendered obvious for the same reasons as claim 14 above.

With regard to claim 24, **Claim 24 recites subject matter that is substantially similar to claim 8** and is therefore rendered obvious for the same reasons as claim 8 above.

With regard to claim 25, **Claim 25 recites subject matter that is substantially similar to claim 16** and is therefore rendered obvious for the same reasons as claim 16 above.

With regard to claim 26, **Claim 26 recites subject matter that is substantially similar to claim 18** and is therefore rendered obvious for the same reasons as claim 18 above.

With regard to claim 27, **Claim 27 recites subject matter that is substantially similar to claims 3** and 19 and is therefore rendered obvious for the same reasons as claims 3 and 19 above.

With regard to claim 28, **Claim 28 recites subject matter that is substantially similar to claims 3** and 19 and is therefore rendered obvious for the same reasons as claims 3 and 19 above.

With regard to claim 29, **Claim 29 recites subject matter that is substantially similar to claims 3** and 19 and is therefore rendered obvious for the same reasons as claims 3 and 19 above.

With regard to claim 30, **Claim 30 recites subject matter that is substantially similar to the combinations of claims 13 and 15** and is therefore rendered obvious for the same reasons as claims 13 and 15 above.

Summary

Claims 1-30 are REJECTED.

Conclusion

Extensions of time under 37 CFR 1.136(a) do not apply in reexamination proceedings. The provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Further, in 35 U.S.C. 305 and in 37 CFR 1.550(a), it is required that reexamination proceedings "will be conducted with special dispatch within the Office."

In order to ensure full consideration of any amendments, affidavits or declarations, or other documents as evidence of patentability, such documents must be submitted in response to this Office action. Submissions after the next Office action, which is intended to be a final action, will be governed by the requirements of 37 CFR 1.116, after final rejection and 37 CFR 41.33 after appeal, which will be strictly enforced.

The patent owner is reminded of the continuing responsibility under 37 CFR 1.565(a) to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving the patent throughout the course of this reexamination proceeding. The requester is also reminded of the ability to similarly appraise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP § § 2207, 2282, and 2286.

All correspondence relating to this ex parte reexamination proceeding should be directed:

Electronically: Registered users may submit via Patent Center at https://patentcenter.uspto.gov/.

By Mail to: Mail Stop *Ex Parte* Reexam
Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to: (571) 273-9900
Central Reexamination Unit

By hand: Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

For electronic transmissions, 37 CFR 1.8(a)(1)(i)(C) and (ii) states that correspondence (except for a request for reexamination and a corrected or replacement request for reexamination) will be considered timely filed if (a) it is transmitted via the USPTO patent electronic filing system in accordance with 37 CFR 1.6(a)(4), and (b) includes a certificate of transmission for each piece of correspondence stating the date of transmission, which is prior to the expiration of the set period of time in the Office action.

Any inquiry concerning this communication should be directed to Central Reexamination Unit at telephone number (571) 272-7705.

/DENNIS G BONSHOCK/
Primary Examiner, Art Unit 3992

Conferees:

/ADAM L BASEHOAR/
Primary Examiner, Art Unit 3992

/ALEXANDER J KOSOWSKI/
Supervisory Patent Examiner, Art Unit 3992

Reexamination	Application/Control No.	Applicant(s)/Patent Under Reexamination
	90/019,244	7627781
	Certificate Date	Certificate Number

Requester Correspondence Address: ☐ Patent Owner ☑ Third Party
GREENBERG TRAURIG (NJ) 500 CAMPUS DRIVE, SUITE 400 P.O. BOX 677 FLORHAM PARK NJ 07932

LITIGATION REVIEW ☑	/DGB/ (examiner initials)	03 October 2023 (date)
Case Name		Director Initials
Case No. 2:23-cv-00640 (W.D. Tex., June 6, 2023) Foras Technologies Ltd . v. Nissan Motor Company, LTD et al.		
Case No. 2:23-cv-00321 (E.D. Tex., July 12, 2023) Foras Technologies Ltd . v. Toyota Motor Company, Inc. et al.		
Case No. 2:23-cv-00150 (E.D. Tex., April 5, 2023) Foras Technologies Ltd. v. Toyota Motor Company, Inc. et al.		

COPENDING OFFICE PROCEEDINGS	
TYPE OF PROCEEDING	NUMBER
NONE	

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	90/019,244	7627781
	Examiner	Art Unit
	DENNIS G BONSHOCK	3992

CPC - Searched*		
Symbol	Date	Examiner

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
Prosecution History	02/01/2024	/DGB/
Litigation History	02/01/2024	/DGB/
PDP	02/01/2024	/DGB/
P-TACTS	02/01/2024	/DGB/

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/019,244	09/01/2023	7627781		7031

56436 7590 10/10/2023

Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528

EXAMINER	
BONSHOCK, DENNIS G	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
10/10/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

GREENBERG TRAURIG (NJ)
500 CAMPUS DRIVE, SUITE 400
P.O. BOX 677
FLORHAM PARK NJ 07932

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/019,244* .

PATENT UNDER REEXAMINATION *7627781* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Order Granting Request For Ex Parte Reexamination	Control No. 90/019,244	Patent Under Reexamination 7627781	
	Examiner DENNIS G BONSHOCK	Art Unit 3992	AIA (FITF) Status No

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

The request for *ex parte* reexamination filed 09/01/2023 has been considered and a determination has been made. An identification of the claims, the references relied upon, and the rationale supporting the determination are attached.

Attachments: a)☐ PTO-892, b)☑ PTO/SB/08, c)☐ Other: ____

1. ☑ The request for *ex parte* reexamination is GRANTED.

RESPONSE TIMES ARE SET AS FOLLOWS:

For Patent Owner's Statement (Optional): TWO MONTHS from the mailing date of this communication (37 CFR 1.530 (b)). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).**

For Requester's Reply (optional): TWO MONTHS from the **date of service** of any timely filed Patent Owner's Statement (37 CFR 1.535). **NO EXTENSION OF THIS TIME PERIOD IS PERMITTED.** If Patent Owner does not file a timely statement under 37 CFR 1.530(b), then no reply by requester is permitted.

cc:Requester (if third party requester)

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

DECISION ON REQUEST FOR REEXAMINATION

A substantial new question (SNQ) of patentability affecting claims 1-30 of United States Patent Number: 7,627,781 issued to Michaelis et al. (hereinafter the '781 Patent) is raised by the Request for *ex parte* reexamination filed 9/1/2023.

The Office's determination in both the order for reexamination and the examination stage of the reexamination ordered will generally be limited solely to a review of the claim(s) for which reexamination was requested (see: MPEP 2240 & 2243). Additionally, if a Requester fails to set forth the pertinency and manner of applying the cited art to a claim as required by 37 CFR 1.510(b), that claim will generally not be reexamined. This matter was squarely addressed in *Sony Computer Entertainment America Inc. v. Dudas*, 85 USPQ2d 1594 (E.D. Va 2006). The District Court upheld the Office's discretion to not reexamine claims in an *Inter Partes* Reexamination proceeding other than those claims for which reexamination had specifically been requested.

Information Disclosure Statement

Regarding Information Disclosure Statement (IDS) submissions, MPEP 2256 recites the following: "Where patents, publications, and other such items of information

are submitted by a party (patent owner or requester) in compliance with the requirements of the rules, the requisite degree of consideration to be given to such information will be normally limited by the degree to which the party filing the information citation has explained the content and relevance of the information. The initials of the examiner placed adjacent to the citations on the form PTO/SB/08A and 08B or its equivalent, without an indication to the contrary in the record, do not signify that the information has been considered by the examiner any further than to the extent noted above."

Accordingly, the IDS submission filed by Third Party Requester (3PR) on 9/1/2023 has been considered by the Examiner only with the scope required by MPEP 2256, unless otherwise noted.

References

The Request presents that the following references raise an SNQ with regard to claims 1-30 of the '781 Patent:

- U.S. Patent No. 7,085,959 to Safford (hereinafter Safford)
- U.S. Patent No. 5,450,576 to Kennedy et al. (hereinafter Kennedy)
- U.S. Patent No. 7,251,746 to Fox (hereinafter Fox)
- U.S. Patent No. 7,328,371 to Kalyanasundharam et al. (hereinafter Kalyanasundharam)
- U.S. Patent Application Pub. No. 2004/0123201 to Nguyen et al. (hereinafter Nguyen')

- U.S. Patent No. 5,491,788 to Cepulis (hereinafter Cepulis)

Prosecution History

Original Examination:

The '781 Patent was filed as application 10/973,077 (hereinafter the '077 application) on October 25, 2004.

During the original prosecution of Michaelis patent, a Non-Final rejection was mailed on 9/20/2007. In this initial action the Office rejected the claims under obvious-type double patenting over claims 1-30 of US application 10/972,588 and further under 35 USC 102 over Safford et al. (US patent 7,085,959), while further rejecting other claims over a combination of Safford and Marshal (US patent 5,915,082).

On 12/12/2007, Patent Owner responded arguing specifically that: ***"In contrast, in Safford, transfer of the architected state of a processor is a transfer of information from one processor to another, as required for one processor to start processing instructions at the point that another processor was halted, while the system is running. Safford does not discuss transfer of system information (which processer is the boot processor) that remains permanent and survives a system reset. Safford discusses rebooting individual processors while a system is running (for example, column 4, lines 57-60, and claim3), but that is not the same as rebooting the entire system, which is initiated by the one designated boot processor after a system reset or power shutdown."***

In the Office's 3/19/2008 Non-Final rejection Examiner Le maintained the Safford based rejections while further adding an additional grounds of rejection under 35 USC

103 over Fox et al. (US patent 7,251,749) in further view of Quach (US patent 6,625,749). There was some debate as to whether Safford was valid prior art, but Patent Owner (then Applicant) eventually admitted that Safford was eligible:

> NOTE: In applicant's reply to the previous office action, applicant asserted that Safford was not proper prior art under 35 U.S.C. § 103. However, that assertion was not correct, and is hereby retracted, with apologies for the mistake. Safford was published on January 8, 2004, before the filing of the present application on October 25, 2004.
>
> *File History* (Ex. 1002), p. 229

On 9/10/2008, Patent Owner filed a Terminal Disclaimer that was approved by the Office on 9/11/2008 and further amended the claims as follows:

1. *(currently amended) A system comprising:*

a plurality of processors;

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; and

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, wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said bool processor.

In the Office's 9/23/2008 Notice of Allowance, the claims were allowed apparently in view of the claim amendments of 9/10/2008, corresponding arguments presented, and discussion in an interview on 9/5/2008.

Litigation History

The '781 Patent is involved in the following litigation:

- Foras Technologies Ltd. v. Nissan Motor Company, LTD et al., Case No. 2:23-cv-00640 (W.D. Tex., June 6, 2023)
- Foras Technologies Ltd. v. Toyota Motor Company, Inc. et al., Case No. 2:23-cv-00321 (E.D. Tex., July 12, 2023)
- Foras Technologies Ltd. v. Toyota Motor Company, Inc. et al., Case No. 2:23-cv-00150 (E.D. Tex., April 5, 2023)

Substantial New Question of Patentability

The presence or absence of "a substantial new question of patentability" determines whether or not reexamination is ordered. For a substantial new question of patentability to be present, it is only necessary that: (A) the prior art patents and/or printed publications raise a substantial question of patentability regarding at least one claim, i.e., the teaching of the (prior art) patents and printed publications is such that a reasonable examiner would consider the teaching to be important in deciding whether or not the claim is patentable; and (B) the same question of patentability as to the claim has not been decided by the Office in an earlier concluded examination or review of the patent, raised to or by the Office in a pending reexamination or supplemental

examination of the patent, or decided in a final holding of invalidity (after all appeals) by a federal court in a decision on the merits involving the claim. If a reexamination proceeding was terminated/vacated without resolving the substantial question of patentability question, it can be re-presented in a new reexamination request. It is not necessary that a "prima facie" case of unpatentability exist as to the claim in order for a substantial new question of patentability to be present as to the claim. Thus, a substantial new question of patentability as to a patent claim could be present even if the examiner would not necessarily reject the claim as either fully anticipated by, or obvious in view of, the prior art patents or printed publications (see: MPEP 2242(I)).

In a decision to order reexamination made on or after November 2, 2002, reliance on old art does not necessarily preclude the existence of a substantial new question of patentability that is based exclusively on that old art. See Public Law 107-273, 116 Stat. 1758, 1899-1906 (2002), which expanded the scope of what qualifies for a substantial new question of patentability upon which a reexamination may be based. Determinations on whether a substantial new question of patentability exists in such an instance shall be based upon a fact-specific inquiry done on a case-by-case basis. For example, a substantial new question of patentability may be based solely on old art where the old art is being presented/viewed in a new light, or in a different way, as compared with its use in the earlier examination(s), in view of a material new argument or interpretation presented in the request. Such material new argument or interpretation may be based solely on claim scope of the patent being reexamined (see: MPEP 2242(II)(A)).

Claim 1 is presented below italicized showing the limitations that are believed to be the critical limitations, given the prosecution history recited above, and which are used by the Examiner to show how specific teachings of the proposed references raise a substantial new question of patentability.

1. (currently amended) A system comprising:

a plurality of processors;

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; and

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, wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor.

In the 9/1/2023 Request, the 3PR (Requester) suggests that the following combination of references provide elements which are allegedly equivalent to claims 1-30 of the '781 Patent.

Proposed Ground 1: Claims 1-6 and 9-15 over a combination of Kalyanasundharam (Ex. 1007) in view of Kennedy (Ex. 1005)

The Requestor alleges that the combination of the Kalyanasundharam and Kennedy references raises a substantial new question of patentability with respect to claim 1 of the '781 patent. Reading of the Kalyanasundharam and Kennedy references on the claims is provided on pages 16-31 and in Exhibit AA of the Request.

Each of Kalyanasundharam and Kennedy references are new art that provide new, non-cumulative technological teachings that were not previously considered nor discussed on the record during prosecution of the '781 patent.

With regard to the limitation of **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,** Kalyanasundharam discloses transferring of the role of primary processor to an auxiliary processor in response to a mismatch involving the primary processor. Kalyanasundharam further teaches lockstep execution of processors in noting that "the processor cores 18A-18B may execute in lockstep (that is, may execute the same instructions on the same clock cycle)." Kalyanasundharam (Ex. 1007) at 4:15-18. Kalyanasundharam further teaches recognizing a loss of lockstep (LOL), where "If the masked comparison results indicate that the

communication from the primary processor core mismatches with all auxiliary processor cores (that are enabled)...the control circuit 84 may disable the primary processor core and select one of the auxiliary processor cores to be the primary processor core (block 102)." Id. at 13:15- 22. "As with the discussion above for disabling the auxiliary processor core, the control circuit 83 may simply disable the primary processor core from comparison by clearing the enable bit in the configuration register 42, or may disable the primary processor core from executing at all." Id. at 13:22- 27. "The control circuit 84 may also update the PC field of the configuration register 42 to reflect the newly selected primary processor core." Id. at 13:27-29.

Kennedy supplements in the teaching that, "if this 'default' boot processor fails[,|" then an alternate processor "should take over boot operations to enhance availability of the system." (see Kennedy (Ex. 1005) at 7:5-13).

With regard to the limitation of **wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor,** Kalyanasundharam teaches that "a fixed or programmable number of the processor cores (less than the total number in the node) may participate in redundant execution." Kalyanasundharam (Ex. 1007) at 11:35-38. "The configuration register 42 may further include a set of core enables which indicate which cores are participating in redundant execution mode." Id. at 12:12-15. "The control circuit 84 may...optionally disable a processor core or cores from comparison in the redundant execution mode." Id. at 12:29-32; "Additionally, the core enables in the configuration register 42 may be

used to selectively enable various pairs for redundant execution." Id. at 15:7-10. "If the primary processor core is programmably selected, a different processor core may be selected as the primary processor core and the previous primary processor core may be disabled." Id. at 11:45-48. "For example, servers are implemented on fault-tolerant systems so that the servers may be available as close to 24 hours a day as possible. Any mission-critical computing functions may benefit from being assigned to fault-tolerant computer systems." Id. at 1:14-19.

Though Kalyanasundharam does not explicitly use the term "hot spare" or the phrase "held in reserve" in connection with its processors, Kalyanasundharam described its processors as having the same functionality in that Kalyanasundharam teaches selectively enabling processors for participation, selectively disabling processors from participation, and maintaining configuration register 42 that tracks which processors are participating and which processors are not participating.

It is agreed that the combination of the Kalyanasundharam and Kennedy references does raise a substantial new question of patentability, as proposed in the request, to at least claim 1 of the '781 patent. There is a substantial likelihood that a reasonable Examiner would consider these teachings important in deciding whether or not this claim is patentable.

Accordingly, the combination of the Kalyanasundharam and Kennedy references does raise a substantial new question to at least claim 1, which question has not been

decided in a previous examination of the '781 patent nor was there a final holding of invalidity by the Federal Courts regarding the '781 patent.

Proposed Grounds 2 and 4: Claims 1-6 and 9-15 over a combination of Safford (Ex. 1004) in view of Kennedy (Ex. 1005) / Claims 1 and 6 over Safford (Ex. 1004)

The Requestor alleges that the combination of the Safford and Kennedy references, as well as Safford alone, raise a substantial new question of patentability with respect to claim 1 of the '781 patent. Reading of the Safford and Kennedy references on the claims is provided on pages 31-42 and in Exhibit BB of the Request.

Safford was previously utilized in the Original Examination of the '077 application, and is therefore old art, however, in attempting to raise a substantial new question of patentability the requester proposes combining the teaching of Safford with the Kennedy reference. The Kennedy references is new art that provide new, non-cumulative technological teachings that were not previously considered nor discussed on the record during prosecution of the '781 patent. Though Safford was present in the original prosecution as discussed above, the combination of Safford and Kennedy has not been considered against the claims of record. Thus, the Safford references is being presented here in a new light.

With regard to the limitation of **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,** Safford teaches that, "[u]sing the impending loss of lock step signal, the computer system 100 may continue operating (processing) using one of the processors 111 and 113 that did not experience an error." Safford (Ex. 1004) at 3:27-30. "The six processors 111, 113, 121, 123, 125, and 127 operate in lock step (i.e., are processing code sequences). For example, the processor 111 operates in lockstep with the processor 113, and the processor 121 operates in lock step with the processor 123 and the processor 125 operates in lock step with the processor 127." Id. at 4:19-24. "The processor 125 may be designated as a 'hot standby,' and is sitting idle in lock step mode with the processor 127. 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. at 4:25-30. "In block 210, the processor 111 signals the node controller 130 that the first processor pair 111/113 has broken lock step and that the first processor pair 111/113 should be taken 'off-line.'" Id. at 4:37-40. Safford taught "moving an architected state of the processor unit experiencing the loss of lock step to the spare processor unit, wherein the spare processor unit becomes an active processor unit in the computer system." Id.

Kennedy supplements in the teaching that, "if this 'default' boot processor fails[,|" then an alternate processor "should take over boot operations to enhance availability of the system." (see Kennedy (Ex. 1005) at 7:5-13).

With regard to the limitation of **wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor,** 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. 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." Safford (Ex. 1004). at 4:25-30.

The Examiner doesn't believe Safford alone raises a substantial new question of patentability, as it lacks the boot features claimed, but agrees that the combination of the Safford and Kennedy references does raise a substantial new question of patentability, as proposed in the request, to at least claim 1 of the '781 patent. There is a substantial likelihood that a reasonable Examiner would consider these teachings important in deciding whether or not this claim is patentable.

Accordingly, the combination of the Safford and Kennedy references does raise a substantial new question to at least claim 1, which question has not been decided in a previous examination of the '781 patent nor was there a final holding of invalidity by the Federal Courts regarding the '781 patent.

Proposed Ground 3: Claims 1-30 over a combination Kennedy (Ex. 1005) in view of Nguyen (Ex. 1008) and Fox (Ex. 1006)

The Requestor alleges that the combination of the Kennedy, Nguyen, and Fox references, raise a substantial new question of patentability with respect to claim 1 of the '781 patent. Reading of the Kennedy, Nguyen, and Fox references on the claims is provided on pages 42-57 and in Exhibit CC of the Request.

Each of the Kennedy, Nguyen, and Fox references is new art that provide new, non-cumulative technological teachings that were not previously considered nor discussed on the record during prosecution of the '781 patent. Though Fox was cited on an IDS after allowance, it was never specifically applied against the claims either itself or as in the proposed combination. The combination of Kennedy, Nguyen, and Fox has not been considered against the claims of record.

With regard to the limitation of **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,** Kennedy teaches that, "if this 'default' boot processor fails[,|" then an alternate processor "should take over boot operations to enhance availability of the system." (see Kennedy (Ex. 1005) at 7:5-13). In addition,

Kennedy taught that "one default boot CPU is designated to boot the system, and if this CPU fails, control transfers automatically to an alternate processor." Id. at 2:17-21.

Nguyen supplements in teaching that "If an error originates in the master core rather than the slave core, recovery may be handled by changing the master/slave designations of the execution cores" (Nguyen (Ex. 1008) [0054]-[0055]]), where "[e]xecution cores 120 operate in lock step." Nguyen (Ex. 1008) [0024], "for FRC-enabled processors, such errors are likely to be manifested as FRC errors, since they take the execution cores out of lock-step." Id. [0008]. "[E]ach execution core may include a status bit that is set if an error is detected. The master execution core may check this bit to determine if it has also generated an error." Id. [0050].

With regard to the limitation of **wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor,** Kennedy modified by Nguyen teaches a spare processor that assumes the role of boot processor in response to a loss of lockstep (supra). Kennedy does not explicitly identify the spare processor as having been a hot spare processor that is held in reserve. Fox, however, explicitly transfers the role of boot processor to a hot spare processor. See, e.g., Fox (Ex. 1006), Abstract. For example, according to Fox, "[t]he invention provides a substantially seamless transfer of executing processes from the failing processor to the replacement processor, without halting the execution of the operating system (OS) and/or applications on the overall MP. That is, a failing processor

is able to automatically fail-over to a hot-spare processor without affecting the OS or executing applications." Fox (Ex. 1006) at 4:16-23.

It is agreed that the combination of the Kennedy, Nguyen, and Fox references does raise a substantial new question of patentability, as proposed in the request, to at least claim 1 of the '781 patent. There is a substantial likelihood that a reasonable Examiner would consider these teachings important in deciding whether or not this claim is patentable.

Accordingly, the combination of the Kennedy, Nguyen, and Fox references does raise a substantial new question to at least claim 1, which question has not been decided in a previous examination of the '781 patent nor was there a final holding of invalidity by the Federal Courts regarding the '781 patent.

35 USC 325(d)

A review of the post grant history for the underlying patent indicates that there have been no other Office post grant challenges made to the patent (Reexamination Proceedings or *Inter Partes* Review, Post Grant Review, Covered Business Method trials). Accordingly, a discretionary denial of reexamination pursuant to 35 USC 325(d) is not applicable.

Summary

Claims 1-30 are hereby ORDERED to be reexamined.

Conclusion

Extensions of time under 37 CFR 1.136(a) do not apply in reexamination proceedings. The provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Further, in 35 U.S.C. 305 and in 37 CFR 1.550(a), it is required that reexamination proceedings "will be conducted with special dispatch within the Office."

The patent owner is reminded of the continuing responsibility under 37 CFR 1.565(a) to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving the patent throughout the course of this reexamination proceeding. The requester is also reminded of the ability to similarly appraise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP § § 2207, 2282, and 2286.

All correspondence relating to this *ex parte* reexamination proceeding should be directed:

By Mail to: Mail Stop Ex Parte Reexam
Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to: (571) 273-9900

Central Reexamination Unit

By hand: Customer Service Window

Randolph Building

401 Dulany Street

Alexandria, VA 22314

By EFS-Web:

Registered users of EFS-Web may alternatively submit such correspondence via the electronic filing system EFS-Web, at

https://efs.uspto.gov/efile/myportal/efs-registered

EFS-Web offers the benefit of quick submission to the particular area of the Office that needs to act on the correspondence. Also, EFS-Web submissions are "soft scanned" (i.e., electronically uploaded) directly into the official file for the reexamination proceeding, which offers parties the opportunity to review the content of their submissions after the "soft scanning" process is complete.

Any inquiry concerning this communication or earlier communications from the Reexamination Legal Advisor or Examiner, or as to the status of this proceeding, should be directed to the Central Reexamination Unit at telephone number (571) 272-7705.

/DENNIS G BONSHOCK/
Primary Examiner, Art Unit 3992

Conferees:

/ADAM L BASEHOAR/
Primary Examiner, Art Unit 3992

/ALEXANDER J KOSOWSKI/
Supervisory Patent Examiner, Art Unit 3992

Reexamination	Application/Control No.	Applicant(s)/Patent Under Reexamination
	90/019,244	7627781
	Certificate Date	Certificate Number

Requester Correspondence Address: ☐ Patent Owner ☑ Third Party
GREENBERG TRAURIG (NJ) 500 CAMPUS DRIVE, SUITE 400 P.O. BOX 677 FLORHAM PARK NJ 07932

LITIGATION REVIEW ☑	/DGB/ (examiner initials)	03 October 2023 (date)
Case Name		Director Initials
Case No. 2:23-cv-00640 (W.D. Tex., June 6, 2023) Foras Technologies Ltd . v. Nissan Motor Company, LTD et al.		
Case No. 2:23-cv-00321 (E.D. Tex., July 12, 2023) Foras Technologies Ltd . v. Toyota Motor Company, Inc. et al.		
Case No. 2:23-cv-00150 (E.D. Tex., April 5, 2023) Foras Technologies Ltd. v. Toyota Motor Company, Inc. et al.		

COPENDING OFFICE PROCEEDINGS	
TYPE OF PROCEEDING	NUMBER
NONE	

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	90/019,244	7627781
	Examiner	Art Unit
	DENNIS G BONSHOCK	3992

CPC - Searched*		
Symbol	Date	Examiner

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
Prosecution History	10/04/2023	/DGB/
Litigation History	10/04/2023	/DGB/
PDP	10/04/2023	/DGB/
P-TACTS	10/04/2023	/DGB/

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 7031

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
90/019,244	09/01/2023 RULE	714	3992	

APPLICANTS

FORAS TECHNOLOGIES LTD(OWNER, Residence Not Provided;
LLC UNIFIED PATENTS,, CHEVY CHASE, MD;

INVENTORS

7627781, Residence Not Provided;
GREENBERG TRAURIG (NJ), Residence Not Provided;

**** CONTINUING DATA ***************************

This application is a REX of 10/973,077 10/25/2004 PAT 7627781

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
❑ Yes ☑ No	❑ Yes ☑ No	/Dennis Bonshock/ Examiner's Signature	❑ Initials			30	4

ADDRESS

Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528
UNITED STATES

TITLE

SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

FILING FEE RECEIVED	FEES: Authority has been given in Paper	
0.00	No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	❑ All Fees ❑ 1.16 Fees (Filing) ❑ 1.17 Fees (Processing Ext. of time) ❑ 1.18 Fees (Issue) ❑ Other _____ ❑ Credit

BIB (Rev. 05/07).

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**		
	Application Number	
	Filing Date	
	First Named Inventor	
	Art Unit	
	Examiner Name	
	Attorney Docket Number	185945.012400

U.S.PATENTS						Remove
Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	7627781		2009-12-01	Michaelis et al.	
	2	7085959		2022-06-17	Safford	
	3	5450576		1995-09-12	Kennedy et al.	
	4	7251746		2007-07-31	Fox	
	5	7328371		2008-02-05	Kalyanasundharam	
	6	7191328		2007-03-13	Hobson	
	7	5491788		2014-05-14	Cepulis et al.	
If you wish to add additional U.S. Patent citation information please click the Add button.						Add
U.S.PATENT APPLICATION PUBLICATIONS						Remove

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /D.G.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**		
	Application Number	
	Filing Date	
	First Named Inventor	
	Art Unit	
	Examiner Name	
	Attorney Docket Number	185945.012400

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20040123201		2001-06-24	Nguyen et al.	
	2	20030051190		2004-01-06	Marisetty et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	U.S. File History of the '781 Patent ("U.S. File History")	☐
	2	Declaration of Andrew Wolfe ("Wolfe Decl.")	☐
	3	Wendy Bartlett, et. al., Commercial Fault Tolerance: A Tale of Two Systems, IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1 (January-March 2004) ("Bartlett")	☐

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /D.G.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	
	First Named Inventor	
	Art Unit	
	Examiner Name	
	Attorney Docket Number	185945.012400

	4	Intel, Extensible Firmware Interface Specification, Version 1.10 (Dec. 1, 2002) (“Intel EFI Specification”)	☐
	5	Bozidar Levi, UNIX Administration, A Comprehensive Sourcebook for Effective Systems and Network Management (2002) (“Levi”)	☐
	6	Intel IA-64 Architecture Software Developer’s Manual, Revision 1.1 (July 2000) (“Intel Itanium Manual”)	☐
	7	Intel Multiprocessor Specification, Version 1.4 (May 1997) (“Intel Multiprocessor Specification”)	☐
	8		☐
	9		☐
	10		☐
	11		☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/DENNIS G BONSHOCK/	Date Considered	10/04/2023

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /D.G.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**		
	Application Number	
	Filing Date	
	First Named Inventor	
	Art Unit	
	Examiner Name	
	Attorney Docket Number	185945.012400

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☐ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Jon Bills/	Date (YYYY-MM-DD)	2023-09-01
Name/Print	Jon Bills	Registration Number	71895

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /D.G.B/

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /D.G.B/

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 7031

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
90/019,244	09/01/2023 RULE	714	3992	185945.012400

APPLICANTS

FORAS TECHNOLOGIES LTD(OWNER, Residence Not Provided;
LLC UNIFIED PATENTS,, CHEVY CHASE, MD;

INVENTORS

7627781, Residence Not Provided;
GREENBERG TRAURIG (NJ), Residence Not Provided;

**** CONTINUING DATA ***************************

This application is a REX of 10/973,077 10/25/2004 PAT 7627781

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

Foreign Priority claimed ❑ Yes ❑ No 35 USC 119(a-d) conditions met ❑ Yes ❑ No Verified and Acknowledged _____ Examiner's Signature	❑ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
				30	4

ADDRESS

Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528
UNITED STATES

TITLE

SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

FILING FEE RECEIVED		
0.00	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	❑ All Fees ❑ 1.16 Fees (Filing) ❑ 1.17 Fees (Processing Ext. of time) ❑ 1.18 Fees (Issue) ❑ Other _____ ❑ Credit

BIB (Rev. 05/07).

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/019,244	09/01/2023	7627781	185945.012400	7031

56436 7590 09/13/2023

Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528

EXAMINER	
BONSHOCK, DENNIS G	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
09/13/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

GREENBERG TRAURIG (NJ)
500 CAMPUS DRIVE, SUITE 400
P.O. BOX 677
FLORHAM PARK NJ 07932

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/019,244* .

PATENT UNDER REEXAMINATION *7,627,781* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Ex Parte Reexamination Interview Summary – Pilot Program for Waiver of Patent Owner's Statement	Control No. 90/019,244	Patent Under Reexamination is Requested 7,627,781	
	Examiner BONSHOCK, DENNIS	Art Unit 3992	AIA (FITF) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address. --

All participants (USPTO official and patent owner):

(1) Hewlett Packard Enterprise (3) _____

(2) Renee Preston (4) _____

Date of Telephonic Interview: 11 September 2023.

A. The USPTO official requested waiver of the patent owner's statement pursuant to the pilot program for waiver of patent owner's statement in *ex parte* reexamination proceedings.*

☐ The patent owner **agreed** to waive its right to file a patent owner's statement under 35 U.S.C. 304 in the event reexamination is ordered for the above-identified patent.

☐ The patent owner **did not agree** to waive its right to file a patent owner's statement under 35 U.S.C. 304 at this time.

☑ USPTO personnel were unable to reach the patent owner.**

B. The Patent Owner of record telephoned the Office and indicated they would like to participate in the pilot program for waiver of patent owner's statement in *ex parte* reexamination proceedings.*

☐ The Patent owner of record telephoned the Office and **agreed** to waive its right to file a patent owner's statement under 35 U.S.C. 304 in the event reexamination is ordered for the above-identified patent.

The patent owner is not required to file a written statement of this telephone communication under 37 CFR 1.560(b) or otherwise. However, any disagreement as to this interview summary must be brought to the immediate attention of the USPTO, and no later than one month from the mailing date of this interview summary. Extensions of time are governed by 37 CFR 1.550(c).

*For more information regarding this pilot program, see *Pilot Program for Waiver of Patent Owner's Statement in Ex Parte Reexamination Proceedings, 75 Fed. Reg.* 47269 (August 5, 2010), available on the USPTO Web site at http://www.uspto.gov/patents/law/notices/2010.jsp.

**The patent owner may contact the USPTO personnel at (571) 272-7705 or at the telephone number provided below if the patent owner decides to waive the right to file a patent owner's statement under 35 U.S.C. 304.

/RENEE M PRESTON/
CRU Paralegal, OCRU (571) 272-1607
Signature and telephone number of the USPTO official, who contacted, was contacted by, or attempted to contact the patent owner.

cc: Requester (if third party requester)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

REEXAM CONTROL NUMBER	FILING OR 371 (c) DATE	PATENT NUMBER
90/019,244	09/01/2023	

CONFIRMATION NO. 7031
REEXAMINATION REQUEST
NOTICE

GREENBERG TRAURIG (NJ)
500 CAMPUS DRIVE, SUITE 400
P.O. BOX 677
FLORHAM PARK, NEW JERSEY 07932

OC000000061668572

NOTICE OF REEXAMINATION REQUEST FILING DATE

(Third Party Requester)

Requester is hereby notified that the filing date of the request for reexamination is 09/01/2023, the date that the filing requirements of 37 CFR § 1.510 were received.

A decision on the request for reexamination will be mailed within three months from the filing date of the request for reexamination. (See 37 CFR 1.515(a)).

A copy of the Notice is being sent to the person identified by the requester as the patent owner. Further patent owner correspondence will be the latest attorney or agent of record in the patent file. (See 37 CFR 1.33). Any paper filed should include a reference to the present request for reexamination (by Reexamination Control Number).

cc: Patent Owner
56436
Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528

/pflewis/

Legal Instruments Examiner
Central Reexamination Unit 571-272-7705; FAX No. 571-273-9900

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

REEXAM CONTROL NUMBER	FILING OR 371 (c) DATE	PATENT NUMBER
90/019,244	09/01/2023	

CONFIRMATION NO. 7031

REEXAM ASSIGNMENT NOTICE

56436
Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528

OC000000061668593

Date Mailed: 09/11/2023

NOTICE OF ASSIGNMENT OF REEXAMINATION REQUEST

The above-identified request for reexamination has been assigned to Art Unit 3992. All future correspondence to the proceeding should be identified by the control number listed above and directed to the assigned Art Unit.

A copy of this Notice is being sent to the latest attorney or agent of record in the patent file or to all owners of record. (See 37 CFR 1.33(c)). If the addressee is not, or does not represent, the current owner, he or she is required to forward all communications regarding this proceeding to the current owner(s). An attorney or agent receiving this communication who does not represent the current owner(s) may wish to seek to withdraw pursuant to 37 CFR 1.36 in order to avoid receiving future communications. If the address of the current owner(s) is unknown, this communication should be returned within the request to withdraw pursuant to Section 1.36.

NOTICE OF USPTO EX PARTE REEXAMINATION PATENT OWNER STATEMENT WAIVER PROGRAM

The USPTO has implemented a pilot program where, after a reexamination proceeding has been granted a filing date and before the examiner begins his or her review, the patent owner may orally waive the right to file a patent owner's statement. See *"Pilot Program for Waiver of Patent Owner's Statement in Ex Parte Reexamination Proceedings,"* 75 FR 47269 (August 5, 2010). One goal of the pilot program is to reduce the pendency of reexamination proceedings and improve the efficiency of the reexamination process.

Ordinarily when ex parte reexamination is ordered, the USPTO must wait until after the receipt of the patent owner's statement and the third party requester's reply, or after the expiration of the time period for filing the statement and reply (a period that can be as long as 5 to 6 months), before mailing a first determination of patentability. The USPTO's first determination of patentability is usually a first Office action on the merits or a Notice of Intent to Issue Reexamination Certificate (NIRC).

Under the pilot program, the patent owner's oral waiver allows the USPTO to act on the first determination of patentability immediately after determining that reexamination will be ordered, and in a suitable case issue the reexamination order and the first determination of patentability (which could be a NIRC if the claims under reexamination are confirmed) at the same time.

Benefits to the Patent Owner for participating in this pilot program include reduction in pendency.

To participate in this pilot program, Patent Owners may contact the USPTO's Central Reexamination Unit (CRU) at 571-272-7705. The USPTO will make the oral waiver of record in the reexamination file in an interview summary and a copy will be mailed to the patent owner and any third party requester.

cc: Third Party Requester(if any)
GREENBERG TRAURIG (NJ)
500 CAMPUS DRIVE, SUITE 400
P.O. BOX 677
FLORHAM PARK, NEW JERSEY 07932

/pflewis/

Legal Instruments Examiner
Central Reexamination Unit 571-272-7705; FAX No. 571-273-9900



Assignment Query System

Close
Print
Main Page
Patents
Trademarks

Patent Assignment Abstract of Title

Total Assignments: 4

Application #: 10973077 **Filing Dt:** 10/25/2004 **Patent #:** 7627781 **Issue Dt:** 12/01/2009
PCT #: NONE **Intl Reg #:** **Publication #:** US20060107112 **Pub Dt:** 05/18/2006
Inventors: Scott L. Michaelis, Anurupa Rajkumari
Title: SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

Assignment: 1

Reel/Frame:	Received:	Recorded:	Mailed:	Pages:
015941 / 0381	11/04/2004	10/25/2004	04/25/2005	3

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors: MICHAELIS, SCOTT L. **Exec Dt:** 10/20/2004
RAJKUMARI, ANURUPA **Exec Dt:** 10/20/2004

Assignee: HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.
20555 SH 249
HOUSTON, TEXAS 77070

Correspondent: HEWLETT-PACKARD COMPANY
RECORDS MANAGER
INTELLECTUAL PROPERTY ADMINISTRATION
P.O. BOX 272400
FORT COLLINS, CO 80527-2400

Assignment: 2

Reel/Frame:	Received:	Recorded:	Mailed:	Pages:
037079 / 0001	11/09/2015	11/09/2015	11/20/2015	1112

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. **Exec Dt:** 10/27/2015

Assignee: HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP
11445 COMPAQ CENTER DRIVE WEST
HOUSTON, TEXAS 77070

Correspondent: HEWLETT PACKARD ENTERPRISE
3404 E. HARMONY ROAD MS 79
FORT COLLINS, CO 80528

Assignment: 3

Reel/Frame:	Received:	Recorded:	Mailed:	Pages:
052567 / 0734	05/05/2020	05/05/2020	05/06/2020	6

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors: HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP **Exec Dt:** 04/23/2020
HEWLETT PACKARD ENTERPRISE COMPANY **Exec Dt:** 04/23/2020

Assignee: SONRAÍ MEMORY, LTD.
SUITE 23, THE HYDE BUILDING, THE PARK, CARRICKMINE
DUBLIN, IRELAND DUBLIN 18

Correspondent: ADAM MULLOOLY
SUITE 23, THE HYDE BUILDING, THE PARK, CARRICKMINE
DUBLIN, DUBLIN 18 IRELAND

Assignment: 4

Reel/Frame:	Received:	Recorded:	Mailed:	Pages:
058992 / 0571	02/11/2022	02/11/2022	02/22/2022	5

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: SONRAÍ MEMORY, LTD. **Exec Dt:** 01/19/2022

Assignee: FORAS TECHNOLOGIES LTD.
THE HYDE BUILDING, SUITE 23,
THE PARK, CARRICKMINES.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 7031

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
90/019,244	09/01/2023 RULE	714	3992	185945.012400

APPLICANTS

LP HEWLETT-PACKARD DEVELOPMENT COMPANY,, Residence Not Provided;
LLC UNIFIED PATENTS,, CHEVY CHASE, MD;

INVENTORS

7627781, Residence Not Provided;
GREENBERG TRAURIG (NJ), Residence Not Provided;

**** CONTINUING DATA ***************************

This application is a REX of 10/973,077 10/25/2004 PAT 7627781

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

Foreign Priority claimed ❑ Yes ❑ No 35 USC 119(a-d) conditions met ❑ Yes ❑ No Verified and Acknowledged _____ Examiner's Signature	❑ Met after Allowance _____ Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
				30	4

ADDRESS

Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528
UNITED STATES

TITLE

SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

FILING FEE RECEIVED		
0.00	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	❑ All Fees ❑ 1.16 Fees (Filing) ❑ 1.17 Fees (Processing Ext. of time) ❑ 1.18 Fees (Issue) ❑ Other _____ ❑ Credit

BIB (Rev. 05/07).

Litigation Search Report CRU 3999

Reexamination Control No. 90/019244

To: Examiner	**From: Karen Ward**
Location: CRU	
Art Unit: 3999	**Phone: (571) 272-7932**
Date: 09/07/2023	**E-mail: karen.ward@uspto.gov**

Search Notes

U.S. Patent No. 7,627,781

1. Performed a KeyCite Search in Westlaw, which retrieves all history on the patent including any litigation.
2. Performed a search on the patent in Lexis CourtLink for any open dockets or closed cases.
3. Performed a search in Lexis in the Federal Courts and Administrative Materials databases for any cases found.
4. Performed a search in Lexis in the IP Journal and Periodicals database for any articles on the patent.
5. Performed a search in Lexis in the news databases for any articles about the patent or any articles about litigation on this patent.

Litigation: Cases Found

Status	Docket	Description
OPEN	1:23cv640	Foras Technologies Ltd. V. Nissan Motor Company, Ltd. Et Al
OPEN	2:23cv150	Foras Technologies Ltd. V. Toyota Motor North America, Inc. Et Al

Citing References (20)

Treatment	Title	Date	Type	Depth	Headnote(s)
—	**1. SPARE PROCESSOR ESTABLISHING SYSTEM USED IN MULTI-PROCESSOR SYSTEM, TRANSFERS ROLE OF BOOT PROCESSOR TO SPARE PROCESSOR** Out Of Plan DWPI 2006-372232 ... said boot processor, for transferring, during system runtime, the role of boot processor to said spare processor.First Derwent Appearance:2006.38 Publication No. (Derwent): US 7627781 B2 Original Title (English):SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR Publication Date ...	Oct. 25, 2004	DWPI	—	—
—	**2. RF 058982/0571** Out Of Plan ... 22 PATENTS AFFECTED Title SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR Granted Patent Number US Pat. 7627781 Granted Patent Date 2009-12-01 Published Application Number US Pat. App. 20060107112 Published Application Date 2006-05-18 Application Number 10/973077 ...	Feb. 11, 2022	Assignments	—	—
—	**3. RF 052567/0734** Out Of Plan ... 2004-10-25 Title SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR Granted Patent Number US Pat. 7627781 Granted Patent Date 2009-12-01 Published Application Number US Pat. App. 20060107112 Published Application Date 2006-05-18 Application Number 10/973077 ...	May 05, 2020	Assignments	—	—
—	**4. RF 037079/0001** Out Of Plan ... 2005-02-25 Title SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR Granted Patent Number US Pat. 7627781 Granted Patent Date 2009-12-01 Published Application Number US Pat. App. 20060107112 Published Application Date 2006-05-18 Application Number 10/973077 ...	Nov. 09, 2015	Assignments	—	—
—	**5. RF 037079/0001-4**	Nov. 09, 2015	Assignments	—	—
—	**6. RF 015941/0381** Out Of Plan ... 01 PATENTS AFFECTED Title SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR Granted Patent Number US Pat. 7627781 Granted Patent Date 2009-12-01 Published Application Number US Pat. App. 20060107112 Published Application Date 2006-05-18 Application Number 10/973077 ...	Oct. 25, 2004	Assignments	—	—

WESTLAW © 2023 Thomson Reuters. No claim to original U.S. Government Works.

Treatment	Title	Date	Type	Depth	Headnote(s)
—	**7.** PatStat 7627781 ... Patent Status File Patent Number: US 7627781 Change Code: COR Description: Certificate of Correction Reissue Number:OG Date: 11/09/2010 ...	Nov. 09, 2010	Patent Status Files	—	—
—	**8.** CONFIGURE STORAGE CLASS MEMORY COMMAND Out Of Plan US PAT 11163444 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 710/305 G06F 3/0613 cited by examiner US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	Nov. 02, 2021	Patents	—	—
—	**9.** CONFIGURE STORAGE CLASS MEMORY COMMAND Out Of Plan US PAT 10387040 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2003/07 Rooney et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	Aug. 20, 2019	Patents	—	—
—	**10.** DATA RETURNED RESPONSIVE TO EXECUTING A START SUBCHANNEL INSTRUCTION Out Of Plan US PAT 10013256 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2003/07 Rooney et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	July 03, 2018	Patents	—	—

WESTLAW © 2023 Thomson Reuters. No claim to original U.S. Government Works.

Treatment	Title	Date	Type	Depth	Headnote(s)
—	**11. CONFIGURE STORAGE CLASS MEMORY COMMAND** Out Of Plan US PAT 9747033 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2003/07 Rooney et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	Aug. 29, 2017	Patents	—	—
—	**12. DATA RETURNED RESPONSIVE TO EXECUTING A START SUBCHANNEL INSTRUCTION** Out Of Plan US PAT 9477417 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2002/10 Lethin et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	Oct. 25, 2016	Patents	—	—
—	**13. MOVING BLOCKS OF DATA BETWEEN MAIN MEMORY AND STORAGE CLASS MEMORY** Out Of Plan US PAT 9418006 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2002/10 Lethin et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	Aug. 16, 2016	Patents	—	—
—	**14. CLEARING BLOCKS OF STORAGE CLASS MEMORY** Out Of Plan US PAT 9411737 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2002/10 Lethin et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	Aug. 09, 2016	Patents	—	—

WESTLAW © 2023 Thomson Reuters. No claim to original U.S. Government Works.

Treatment	Title	Date	Type	Depth	Headnote(s)
—	**15. CONFIGURE STORAGE CLASS MEMORY COMMAND** Out Of Plan US PAT 9372640 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2002/10 Lethin et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	June 21, 2016	Patents	—	—
—	**16. DECONFIGURE STORAGE CLASS MEMORY COMMAND** Out Of Plan US PAT 9323668 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2002/10 Lethin et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.714/013 cited by examiner US United States US 7809918 B1 2010/10 Samuel 711/171 cited by ...	Apr. 26, 2016	Patents	—	—
—	**17. CHAINING MOVE SPECIFICATION BLOCKS** Out Of Plan US PAT 9164882 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2002/10 Lethin et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	Oct. 20, 2015	Patents	—	—
—	**18. USING EXTENDED ASYNCHRONOUS DATA MOVER INDIRECT DATA ADDRESS WORDS** Out Of Plan US PAT 9122573 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2002/10 Lethin et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	Sep. 01, 2015	Patents	—	—

WESTLAW © 2023 Thomson Reuters. No claim to original U.S. Government Works.

Treatment	Title	Date	Type	Depth	Headnote(s)
—	**19.** CHAINING MOVE SPECIFICATION BLOCKS Out Of Plan US PAT 9122574 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... US 6308255 2001/10 Gorishek, IV et al.US US 6463582 2002/10 Lethin et al.US US 7373378 2008/05 Bauman et al.US US 7627781 2009/12 Michaelis et al.US US 7809918 2010/10 Samuel US US 7844756 2010/11 Cannon et al.US US 2003/0188119 2003 ...	Sep. 01, 2015	Patents	—	—
—	**20.** CONFIGURE STORAGE CLASS MEMORY COMMAND Out Of Plan US PAT 9116634 , U.S. PTO Utility An abstraction for storage class memory is provided that hides the details of the implementation of storage class memory from a program, and provides a standard channel programming... ... 2002/10 Lethin et al.cited by applicant US United States US 7373378 B2 2008/05 Bauman et al.cited by applicant US United States US 7627781 B2 2009/12 Michaelis et al.cited by applicant US United States US 7809918 B1 2010/10 Samuel cited by applicant US United States ...	Aug. 25, 2015	Patents	—	—

WESTLAW © 2023 Thomson Reuters. No claim to original U.S. Government Works.

Cases 0

Filters 1

Cases Clear

No documents found in Cases filtered by Cases.

Try the following:

- Remove one or more filters: Cases
- Review your results in the other categories.
- Modify your search (edit or remove some search terms or add synonyms).
- Check for spelling errors.
- Use the options available from the Actions menu above to modify your search.

Copyright © 2023 LexisNexis.

News 0

Filters 1

News Clear

No documents found in News filtered by News.

Try the following:

- Remove one or more filters: News
- Review your results in the other categories.
- Modify your search (edit or remove some search terms or add synonyms).
- Check for spelling errors.
- Use the options available from the Actions menu above to modify your search.

LexisNexis

About
Privacy Policy
Cookie Policy
Terms & Conditions

Customer Support
1-800-59 COURT / 1-800-592-6878
On Demand Training
Live Chat Support

RELX™

Copyright © 2023 LexisNexis.

☐ US7627781B2 - System and method for establishing a spare processor for recovering from loss of lockstep in a boot processor

Original

Bibliographic information & Images

Publication number and date
US7627781B2
2009-12-01

Granted

Application number and date
US10973077 (/api/control?client=&query=AN:("US10973077")&autosearch=true)
2004-10-25

Priority number and date
US10973077 (/api/control?client=&query=PRN:("US10973077")&autosearch=true)
2004-10-25

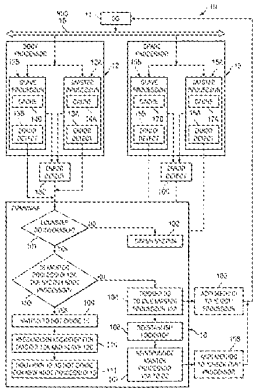
Ultimate owner
FORAS TECHNOLOGIES (/api/control?client=&query=uo:("FORAS%20TECHNOLOGIES")&autosearch=true)

Original assignee
Scott L. Michaelis (/api/control?client=&query=pa:("Scott%20L.%20Michaelis")&autosearch=true)

Current assignee
SONRAÍ MEMORY, LTD. (/api/control?client=&query=pacu:("SONRA%C3%8D%20MEMORY%2C%20LTD.")&autosearch=true)

Inventor
Scott L. Michaelis (/api/control?client=&query=in:("Scott%20L.%20Michaelis")&autosearch=true)

1/5 Clipped image









Abstract

A system comprises a plurality of processors, and data storage storing information that assigns a role of boot processor to one of the plurality of processors and assigns a role of spare processor to another of the plurality of processors. The system further comprises logic operable, responsive to detecting loss of lockstep for the boot processor, for transferring, during system runtime, the role of boot processor to the spare processor.

Original

Claims

1. A system comprising: a plurality of processors; 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; and 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, wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor.

2. The system of claim 1 wherein said data storage is non-volatile data storage.

3. The system of claim 1 further comprising: a user interface operable to receive input specifying said one of said plurality of processors to be assigned said role of boot processor.

4. The system of claim 1 further comprising: a user interface operable to receive input specifying said another of said plurality of processors to be assigned said role of spare processor.

5. The system of claim 1 wherein each of said plurality of processors is a processor module that includes a master processor and a slave processor.

6. The system of claim 1 wherein said another of said plurality of processors that is assigned said role of spare processor is an active spare before being transferred said role of boot processor.

7. The system of claim 1 further comprising: an operating system, wherein, responsive to the logic transferring the role of boot processor to the spare processor, the operating system recognizes the spare processor as the boot processor.

8. The system of claim 1 wherein said plurality of processors are Itanium Processor Family (IPF) processors.

9. A method comprising: assigning a role of boot processor to one of a plurality of processors in a system; assigning a role of spare processor to another of said plurality of processors; 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; and responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system.

10. The method of claim 9 wherein said assigning said role of boot processor comprises storing information to non-volatile data storage that identifies said one of said plurality of processors as said boot processor.

11. The method of claim 9 wherein said assigning said role of spare processor comprises storing information to non-volatile data storage that identifies said another of said plurality of processors as said spare processor.

12. The method of claim 9 further comprising: determining at least one processor that is coupled to a common communication bus with said one of said plurality processors assigned the role of boot processor; and selecting said another of said plurality of processors to be assigned the role of spare processor from said determined at least one processor.

13. The method of claim 9 further comprising: during system boot-up, system firmware identifying the spare processor and holding said spare processor in reserve as a hot spare.

14. The method of claim 9 further comprising: during system boot-up, system firmware booting the spare processor as an active spare.

15. The method of claim 9 wherein system firmware performs said changing operation.

16. A system comprising: a plurality of lockstep processor modules that each includes a master processor and a slave processor that is in lockstep with said master processor; a first communication bus with which at least a first two of said plurality of lockstep processor modules are communicatively coupled; a second communication bus with which at least a second two of said plurality of lockstep processor modules are communicatively coupled; and 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, wherein said spare processor is communicatively coupled to the same one of said first and second communication buses as said boot processor.

17. The system of claim 16 further comprising: system firmware operable, responsive to detecting loss of lockstep for said boot processor, changing the assignment of the role of boot processor to said spare processor.

18. The system of claim 17 wherein said system firmware performs said changing during system runtime without requiring an operating system of said system to be shutdown.

19. The system of claim 16 further comprising: a user interface for receiving information selecting said one of said plurality of processors to be assigned the role of boot processor.

20. The system of claim 16 further comprising: a user interface for receiving information selecting said another of said plurality of processors to be assigned the role of spare processor.

21. The system of claim 20 wherein said user interface presents for selection as said spare processor at least one processor that is communicatively to the same one of said first and second communication buses as said boot processor.

22. The system of claim 16 further comprising: system firmware that, during system boot-up, identifies the spare processor and holds said spare processor in reserve as a hot spare.

23. The system of claim 16 further comprising: system firmware that, during system boot-up, boots the spare processor as an active spare.

24. The system of claim 16 wherein said plurality of lockstep processor modules include Itanium Processor Family (IPF) processors.

25. A method comprising: selecting a first one of a plurality of lockstep processor modules in a system as a boot processor, wherein each of said plurality of lockstep processor modules includes a master processor and a slave processor that is in lockstep with said master processor, and wherein said boot processor is communicatively coupled to a first of a plurality of communication buses; selecting a second one of said plurality of lockstep processor modules as a spare processor, wherein said spare processor is selected as one that is communicatively coupled to the same communication bus as said boot processor; and responsive to detecting loss of lockstep for said boot processor, said system causing said spare processor to dynamically assume the role of boot processor.

26. The method of claim 25 wherein said system causing said spare processor to dynamically assume the role of boot processor comprises: said system causing said spare processor to dynamically assume the role of boot processor without shutting down an operating system of said system.

27. The method of claim 25 further comprising: receiving input to said system specifying said first one of said plurality of lockstep processor modules to be said boot processor.

28. The method of claim 25 further comprising: receiving input to said system specifying said second one of said plurality of lockstep processor modules to be said spare processor.

29. The method of claim 25 further comprising: storing information identifying said selected first one of said plurality of lockstep processor modules as said boot processor and identifying said selected second one of said plurality of lockstep processor modules as said spare processor.

30. The method of claim 29 wherein said system causing said spare processor to dynamically assume the role of boot processor comprises: system firmware determining said spare processor from said stored information; and said system firmware causing an operating system to recognize said spare processor as said boot processor.

Original

Families

Display in a list (/api/control?client=&familiesSearch=_) INPADOC

This section contains all family relations of this particular document. For more information about the different families, refer to the User Manual (Glossary/Families).

Extended (2) Priority date/claims	Complete (2) Priority date/claims	Main (2) Priority date/claims	Domestic (2) Priority date/claims	Publication number	Application number	Title	Status
2004-10-25 US10973077	2004-10-25 US10973077	2004-10-25 US10973077	2004-10-25	US7627781B2	US10973077	System and method for establishing a spare processor for recovering from loss of lockstep in a...	Granted
				US20060107112A1	US10973077	System and method for establishing a spare processor for recovering from loss of lockstep in a...	Granted

Description

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application is related to the following concurrently filed and commonly assigned U.S. Patent Applications: U.S. patent application Ser. No. 10/972,835 (U.S. Published Patent Application 20060107114) titled "SYSTEM AND METHOD FOR USING INFORMATION RELATING TO A DETECTED LOSS OF LOCKSTEP FOR DETERMINING A RESPONSIVE ACTION"; U.S. patent application Ser. No. 10/973,004 (U.S. Published Patent Application 20060107117, now U.S. Pat. No. 7,308,566) titled "SYSTEM AND METHOD FOR CONFIGURING LOCKSTEP MODE OF A PROCESSOR MODULE"; U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION"; U.S. patent application Ser. No. 10/972,588 (U.S. Published Patent Application 20060090064) titled "SYSTEM AND METHOD FOR SWITCHING THE ROLE OF BOOT PROCESSOR TO A SPARE PROCESSOR RESPONSIVE TO DETECTTON OF LOSS OF LOCKSTEP IN A BOOT PROCESSOR"; U.S. patent application Ser. No. 10/973,003 (U.S. Published Patent Application 20060107116) titled "SYSTEM AND METHOD FOR REESTABLISHING LOCKSTEP FOR A PROCESSOR MODULE FOR WHICH LOSS OF LOCKSTEP IS DETECTED"; U.S. patent application Ser. No. 10/972,888 (U.S. Published Patent Application 20060107115) titled "SYSTEM AND METHOD FOR SYSTEM FIRMWARE CAUSING AN OPERATING SYSTEM TO IDLE A PROCESSOR"; U.S. patent application Ser. No. 10/973,075 (U.S. Published Patent Application 20060107111) titled "SYSTEM AND METHOD FOR REINTRODUCING A PROCESSOR MODULE TO AN OPERATING SYSTEM AFTER LOCKSTEP RECOVERY"; and U.S. patent application Ser. No. 10/972,796 (U.S. Published Patent Application 20060107106) titled "SYSTEM AND METHOD FOR MAINTAINING IN A MULTI-PROCESSOR SYSTEM A SPARE PROCESSOR THAT IS IN LOCKSTEP FOR USE IN RECOVERING FROM LOSS OF LOCKSTEP FOR ANOTHER PROCESSOR", the disclosures of which are hereby incorporated herein by reference.

DESCRIPTION OF RELATED ART

[0002] Silent Data Corruption ("SDC") is a difficult problem in the computing industry. In general, SDC refers to data that is corrupt, but which the system does not detect as being corrupt. SDCs primarily occur due to one of two factors: a) a broken hardware unit or b) a "cosmic" event that causes values to change somewhere in the system. Broken hardware means that a "trusted" piece of hardware is silently giving wrong answers. For example, the arithmetic unit in a processor is instructed to add 1+1 and it returns the incorrect answer 3 instead of the correct answer 2. An example of a cosmic event is when a charged particle (e.g., alpha particle or cosmic ray) strikes a region of a computing system and causes some bits to change value (e.g., from a 0 to a 1 or from a 1 to a 0).

[0003] Numerous techniques have been developed for detecting SDC to prevent the SDC from remaining "silent" or "undetected" within a system, as well as preventing such SDC from propagating through the system. Examples of these techniques include parity-based mechanisms and error correcting codes (ECCs) on buses and memory locations, as well as checksums and/or cyclic redundancy checks (CRC) over regions of memory. Parity-based mechanisms are often employed in processors, wherein a parity bit is associated with each block of data when it is stored. The parity bit is set to one or zero according to whether there is an odd or even number of ones in the data block. When the data block is read out of its storage location, the number of ones in the block is compared with the parity bit. A discrepancy between the values indicates that the data block has been corrupted. ECCs are parity-based mechanisms that track additional information for each data block. The additional information allows the corrupted bit(s) to be identified and corrected.

[0004] Parity/ECC mechanisms have been employed extensively for caches, memories, and similar data storage arrays. In the remaining circuitry on a processor, such as data paths, control logic, execution logic, and registers (the "execution core"), it is more difficult to apply parity/ECC mechanisms for SDC detection. Thus, there is typically some unprotected area on a processor in which data corruption may occur and the parity/ECC mechanisms do not prevent the corrupted data from actually making it out onto the system bus. One approach to SDC detection in an execution core (or other unprotected area of the processor chip) is to employ "lockstep processing." Generally, in lockstep processing two processors are paired together, and the two processors perform exactly the same operations and the results are compared (e.g., with an XOR gate). If there is ever a discrepancy between the results of the lockstep processors, an error is signaled. The odds of two processors experiencing the exact same error at the exact same moment (e.g., due to a cosmic event occurring in both processors at exactly the same time or due to a mechanical failure occurring in each processor at exactly the same time) is nearly zero.

[0005] A pair of lockstep processors may, from time to time, lose their lockstep. "Loss of lockstep" (or "LOL") is used broadly herein to refer to any error in the pair of lockstep processors. One example of LOL is detection of data corruption (e.g., data cache error) in one of the processors by a parity-based mechanism and/or ECC mechanism. Another example of LOL is detection of the output of the paired processors not matching, which is

referred to herein as a "lockstep mismatch." It should be recognized that in some cases the data in the cache of a processor may become corrupt (e.g., due to a cosmic event), which once detected (e.g., by a parity-based mechanism or ECC mechanism of the processor) results in LOL. Of course, unless such corrupt data is acted upon by the processor, the output of that processor will not fail to match the output of its paired processor and thus a "lockstep mismatch" will not occur. For example, suppose that a value of "1" is stored to first location of cache in each of a pair of lockstep processors and a value of "1" is also stored to a second location of cache in each of the pair of lockstep processors. Further suppose that a cosmic event occurs for a first one of the processors, resulting in the first location of its cache being changed from "1" to "0", and thus corrupted. This data corruption in the first processor is a LOL for the pair. An error detection mechanism of this first processor may detect the data corruption, thus detecting the LOL. If the processors are instructed to act on the data of their first cache locations, then a lockstep mismatch will occur as the output of each of the processors will not match. For instance, if the processors each add the data stored to the first location of their respective cache with the data stored to the second location of their respective cache, the first processor (having the corrupt data) will output a result of "1" (0+1=1) while the second processor outputs a result of "2" (1+1=2), and thus their respective outputs will not match.

[0006] By employing such techniques as parity-based error detection mechanisms and output comparisons for lockstep paired processors, SDC detection can be enhanced such that practically no SDC occurring in a processor goes undetected (and thus such SDC does not remain "silent") but instead results in detection of LOL. However, the issue then becomes how best for the system to respond to detected LOL. The traditional response to detected LOL has been to crash the system to ensure that the detected error is not propagated through the system. That is, LOL in one pair of lockstep processors in a system halts processing of the system even if other processors that have not encountered an error are present in the system. However, with the increased desire for many systems to maintain high availability, crashing the system each time LOL is detected is not an attractive proposition. This is particularly unattractive for large systems having many processors because cosmic events typically occur more frequently as the processor count goes up, which would result in much more frequent system crashes in those large systems. High availability is a major desire for many customers having large, multi-processor systems, and thus having their system crash every few weeks is not an attractive option. Of course, permitting corrupt data to propagate through the system is also not a viable option.

[0007] Prior solutions attempting to resolve at least some detected SDCs without requiring the system to be crashed have been Operating System ("OS") centric. That is, in certain solutions the OS has been implemented in a manner to recover from a detected LOL without necessarily crashing the system. This OS-centric type of solution requires a lot of processor and platform specific knowledge to be embedded in the OS, and thus requires that the OS provider maintain the OS up-to-date as changes occur in later versions of the processors and platforms in which the OS is to be used. This is such a large burden that most commonly used OSs do not support lockstep recovery.

[0008] Certain solutions have attempted to recover from a LOL without involving the OS in such recovery procedure. For instance, in one technique upon LOL being detected, firmware is used to save the state of one of the processors in a lockstep pair (the processor that is considered "good") to memory, and then both processors of the pair are reset and reinitialized. Thereafter, the state is copied from the memory to each of the processors in the lockstep pair. This technique makes the processors unavailable for an amount of time without the OS having any knowledge regarding this unavailability, and if the amount of time required for recovery is too long, the system may crash. That is, typically, if a processor is unresponsive for X amount of time, the OS will assume that the processor is hung and will crashdump the system so that the problem can be diagnosed. Further, in the event that a processor in the pair cannot be reset and reinitialized (e.g., the processor has a physical problem and fails to pass its self-test), this technique results in crashing the system.

BRIEF SUMMARY OF THE INVENTION

[0009] According to at least one embodiment, a system comprises a plurality of processors, and data storage storing information that assigns a role of boot processor to one of the plurality of processors and assigns a role of spare processor to another of the plurality of processors. The system further comprises logic operable, responsive to detecting loss of lockstep for the boot processor, for transferring, during system runtime, the role of boot processor to the spare processor.

[0010] According to at least one embodiment, a method comprises assigning a role of boot processor to one of a plurality of processors in a system; assigning a role of spare processor to another of the plurality of processors; and responsive to detecting loss of lockstep for the boot processor, changing, during system runtime, the assignment of the role of boot processor to the spare processor.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 shows an example multi-processor system in which firmware utilizes a spare processor for recovering from LOL for the system's boot processor;

[0012] FIG. 2 shows a block diagram of one embodiment implemented for the IA-64 processor architecture;

[0013] FIG. 3 shows an example system having multi-processor cells in which an embodiment for using firmware for recovering from LOL may be employed;

[0014] FIG. 4 shows an operational flow diagram for establishing a hot spare for the boot processor according to one embodiment;

[0015] FIG. 5 shows an operational flow diagram for establishing an active spare for the boot processor and assigning such active spare the role of boot processor responsive to detecting LOL for the boot processor in accordance with one embodiment; and

[0016] FIG. 6 shows a general flow diagram for operation of certain embodiments.

DETAILED DESCRIPTION

[0017] As described further herein and in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION," the disclosure of which is incorporated herein by reference, certain techniques are provided for recovering from LOL detected for a boot processor in a multi-processor system, in which a spare processor is utilized for such recovery. For instance, upon LOL being detected for the system's boot processor, the spare processor assumes the role of boot processor, and then recovery of lockstep for the original boot processor is attempted. Accordingly, for such lockstep recovery technique, a spare processor is used for attempting to recover from LOL detected for the boot processor. Embodiments are provided herein for establishing a spare processor for the boot processor. That is, embodiments described herein provide techniques for establishing, in a multi-processor system, a spare processor that assumes the role of boot processor upon LOL being detected for the system's boot processor.

[0018] In certain embodiments, information (referred to herein as a "device tree") is stored to the multi-processor system, wherein such information identifies the processor in the system that is the boot processor and identifies the processor that is the spare for the boot processor. The information is stored such that it is accessible by the system's firmware, and the firmware, 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. The firmware then attempts to recover

lockstep for the processor for which LOL was detected. If 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. Thus, 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).

[0019] In some implementations described herein, the spare processor for the boot processor is a "hot" spare, in which it is idled and not made available to the OS for processing instructions during normal system operation. That is, the hot spare processor is held in reserve, idling until such time as it assumes the role of boot processor responsive to detected LOL for the boot processor. In other implementations, the spare processor is not held in reserve, but is instead an "active" processor that is made available to the OS for executing instructions during normal system operation. Upon LOL being detected for the boot processor, the designated spare processor is dynamically transformed into a hot spare (i.e., it is idled) such that it can assume the role of the boot processor. For instance, upon the system's firmware detecting a LOL for the boot processor, the firmware accesses the device tree and determines the designated spare processor, and then the firmware instructs the OS to idle the designated spare processor (i.e., stop scheduling instructions for the processor) to make the "active" spare a "hot" spare, which then assumes the role of boot processor.

[0020] Concurrently filed and commonly assigned U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION," discloses a technique in which lockstep can be recovered for processors (particularly, non-boot processors) without requiring use of a spare processor. Rather, upon LOL being detected for a lockstep pair of processors, the firmware instructs the OS to idle (or "eject") the lockstep pair of processors for which LOL was detected and then the firmware attempts to reestablish lockstep (e.g., by resetting the processors). If successful in reestablishing lockstep, the firmware triggers the OS to recognire the lockstep pair of processors as again being available for servicing instructions.

[0021] However, for various reasons, in certain system architectures problems arise in attempting to idle (or eject) the boot processor from the system. Thus, in certain embodiments, a spare processor is used for recovering from LOL for the system's boot processor. Again, embodiments are provided herein for establishing/defining such a spare processor for the boot processor. In certain embodiments, the spare processor is used for recovering from LOL only in the case in which LOL is detected for the boot processor. For instances in which LOL is detected for non-boot processors, the lockstep may be recovered without the use of a spare by using the recovery technique described in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION," for example.

[0022] Various techniques are provided herein for designating the spare for a boot processor. In certain embodiments, a user interface is provided for receiving input specifying the boot processor and/or its spare. In certain embodiments, the system's firmware selects the boot processor and/or its spare. In accordance with one embodiment provided herein, the multi-processor system includes multiple buses, wherein certain processors are communicatively coupled to a first bus and certain other processors are communicatively coupled to a different bus. In such an architecture, the spare processor is designated as a processor that is coupled to the same bus as the boot processor. Thus, if a user is allowed to select the boot processor and the spare, once the boot processor is selected, the user interface may present the other processor(s) that are coupled to the same bus as the selected boot processor for selection of a spare therefrom. Alternatively, once the boot processor is selected by a user, the system firmware may select a spare as a processor that is coupled to the same bus as the boot processor. In certain embodiments, the system firmware may select the boot processor, as well as a spare for the boot processor.

[0023] Turning to FIG. 1, an example embodiment of a system **10** that establishes a spare processor for use in recovering from detected loss of lockstep (LOL) for a system boot processor is shown. System **10** includes OS **11**, as well as master processor **12**A and slave processor **12**B (collectively referred to as a lockstep processor pair **12**). In certain implementations the lockstep processor pair **12** may be implemented on a single silicon chip, which is referred to as a "dual core processor" in which master processor **12**A is a first core and slave processor **12**B is a second core. Further, lockstep processor pair **12** may be referred to as a processor or CPU "module" because it includes a plurality of processors (**12**A and **12**B) in such module. As described further herein, to certain portions of system **10**, such as its OS **11**, the processor module **12** may appear as a single processor because slave processor **12**B is merely used for redundant processing for checking the output of master processor **12**A. In this example, lockstep processor pair **12** is designated as the boot processor for system **10**. As described further herein, in certain embodiments lockstep processor pair **12** is designated as the system boot processor by storing information to the system that specifies such designation.

[0024] Master processor **12**A includes cache **14**A, and slave processor **12**B includes cache **14**B. OS **11** and lockstep processor pair **12** are communicatively coupled to bus **16**. Typically, master processor **12**A and slave processor **12**B are coupled to bus **16** via an interface that allows each of such processors to receive the same instructions to process, but such interface only communicates the output of master processor **12**A back onto bus **16**. The output of slave processor **12**B is used solely for checking the output of mater processor **12**A.

[0025] In this example, master processor **12**A includes error detect logic **13**A, and slave processor **12**B includes error detect logic **13**B. While shown as included in each of the processors **12**A and **12**B in this example, in certain embodiments the error detect logic **13**A and **13**B may be implemented external to processors **12**A and **12**B. Error detect logic **13**A and **13**B include logic for detecting errors, such as data cache errors, present in their respective processors **12**A and **12**B. Examples of error detect logic **13**A and **13**B include known parity-based mechanisms and ECC mechanisms. Error detect logic **13**C is also included, which may include an XOR (exclusive OR) gate, for detecting a lockstep mismatch between master processor **12**A and slave processor **12**B. As mentioned above, a lockstep mismatch refers to the output of master processor **12**A and slave processor **12**B failing to match. While shown as external to the lockstep processor pair **12** in this example, in certain embodiments error detect logic **13**C may be implemented on a common silicon chip with processors **12**A and **12**B.

[0026] Lockstep mismatch is one way of detecting a LOL between the master processor **12**A and slave processor **12**B. A detection of an error by either of error detect logic **13**A and **13**B also provides detection of LOL in the processors **12**A and **12**B. Because the detection of LOL by error detect logic **13**A and **13**B may occur before an actual lockstep mismatch occurs, the detection of LOL by error detect logic **13**A and **13**B may be referred to as a detection of a "precursor to lockstep mismatch". In other words, once an error (e.g., corrupt data) is detected by error detect logic **13**A or **13**B, such error may eventually propagate to a lockstep mismatch error that is detectable by error detect logic **13**C.

[0027] System **10** further includes a second lockstep processor pair **15** that is communicatively coupled to bus **16**. Lockstep processor pair **15** includes master processor **15**A and slave processor **15**B. Master processor **15**A includes cache **17**A, and slave processor **15**B includes cache **17**B. As with error detect logic **13**A-**13**C implemented for lockstep processor pair **12**, lockstep processor pair **15** has error detect logic **16**A-**16**C. While only two lockstep processor pairs **12** and **15** are shown for simplicity in the example of FIG. 1, system **10** may include any number of such lockstep processor pairs. As one specific example, in one implementation system **10** includes 64 lockstep processor pairs, wherein the master processors of the pairs perform parallel processing for the system. In the specific example of FIG. 1, lockstep processor pair **15** is designated as a spare for boot processor **12**, as described further herein.

[0028] In certain embodiments described herein, techniques for designating a spare processor for a boot processor are provided, wherein the spare processor is used for recovery when LOL is detected for the system's boot processor. That is, the designated spare processor assumes the role of boot processor in the event that LOL is detected for the boot processor. In accordance with certain embodiments, upon LOL being detected for a boot processor, a hot spare processor (i.e., an idling processor that is available in the system) is transferred the role of boot processor, and then the old boot processor having LOL is reset to reestablish its lockstep. Thus, in certain embodiments, a "hot" spare is designated for the boot processor.

[0029] In other embodiments, 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 processing resource made available to OS **11** for processing instructions that OS **11** schedules for such "active" spare processor. Upon LOL being detected for the boot processor, the designated spare processor is idled (thereby, transforming such active processor into a hot spare), and is transferred the role of boot processor. Then, the former boot processor for which the LOL was detected is reset to reestablish its lockstep. Thus, in certain embodiments, a designated spare processor is not required to be held in reserve, but is instead an available resource for use within the system and is only transformed into an actual spare processor when needed for recovering from LOL for the boot processor.

[0030] Firmware **18** is also included in system **10**, which in this embodiment is invoked upon an error being detected by any of the error detect logics **13**A-**13**C and **16**A-**16**C. In certain embodiments, processors **12**A, **12**B, **15**A, and **15**B are processors from the Itanium Processor Family (IPF). IPF is a 64-bit processor architecture co-developed by Hewlett-Packard Company and Intel Corporation, which is based on Explicitly Parallel Instruction Computing (EPIC). IPF is a well-known family of processors. IPF includes processors such as those having the code names of MERCED, MCKINLEY, and MADISON. In addition to supporting a 64-bit processor bus and a set of 128 registers, the 64-bit design of IPF allows access to a very large memory (VLM) and exploits features in EPIC. While a specific example implementation of one embodiment is described below for the IPF architecture, embodiments described herein for establishing a spare processor for recovering from LOL detected for a boot processor are not limited in application to an IPF architecture, but may be applied as well to other architectures (e.g., 32-bit processor architectures, etc.).

[0031] Processor architecture generally comprises corresponding supporting firmware, such as firmware **18** of system **10**. For example, as described further below in conjunction with the specific example of FIG. 2, the IPF processor architecture comprises such supporting firmware as Processor Abstraction Layer (PAL), System Abstraction Layer (SAL), and Extended Firmware Interface (EFI). Such supporting firmware may enable, for example, the OS to access a particular function implemented for the processor. For instance, the OS may query the PAL as to the size of the cache implemented for the processor, etc. Other well-known functions provided by the supporting firmware (SAL, EFI) include, for example: (a) performing I/O configuration accesses to discover and program the I/O Hardware (SAL_PCI_CONFIG_READ and SAL_PCI_CONFIG-WRITE); (b) retrieving error log data from the platform following a Machine Check Abort (MCA) event (SAL_GET_STATE_INFO); (c) accessing persistent store configuration data stored in non-volatile memory (EFI variable services: GetNextVariableName, GetVariable and SetVariable); and accessing the battery-backed real-time clock/calendar (EFI GetTime and SetTime). Accordingly, the supporting firmware, such as the PAL, is implemented to provide an interface to the processor(s) for accessing the functionality provided by such processor(s). Each of those interfaces provide standard, published procedure calls that are supported.

[0032] In the example system **10** of FIG. 1, firmware **18** utilizes the above-mentioned hot spare technique for recovering from LOL for a boot processor. That is, processor **15** is designated as a hot spare for boot processor **12**, and such processor **15** is held as an idle, spare processor and used in the manner described further herein for assuming the role of boot processor in the event that a LOL is detected for boot processor **12**. Of course, as described further herein, the above-mentioned technique in which a designated spare is not held in reserve but is instead dynamically transformed into a spare processor may be implemented by firmware **18** in alternative embodiments. As described further herein, in certain embodiments, a device tree is stored to a data storage device (e.g., SRAM) such that it is accessible to the firmware **18**, wherein the device tree includes information identifying processor **12** as the system's boot processor and identifying processor **15** as the spare. The information specifying the boot processor and/or spare may be received as input from a user in certain embodiments. For instance, in certain embodiments a user interface is provided that enables a system administrator to designate the boot processor and/or spare processor in the multi-processor system. In other embodiments, the firmware **18** selects the boot processor and/or spare.

[0033] Suppose that during the system boot-up procedure, master processor **12**A assumes the role of boot processor, as shown in the example of FIG. 1. In this example embodiment, lockstep processor pair **15** is held as a hot spare for recovering from a LOL that may be detected for the boot processor **12**A. Of course, additional lockstep processor pairs may be included in system **10** (not specifically shown in the example of FIG. 1), and those additional lockstep processor pairs, which are not the system boot processor, may recover from LOL in a different manner that does not utilize a hot spare processor. Thus, in accordance with certain embodiments hereof, hot spare processor pair **15** is not needed for recovering from LOL detected for any non-boot processor, but may instead be used for recovery of LOL only if such LOL is detected for the boot processor (processor **12**A in this example).

[0034] Firmware **18** is included in this example, and upon detection of LOL by any of error detect logics **13**A-**13**C, firmware **18** determines, in operational block **101**, whether the detected LOL is a recoverable LOL. That is, firmware **18** determines in block **101** whether the detected LOL is of a type from which the firmware can recover lockstep for the lockstep processor pair **12** without crashing the system. As described further herein, lockstep is recoverable for certain detected LOLs (which may be referred to as "recoverable LOLs"), while lockstep is not recoverable for other detected LOLs (which may be referred to as "non-recoverable LOLs"). If the lockstep is not recoverable from the detected LOL, then in the example of FIG. 1 firmware **18** crashes the system in block **102**.

[0035] If the lockstep is recoverable, then operation advances to block **103** whereat the firmware determines whether the processor for which the LOL was detected is the system's boot processor. Because the LOL is detected for the lockstep processor pair **12** in this example, firmware **18** determines whether master processor **12**A is the system's boot processor. This can be determined, for example, by accessing the device tree **203** of FIG. 2 (described further below). The device tree may, in certain implementations, include a field for each processor indicating whether such processor is the system's boot processor. This field in the device tree may be set by the firmware during the boot-up process to identify the corresponding processor that is used as the system's boot processor or it may be pre-set by a system administrator (e.g., via a user interface), as examples. If determined in block **103** that the master processor **12**A is not the system's boot processor, then operation advances in this example to block **104**, and a lockstep recovery process is performed that does not utilize hot spare **15**.

[0036] In the specific example shown in FIG. 1, if the lockstep is determined to be recoverable in block **101** and the processor for which the LOL is detected is determined in block **103** to not be the system boot processor, firmware **18** cooperates with OS **11** via standard OS methods to recover the lockstep. For instance, in the example embodiment of FIG. 1, Advanced Configuration and Power Interface (ACPL) methods are used by firmware **18** to cooperate with OS **11**. Accordingly, no processor or platform specific knowledge is required to be embedded in OS **11**, but instead any ACPI-compatible OS may be used, including without limitation HP-UX and Open VMS operating systems. Example techniques that may be used for recovering from LOL for non-boot processors of a system are described further in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION," the disclosure of which is incorporated herein by reference.

[0037] In the example embodiment of FIG. 1, if determined in block **103** that the processor for which the LOL is detected is not the system boot processor, firmware **18** triggers OS **11** to idle the master processor **12**A in operational block **104**. In this embodiment, firmware **18** utilizes an ACPI method **105** to "eject" master processor **12**A, thereby triggering OS **11** to idle the master processor **12**A (i.e., stop scheduling tasks for the processor). Of course, by idling master processor **12**A, slave processor **12**B will in turn be idled. Thus, idling master processor **12**A results in idling the lockstep processor pair **12**. In this example embodiment, OS **11** is not aware of the presence of slave processor **12**B, but is instead aware of master processor **12**A. The interface of lockstep processor pair **12** to bus **16** manages copying to slave processor **12**B the instructions that are directed by OS **11** to master processor **12**A. Thus, firmware **18** need not direct OS **11** to eject slave processor **12**B, as OS **11** is not aware of such slave

processor **12**B in this example implementation. Again, by idling master processor **12**A, slave processor **12**B is also idled as it merely receives copies of the instructions directed to master processor **12**A. Of course, if in a given implementation OS **11** is aware of slave **12**B, firmware **18** may be implemented to also direct OS **11** to idle such slave processor **12**B in a manner similar to that described for idling master processor **12**A.

[0038] Firmware **18** then attempts to recover lockstep for the lockstep processor pair **12** in block **106**. For instance, firmware **18** resets the processor pair **12**. During such reset of processor pair **12**, system **10** can continue to operate on its remaining available processors. Once the processor pair **12** is reset and lockstep is recovered, firmware **18** reintroduces master processor **12**A to OS **11** in operational block **107**. In this embodiment, firmware **18** updates the ACPI device table information for master processor **12**A to indicate that such master processor **12**A is "present, functioning and enabled." As discussed in the ACPI 2.0 specification for the _STA status method of a device, the _STA (status) object returns the status of a device, which can be one of the following: enabled, disabled, or removed. In this respect, in the result code (bitmap) bit **0** is set if the device is present; bit **1** is set if the device is enabled and decoding its resources; bit **2** is set if the device should be shown in the UI; bit **3** is set if the device is functioning properly (cleared if the device failed its diagnostics); bit **4** is set if the battery is present; and bits **5**-**31** are reserved. A device can only decode its hardware resources if both bits **0** and **1** are set. If the device is not present (bit **0** cleared) or not enabled (bit **1** cleared), then the device must not decode its resources. Bits **0**, **1** and **3** are the "present, enabled and functioning" bits mentioned above. Firmware **18** utilizes an ACPI method **108** to trigger OS **11** to "check for" master processor **12**A, thereby reintroducing the master processor **12**A to OS **11**. As a result of checking for master processor **12**A, OS **11** will recognize that such master processor **12**A is again available and will thus begin scheduling tasks for master processor **12**A once again.

[0039] In view of the above, a recovery technique is provided that does not require utilizing a spare processor for recovering from LOL detected for non-boot processors. For various reasons, in certain system architectures problems arise in attempting to idle (or eject) the boot processor from the system. Thus, in certain embodiments, a hot spare processor is used for recovering from LOL for the system's boot processor. For instance, in the example of FIG. 1, if determined in block **103** that the master processor **12**A is the system's boot processor, operation advances to block **109** whereat the state of the "good" processor in the lockstep processor pair **12** is copied over to each of processors **15**A and **15**B in the spare lockstep processor pair **15**. Example techniques that may be utilized by firmware **18** for switching the state of a "good" processor in the lockstep processor pair 12 in accordance with certain embodiments is described further in co-pending U.S. patent application Ser. No. 10/187,833 (published as publication no. US 2004/0006722 A1) filed Jul. 3, 2002 (now U.S. Pat. No. 7,085,959) and titled "METHOD AND APPARATUS FOR RECOVERY FROM LOSS OF LOCK STEP" and in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/972,588 (U.S. Published Patent Application 20060090064) 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", the disclosures of which are each hereby incorporated herein by reference.

[0040] Thus, operational block **109** essentially makes the spare processor pair **15** the system's boot processor, and then in block **110** firmware **18** resets the lockstep processor pair **12** and reestablishes its lockstep. Once lockstep is reestablished for lockstep processor pair **12**, that pair is held in operational block **111** as a hot spare for the new boot processor pair **15**. That is, tirrnware **18** updates the device tree **203** of FIG. 2 to reflect that lockstep processor pair **12** is a hot spare for the boot processor pair **15**. Thus, should a recoverable LOL be detected for the new boot processor **15**, the above process may be used by firmware **18** to make the spare lockstep processor pair **12** the boot processor and then recover lockstep for pair **15**. An example technique for resetting the lockstep processor pair **12** and reestablishing its lockstep in which the lockstep processor pair **12** is then held as a hot spare is described further in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/972,796 U.S. Published Patent Application 20060107106) titled "SYSTEM AND METHOD FOR MAINTAINING IN A MULTI-PROCESSOR SYSTEM A SPARE PROCESSOR THAT IS IN LOCKSTEP FOR USE IN RECOVERING FROM LOSS OF LOCKSTEP FOR ANOTHER PROCESSOR", the disclosure of which is hereby incorporated herein by reference.

[0041] Embodiments described herein enable a spare processor to be designated for a boot processor, such as spare processor **15** in the above example. In certain implementations, a system administrator can designate through a user interface the boot processor, and, in some cases, further designate the corresponding spare for such boot processor. In other implementations, a system administrator may designate the boot processor and firmware selects the appropriate spare processor. In other embodiments, the firmware may select a boot processor and the corresponding spare processor for such boot processor. The information identifying the boot processor and the spare processor to be used for recovery from LOL detected for the boot processor is stored to the system in a data storage device (e.g., SRAM) that is accessible by the system's firmware to enable the firmware to identify those processors.

[0042] FIG. 2 shows a block diagram of one embodiment of the above system **10**, which is implemented for the IPF processor architecture and is labeled as system $\mathbf{10}_A$. The quintessential model of the traditional IPF architecture is given in the *Intel IA-64 Architecture Software Developer's Manual, Volume 2: IA-64 System Architecture*, in section 11.1 *Firmware Model*, the disclosure of which is hereby incorporated herein by reference. Accordingly, in this example embodiment of system $\mathbf{10}_A$, firmware **18**, labeled as firmware $\mathbf{18}_A$, includes processor abstraction layer (PAL) **201** and platform/system abstraction layer (SAL) **202**. In general, PAL **201** is firmware provided by Intel for its processors, and SAL **202** is developed by an original equipment manufacturer (OEM) for the specific system/platform in which the processors are to be employed. PAL **201**, SAL **202**, as well as an extended firmware interface (EFI) layer (not shown), together provide, among other things, the processor and system initialization for an OS boot in an IPF system.

[0043] The boot-up process of a traditional IPF system, for example, proceeds as follows: When the system is first powered on, there are some sanity checks (e.g., power on self-test) that are performed by microprocessors included in the system platform, which are not the main system processors that run applications. After those checks have passed, power and clocks are given to a boot processor (which may, for example, be master processor **12**A). The boot processor begins executing code out of the system's Read-Only Memory (ROM) (not specifically shown in FIG. 2). The code that executes is the PAL **201**, which gets control of system **10**. PAL **201** executes to acquire all of the processors in system $\mathbf{10}_A$ (recall that there may be many lockstep processor pairs, such as pairs **12** and **15**) such that the processors begin executing concurrently through the same firmware.

[0044] After it has performed its duty of initializing the processor(s), PAL **201** passes control of system $\mathbf{10}_A$ to SAL **202**. It is the responsibility of SAL **202** to discover what hardware is present on the system platform, and initialize it to make it available for the OS **11**. When main memory is initialized and functional, the firmware $\mathbf{18}_A$ is copied into the main memory. Then, control is passed to EFI (not shown), which is responsible for activating boot devices, which typically includes the disk. The EFI reads the disk to load a program into memory, typically referred to as an operating system loader. The EFI loads the OS loader into memory, and then passes it control of system $\mathbf{10}_A$ by branching the boot processor into the entry point of such OS loader program.

[0045] The OS loader program then uses the standard firmware interfaces to discover and initialize system $\mathbf{10}_A$ further for control. One of the things that the OS loader typically has to do in a multi-processor system is to retrieve control of the other processors (those processors other than the boot processor). For instance, at this point in a multi-processor system, the other processors may be executing in do-nothing loops. In an ACPI-compatible system, OS **11** makes ACPI calls to parse the ACPI tables to discover the other processors of a multi-processor system in a manner as is well-known in the art. Then OS **11** uses the firmware interfaces to cause those discovered processors to branch into the operating system code. At that point, OS **11** controls all of the processors and the firmware $\mathbf{18}_A$ is no longer in control of system $\mathbf{10}_A$.

[0046] As OS **11** is initializing, it has to discover from the firmware $\mathbf{18}_A$ what hardware is present at boot time. And in the ACPI standards, it also discovers what hardware is present or added or removed at run-time. Further, the supporting firmware (PAL, SAL, and EFI) are also used during system runtime to support the processor. For example, OS **11** may access a particular function of master processor **12**A via the supporting firmware

18$_A$, such as querying PAL **201** for the number, size, etc., of the processor's cache **14**A. Some other well-known firmware functions that OS **11** may employ during runtime include: (a) PAL **201** may be invoked to configure or change processor features such as disabling transaction queuing (PAL_BUS_SET_FEATURES); (b) PAL **201** may be invoked to flush processor caches (PAL_CACHE_FLUSH); (c) SAL **202** may be invoked to retrieve error logs following a system error (SAL_GET_STATE_INFO, SAL_CLEAR_STATE_INFO); (d) SAL **202** may be invoked as part of hot-plug sequences in which new I/O cards are installed into the hardware (SAL_PCI_CONFIG_READ, SAL_PCI_CONFIG_WRIT); (e) EFI may be invoked to change the boot device path for the next time the system reboots (SetVariable); (f) EFI may be invoked to change the clock/calendar hardware settings; and (g) EFI may be invoked to shutdown the system (ResetSystem).

[0047] A "device tree" is provided, which is shown as device tree **203** in this example. Device tree **203** is stored in SRAM (Scratch RAM) on the cell, which is RAM that is reinitialized. Firmware **18**A builds the device tree **203** as it discovers what hardware is installed in the system. Firmware then converts this information to the ACPI tables format and presents it to OS **11** so that OS **11** can know what is installed in the system. The ACPI device tables (not shown) are only consumed by OS **11** at boot time, so they are never updated as things change. For OS **11** to find the current status, it calls an ACPI "method" to discover the "current status". The _STA method described above is an example of such an ACPI method. When _STA is called, the AML can look for properties on the device specified in the firmware device tree and convert that into the Result Code bitmap described above. So, if lockstep has been lost on a processor, firmware **15**A will set the device tree property that indicates loss of lockstep, then when OS **11** calls _STA for that device, the "lockstep lost" property directs the AML code to return to "0" in the "functioning properly" bit so that OS **11** can know there is a problem with that processor.

[0048] If a lost lockstep is recovered in accordance with the recovery technique described herein, firmware **18**A can indicate that lockstep has been recovered in the device tree **203**. Then when _STA is called on that device responsive to the OS receiving the "check for device" ACPI method, the present, enabled and functioning bits will all be set and OS **11** will know the CPU is safe to use.

[0049] Further, device tree **203** includes information designating which processor is the boot processor, as well as a spare processor for the boot processor. A simple example of device tree **203** is shown below in Table 1:

[0050]

TABLE 1

Device	Status	Lockstep Enabled	Boot Processor?	Spare for Boot Processor?
Processor A	Present, Enabled, and Functioning	Yes	Yes	No
Processor B	N/A	Yes	No	Yes
Processor C	Present, Enabled, and Functioning	Yes	No	No
. . .	. . .	. . .	. . .	. . .

[0051] In the above example of Table 1, 3 processors "Processor A," "Processor B," and "Processor C" are identified in the device tree as being included in the multi-processor system. The status of each of processors Processor A and Processor B is identified as "Present, Enabled, and Functioning." In this example Processor B is a hot spare, so it is held in reserve and does not show up in the static ACPI tables that are built when the system boots. Accordingly the OS does not know about Processor B and does not have a device handle to call the _STA method on. As described above, the _STA method is a known ACPI method for checking the status of the corresponding device. Therefore, the status of Processor B is designated N/A (not applicable) in this example because the _STA method will not be called for this processor as long as it is the hot spare. Of course, upon such spare assuming the role of boot processor, its status is updated to "Present, Enabled, and Functioning." In other embodiments in which Processor B is an active spare, the OS will be aware of such processor and accordingly its status may be also indicated as "Present, Enabled, and Functioning."

[0052] Further, lockstep is identified in this example as enabled for these processors. In certain embodiments, the system administrator can designate, via user interface **20**, whether lockstep mode is to be enabled for the system's processors. Further, Processor A is designated as the system's boot processor in the example device tree shown in Table 1, and Processor B is designated as the spare processor. Thus, upon detection of LOL for one of the processors in the system, the firmware can access this device tree to determine if the processor for which LOL is detected is the system's boot processor; and if determined that the processor for which LOL is detected is the system's boot processor, the firmware can further determine from this device tree the designated spare processor to use in the lockstep recovery process.

[0053] In the example of FIG. 2, a user interface **20** is included, which allows a user (e.g., system administrator) to designate the system's boot processor and spare processor. In certain embodiments, the user interface **20** further allows the user to designate other information, such as whether lockstep mode is enabled for the processors. User interface **20** may be implemented as a command line interface or as a graphical user interface (GUI), as examples. As one example, the user interface **20** may be implemented as an EFI application that provides a command line interface for inputting commands. Such command line interface may support commands for assigning the boot processor and/or spare processor, as well as performing such tasks as configuring and deconfiguring processors, deconfiguring bad memory, specifying which OS to boot, etc. This command line interface implements a typical command <argument> format and has hooks into the firmware to change settings in NVRAM. As another example, the user interface **20** may be implemented within "parmgr", which is an HP-UX application that controls how partitions are configured, memory interleave, CPU configuration, etc. Such parmgr provides a GUI interface and a command line interface. The GUI interface presents a "worksheet" that has the current values filled in and allows the system administrator to update/change those values and then "apply" the changes. At that time, parmgr pushes the changes into NVRAM so they will be honored on the next reset. Various other implementations of a user interface **20** may be employed in alternative embodiments.

[0054] One example system in which firmware **18**$_A$ of FIG. 2 may be employed is shown in FIG. 3. FIG. 3 shows system **30** that includes multi-processor cells A-D, labeled **31**$_A$-**31**$_D$, respectively. Cell A **31**$_A$ is shown in more detail, and should be understood that cells B-D **31**$_B$-**31**$_D$ have substantially the same architecture as that of cell A **31**$_A$ in this example. The architecture of cell A **31**$_A$ described hereafter corresponds to that of Hewlett-Packard's Superdome™ systems. As shown, multi-processor cell A **31**$_A$ includes coherency controller **304** that is communicatively coupled to two buses, labeled **305** and **306**, respectively. A plurality of processor modules are included within cell A **31**$_A$, shown as CPU **0**, CPU **1**, CPU **2** and CPU **3** (and labeled **300**-**303**, respectively). More particularly, a plurality of processors modules are communicatively coupled to each of the buses **305** and **306**. In this example, CPU **0** (**300**) and CPU **1** (**301**) are each communicatively coupled to the first bus **305**, and CPU **2** (**302**) and CPU **3** (**303**) are each communicatively coupled to the second bus **306**. It should be understood that each of CPU **0** (**300**), CPU **1** (**301**), CPU **2** (**302**), and CPU **3** (**303**) is in actuality a lockstep processor pair, such as the lockstep processor pair **12** of FIGS. 1-3. Thus, for instance, CPU **0** (**300**) includes both a master processor and a slave processor, as with processors **12**A and **12**B included in the lockstep processor pair **12** of FIGS. 1-2.

[0055] Coherency controller **304** decodes the address an interrupt is targeted towards and determines which bus (**305** or **306**) the interrupt packet should be delivered to. Each CPU on the destination bus sees the interrupt and compares the interrupt target address with its internal "LID" register to determine if it is the targeted CPU. If the match is correct, the CPU responds to the interrupt. If the match is not correct, the CPU ignores the interrupt packet.

[0056] As shown, the example firmware $\mathbf{18}_A$ described above with FIGS. 1-2 may be employed to manage the recovery from detected LOLs for the processors of cells A-D ($\mathbf{31}_A$-$\mathbf{31}_D$). In this example architecture, when designating a spare for the boot processor, such spare is selected to be another processor communicatively coupled to the same bus as the boot processor. For instance, suppose that during the boot-up of system **30**, CPU **2** (**302**) is designated as the boot processor. Accordingly, firmware $\mathbf{18}_A$ will identify another processor that is communicatively coupled to bus **306** to designate as a spare for such boot processor, such as CPU **3** (**303**). During runtime if a LOL is detected for CPU **2** (**302**), then firmware $\mathbf{18}_A$ may operate according to the process described above in FIG. 1 to switch the designation of boot processor to a designated spare, CPU **3** (**303**). In performing this switch the LID register of the CPU that lost lockstep is copied into the LID register of the spare CPU and the spare CPU pair will then assume the ID of the failed CPU and start responding to interrupts directed to that ID. The LID of the failed CPU is quickly re-written to a different value so that it will no longer respond to interrupts directed to that ID. CPU **2** (**302**) can then be reset to recover its lockstep, and such CPU **2** (**302**) may then be designated as a spare for the new boot processor CPU **3** (**303**).

[0057] It should be recognized that in the example system of FIG. 3, if a hot spare were used for recovering from LOL for all processors, rather than just for the boot processor, much of the potential processing resources would be required to be held in reserve as spare resources. For instance, a hot spare would be required on each of the buses of each cell. Because there are two processors coupled to each bus in this example (i.e., CPU **0** and CPU **1** coupled to bus **305**, and CPU **2** and CPU **3** coupled to bus **306**), reserving a hot spare processor for each bus would result in holding half of the total processing resources in reserve. This is an undesirably expensive and wasteful solution. Thus, according to the illustrated embodiment, a hot spare processor is reserved only for the boot processor. Thus, continuing with the above example in which CPU **2** is the boot processor and CPU **3** is held as a "hot" spare, both of CPU **0** and CPU **1** are active (i.e., neither are held in spare). Further, none of the processors of cells B-D need to be held in reserve as hot spares. In the event of a recoverable LOL occurring for any of non-boot processors of the cells, the lockstep recovery process of FIG. 1 (i.e., operational blocks **104**-**108**) can be utilized without requiring a spare for lockstep recovery for those non-boot processors.

[0058] While in the above example embodiment, a hot spare CPU is maintained for recovery from LOL for the system's boot processor, in an alternative embodiment a hot spare need not be held as such, but rather a processor can be dynamically made a "spare" when needed for recovery from LOL encountered for the system boot processor. That is, assuming as in the above example of FIG. 3 that CPU **2** is the system's boot processor, CPU **3** does not have to be held in reserve (idled) as a hot spare for CPU **2**, but instead CPU **3** can be active and dynamically idled and made a "spare" upon detection of LOL in CPU **2**. Accordingly, CPU **3** may be designated as a spare for the boot processor, but such CPU **3** need not be held in reserve during normal system operation. Instead, the CPU **3** may be dynamically (during system run-time) transformed into a spare processor (e.g., by causing the OS to idle such processor) in the event that a LOL is detected for the boot processor.

[0059] Thus, the resource of this spare CPU need not be wasted during normal runtime, but can be dynamically turned into a spare processor that is available for recovering from LOL encountered on CPU **2** in the manner described above. As an example of this embodiment, 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. The system boot processor role would be assumed by the idled healthy processor pair, just as described above in the example in which the healthy CPU **3** is held idle for the system boot processor, CPU **2**. The only difference being that now the firmware had to take some initial action to turn the healthy processor pair into a "hot spare" by idling it and making it appear to the OS as though it was not functioning. Thus, the LIDs would be swapped. That is, the LID register of the boot processor that lost lockstep is copied into the LID register of the now spare processor pair (CPU **3** in this example), and the LID register of this spare processor pair would be copied into the LID register of the original boot processor. Lockstep is then reestablished on the original boot processor, and it can be reintroduced to the OS with the ID of the original non-boot processor that replaced it as the system boot processor. The processor that was formerly the boot processor may be designated, in device tree **203**, as the spare for the now boot processor.

[0060] Turning to FIG. 4, an operational flow diagram for establishing a hot spare for the boot processor according to one embodiment is shown. The operational loop of FIG. 4 is performed for each processor in a multi-processor system. In block **401**, the processor boots in lockstep mode. As described above, in certain embodiments, whether the processors are to operate in lockstep mode is configurable in the device tree, and thus the firmware may consult the device tree to determine whether to boot each processor in lockstep mode. In operational block **402**, the firmware determines whether the processor is configured to be a hot spare for the system's boot processor. As described above, this determination can be made by accessing the device tree maintained in the firmware. If the processor is not configured to be a hot spare, then operation advances to block **403** whereat the normal boot process continues for this processor. If, on the other hand, the processor is configured to be a hot spare, then operation advances to block **404** whereat hot spare specific properties are set in the device tree. If configured to be a hot spare, the firmware will hide the processor from the system's OS by setting properties on the node associated with this CPU in the firmware device tree. Typically, nodes of type "CPU" are reported to the OS in the ACPI tables. In this case, the "type" will be set in the device tree to "hot_spare" for the designated spare processor so that this processor is not reported in the ACPI namespace. Thus, when the system firmware builds its ACPI tables, it looks for devices of type "CPU", and the hot spare is not placed in the ACPI tables because its type is set to "hot_spare" in this example. Accordingly, the OS will not be aware of the hot spare processor.

[0061] In operational block **405**, the hot spare processor executes in an idle loop waiting for a PAL call to set state. Such PAL call to "set state" copies the saved state from the processor that experienced LOL (i.e., the state of the "good" one of the lockstep pair of processors in the processor module that experienced LOL) into the hot spare processor. Specifically, PAL copies all the needed information from memory into the internal registers of the hot spare processor so that the hot spare processor becomes a new active processor, transforming the role of the hot spare to boot processor. This call does not change the device tree settings. Accordingly, with the spare processor hidden from the OS (i.e., the OS is unaware of the hot spare), the hot spare sits in an idle loop until it is needed for recovering from LOL detected for the boot processor, at which time the system firmware interrupts its loop and assigns it the role of boot processor, as described above.

[0062] FIG. 5 shows an operational flow diagram for establishing an active spare for the boot processor and assigning such active spare the role of boot processor responsive to detecting LOL for the boot processor in accordance with one embodiment. Operational blocks **501**-**504** of FIG. 5 are performed for each processor in a multi-processor system for establishing an active spare in the system. In block **501**, the processor boots in lockstep mode. As described above, in certain embodiments, whether the processors are to operate in lockstep mode is configurable in the device tree, and thus the firmware may consult the device tree to determine whether to boot each processor in lockstep mode. In operational block **502**, the firmware determines whether the processor is configured to be an active spare for the system's boot processor. As described above, this determination can be made by accessing the device tree maintained in the firmware. If the processor is not configured to be an active spare, then operation advances to block **503** whereat the normal boot process continues for this processor. If, on the other hand, the processor is configured to be an active spare, then operation advances to block **504** whereat active spare specific properties are set in the device tree. For instance, in this example, the device tree would include a property called "Active Spare" and this property would be set to TRUE for the active spare processor. If LOL

is detected on the boot processor, firmware searches the devices in the device tree for the "Active Spare=TRUE" condition to find the designated active spare processor. Thereafter, operation returns to operational block **503**, whereat the processor continues its normal boot and is introduced to the OS, rather than being hidden from the OS as the above-described hot spare.

[0063] Operational block **505** represents the system's normal operation until such time as a LOL is detected for the boot processor. Upon error detect logic detecting a LOL for a processor, the error detect logic invokes the system's firmware. For instance, the LOL may invoke the PAL, which in turn may report the LOL to the SAL. The firmware (e.g., SAL) determines if the processor for which the LOL is detected is designated (e.g., in the device tree) as the boot processor. If so, then the firmware determines in block **505** that LOL is detected for the boot processor. Thus, in operational block **506**, the firmware determines (e.g., from the device tree) 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. In operational block **507**, the firmware assigns the active spare processor the role of boot processor. The state of the "good" one of the lockstep processor pair of the boot processor for which the LOL is detected is copied into both of the lockstep processors of the active spare processor. Further, 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.

[0064] In block **508**, the firmware attempts to recover lockstep for the former boot processor for which LOL was detected in block **505**. In operational block **509**, the firmware determines whether the lockstep recovery for the former boot processor is successful. In certain embodiments, if the lockstep cannot be recovered in operational block **508** (e.g., the processor fails its self-test when being reset), then that processor is simply not reintroduced back to the OS and in operational block **510** a message is generated by the system to provide notice of the loss of this processor and that the boot processor no longer has a spare. If the lockstep cannot be reestablished in block **508**, then if a LOL occurs in the new boot processor there will not be a spare processor for taking over its role (and thus the system may crash). If desired, the system administrator can schedule an orderly shutdown and re-boot in order to establish a new boot processor that has a hot spare available and/or schedule service of the system (e.g., for replacing the failed processor).

[0065] If the lockstep recovery is successful, then the former boot processor is assigned the role of active spare (e.g., by designating it as such in the device tree) in block **511**, and the firmware triggers the OS to recognize the active spare as available for executing instructions. Operation then returns to block **505**, and if a LOL is detected for the current boot processor, the procedure of blocks **506**-**512** may be repeated.

[0066] Turning to FIG. 6, a general flow diagram for operation of certain embodiments is shown. In block **601**, the role of boot processor is assigned to one of a plurality of processors in a system. As described above, the role of boot processor may be so assigned in a device tree. Such role of boot processor may be assigned by a user (e.g., via a user interface) in certain embodiments, and assigned by the system's firmware in other embodiments.

[0067] In block **602**, the role of spare processor is assigned to another one of the processors. As described above, in certain embodiments the spare processor may be a hot spare, while in other embodiments the spare processor may be an active spare. As also described above, the role of spare processor may be so assigned in a device tree. Such role of spare processor may be assigned by a user (e.g., via a user interface) in certain embodiments, and assigned by the system's firmware in other embodiments.

[0068] In operational block **603**, the system firmware, responsive to detecting LOL for the boot processor, changes the assignment of the role of boot processor to the spare processor. This change in the role of boot processor occurs dynamically during system runtime, rather than requiring that the system be interrupted (e.g., shutdown and restarted, or crashed).

[0069] Certain systems include a plurality of partitions, wherein each partition includes a boot processor and OS. It should be understood that the above techniques for establishing a spare processor for use in assuming the role of boot processor in the event that a LOL is detected for the boot processor may be employed on each partition. Thus, a device tree may designate a boot processor and spare (e.g., active or hot) for each partition.

Original

Classifications

IPC	CPC	US Class	ECLA	ICO
G06F11/00	G06F11/1641	714/13	G06F11/20P10	S06F11:16C6
	G06F11/1654	714/11	G06F11/16C12	S06F11:20P6
	G06F11/2025		G06F11/20P2C	
	G06F11/2028		G06F11/20P2E	
	G06F11/2038			
	G06F11/2043			

Persons

Ultimate Owner

Original
FORAS TECHNOLOGIES (/api/control?client=&query=uo:("FORAS%20TECHNOLOGIES")&autosearch=true)

Original Assignees

Original	Standardized	Normalized
Scott L. Michaelis (/api/control?client=&query=pa:("Scott%20L.%20Michaelis")&autosearch=true)		
Plano, Texas, United States of America, US		

Anurupa Rajkumari (/api/control?client=&query=pa:("Anurupa%20Rajkumari")&autosearch=true)
Round Rock, Texas, United States of America, US

Current Assignees

Original	Standardized	Normalized
SONRAÍ MEMORY, LTD. (/api/control?client=&query=pacu:("SONRA%C3%8D%20MEMORY%2C%20LTD.")&autosearch=true) SUITE 23, THE HYDE BUILDING, THE PARK, CARRICKMINE, DUBLIN 18, DUBLIN, IRELAND	SONRAI MEMORY (/api/control?client=&query=pacus:("SONRAI%20MEMORY")&autosearch=true)	

Inventor(s)

Original
Scott L. Michaelis (/api/control?client=&query=in:("Scott%20L.%20Michaelis")&autosearch=true) Plano, Texas, United States of America, US
Anurupa Rajkumari (/api/control?client=&query=in:("Anurupa%20Rajkumari")&autosearch=true) Round Rock, Texas, United States of America, US

Legal information

Standardized Post issuance INPADOC

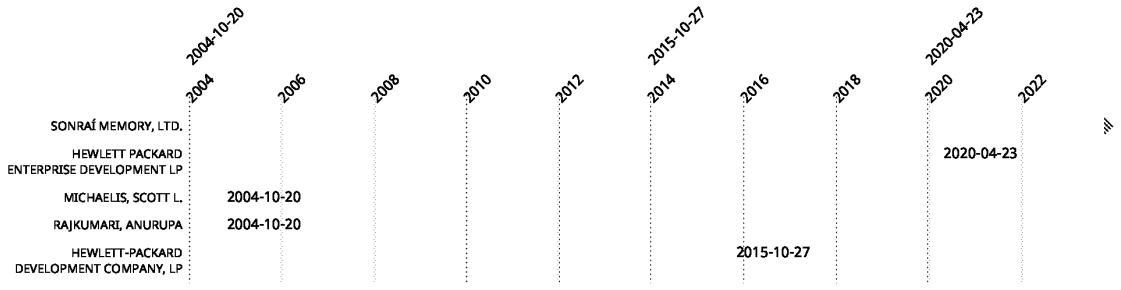
Legal status

Event	Date	Description
Granted	2009-12-01	GRANTED
Filed	2004-10-25	FILED

Designated states

Date	Event code	Description

Ownership



Payment status

Date	Status	Event code	Description
2021-05-19	Fee paid	M1553	PAYMENT OF MAINTENANCE FEE, 12TH YEAR, LARGE ENTITY.

2017-06-28	Fee paid	M1552	PAYMENT OF MAINTENANCE FEE, 8TH YEAR, LARGE ENTITY.
2013-03-08	Fee paid	M1551	PAYMENT OF MAINTENANCE FEE, 4TH YEAR, LARGE ENTITY.

Litigation/Opposition

Docket number	Docket description	Court	Date Filed
1:23cv640	Foras Technologies Ltd. V. Nissan Motor Company, Ltd. Et Al		2023-06-05
2:23cv150	Foras Technologies Ltd. V. Toyota Motor North America, Inc. Et Al		2023-04-04

Page: 1

Pharmaceuticals

Product	Type	Active Ingredient	Applicant	Application No.	Application Type	Approval Date	Patent Expiry Date	Patent usage code	Exclusivity Code	Exclusivity Date	Dosage	Strength	Product Number	Route	RLD	Pa Re

No pharmaceuticals data

Citations

Display in a list (/api/control?client=&citationsSearch=_)

☐	backward	US4358823A	By other than the examiner	1982-11-09	Double redundant processor	TRW	
☐	backward	US5226152A	By other than the examiner	1993-07-06	Functional lockstep arrangement for redundant processors	MOTOROLA	
☐	backward	US5249188A	By other than the examiner	1993-09-28	Synchronizing two processors as an integral part of fault detection	COMMUNICATION SYSTEMS	
☐	backward	US5537655A	By other than the examiner	1996-07-16	Synchronized fault tolerant reset	BOEING	
☐	backward	US5675794A	By other than the examiner	1997-10-07	Method and apparatus for configuring multiple agents in a computer system	INTEL	
☐	backward	US5751932A	By other than the examiner	1998-05-12	Fail-fast, fail-functional, fault-tolerant multiprocessor system	TANDEM COMPUTERS	
☐	backward	US5758058A	By other than the examiner	1998-05-26	Apparatus and method for initializing a master/checker fault detecting microprocessor	INTEL	
☐	backward	US5764660A	By other than the examiner	1998-06-09	Processor independent error checking arrangement	ELSAG INT	
☐	backward	US5896523A	By other than the examiner	1999-04-20	Loosely-coupled, synchronized execution	MARATHON TECHN	
☐	backward	US5915082A	Cited by the examiner	1999-06-22	Error detection and fault isolation for lockstep processor systems	LOCKHEED	
☐	backward	US5991900A	By other than the examiner	1999-11-23	Bus controller	SUN MICROSYSTEMS	
☐	backward	US6065135A	By other than the examiner	2000-05-16	Error detection and fault isolation for lockstep processor systems	LOCKHEED	
☐	backward	US6141770A	By other than the examiner	2000-10-31	Fault tolerant computer system	GEN DYNAMICS INF SYSTEMS	
☐	backward	US6148348A	By other than the examiner	2000-11-14	Bridge interfacing two processing sets operating in a lockstep mode and having a posted write buffer storing write operations upon detection of a lockstep error	SUN MICROSYSTEMS	
☐	backward	US6220581B1	By other than the examiner	2001-04-24	Bicycle seat gas spring adjustment system	MUELLER THOMAS L	
☐	backward	US6263452B1	By other than the examiner	2001-07-17	Fault-tolerant computer system with online recovery and reintegration of redundant components	COMPAQ COMPUTER	
☐	backward	US6330356B1	By other than the examiner	2001-12-11	Dynamic visual registration of a 3-D object with a graphical model	ROCKWELL SCIENCE CENTER	
☐	backward	US6438687B2	Cited by the	2002-08-20	Method and apparatus for improved storage of computer system configuration	MICRON TECHNOLOGY	

			examiner		information		
☐	backward	US6473869B2	By other than the examiner	2002-10-29	Fault resilient/fault tolerant computing	MARATHON TECHNOLOGIES	
☐	backward	US6560682B1	By other than the examiner	2003-05-06	System and method for terminating lock-step sequences in a multiprocessor system	INTEL	

1 - 20 of 102

Non-patent literature

"System and Method for Maintaining in a Multi-Processor System a Spare Processor That Is in Lockstep for Use in Recovering From Loss of Lockstep for Another Processor".
"System and Method for Reintroducing a Processor Module to an Operating System After Lockstep Recovery".
"System and Method for Using Information Relating to a detected Loss of Lockstep for Determining a Responsive Action".
"System and Method for Providing Firmware Recoverable Lockstep Protection".
"System and Method for Configuring Lockstep Mode of a Prcessor Module".
"System and Method for Reestablishing Lockstep for a Processor Module for Which Loss of Lockstep Is Detected".
"System and Method for System Firmware Causing an Operating System to Idle a Processor".
"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".
Notice of Allowance dated Feb. 12, 2009 for U.S. Appl. No. 10/972,588 in the United States.
Notice of Allowance dated Sep. 23, 2008 for U.S. Appl. No. 10/972,588 in the United States.
Office Action dated Mar. 10, 2008 for U.S. Appl. No. 10/972,588 in the United States.
Office Action dated Sep. 12, 2007 for U.S. Appl. No. 10/972,588 in the United States.
Notice of Allowance for U.S. Appl. No. 10/972,835, dated Nov. 20, 2008 in the United States.
Notice of Allowance for U.S. Appl. No. 10/972,835, dated Aug. 1, 2008 in the United States.
Office Action for U.S. Appl. No. 10/972,835, dated Mar. 3, 2008 in the United States.
Office Action for U.S. Appl. No. 10/972,835, dated Jul. 12, 2007 in the United States.
Office Action for U.S. Appl. No. 10/973,003, dated Mar. 5, 2008 in the United States.
Office Action for U.S. Appl. No. 10/973,003, dated Sep. 12, 2007 in the United States.
Notice of Allowance for U.S. Appl. No. 10/973,076, dated Nov. 28, 2008 in the United States.
Notice of Allowance for U.S. Appl. No. 10/973,076, dated Jul. 16, 2008 in the United States.
Notice of Allowance for U.S. Appl. No. 10/973,076, dated Jun. 3, 2008 in the United States.
Final Office Action for U.S. Appl. No. 10/973,076, dated Feb. 27, 2008 in the United States.
Office Action for U.S. Appl. No. 10/973,076, dated Sep. 12, 2007 in the United States.
Notice of Allowance for U.S. Appl. No. 10/972,888, dated Nov. 14, 2007 in the United States.
Office Action for U.S. Appl. No. 10/972,888, dated Jul. 18, 2007 in the United States.
Notice of Allowance for U.S. Appl. No. 10/973,004, dated Aug. 1, 2007 in the United States.
Office Action for U.S. Appl. No. 10/973,004, dated Apr. 10, 2007 in the United States.
Final Office Action for U.S. Appl. No. 10/973,075, dated May 12, 2009 in the United States.
Office Action for U.S. Appl. No. 10/973,075, dated Oct. 29, 2008 in the United State.
Office Action for U.S. Appl. No. 10/973,075, dated Mar. 12, 2008 in the United States.
Office Action for U.S. Appl. No. 10/973,075, dated Jul. 2, 2007 in the United States.
Notice of Allowance for U.S. Appl. No. 10/972,796, dated Dec. 19, 2007 in the United States.
Office Action for U.S. Appl. No. 10/972,796, dated Jul. 12, 2007 in the United States.
Shoichi Araki, Real-Time Tracking of Multiple Moving Object Contours in a Moving Camera Image Sequence, vol. E83-D IEICE Trans. Inf. & Syst, Jul. 7, 2000, 9 pages. (/intapi/scopus/0034229432) **42** Citations in Scopus
Dong Kyu Kim, Active Object Tracking Using Image Mosaic Background, 4 pages.
Jiang-Bin Zheng, Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, Nov. 4-5, 2002, 4 pages.
Jong Ryul Kim, Automatic Localization and Trackign of Moving Objects Using Adaptive Snake Algorithm, ICICS-PCM 2003, Dec. 15-18, 2003, 5 pages, Singapore.
Jianhoa Meng, Cveps-A Compressed Video Editing and Parsing System, Nov. 18, 1996, pp. 1-11.
Final Office Action for U.S. Appl. No. 10/972,835, dated Nov. 21, 2007 in the United States.
Office Action for U.S. Appl. No. 10/973,075, dated Dec. 13, 2007 in the United States.
Notice of Allowance, dated Jul. 14, 2009, U.S. Appl. No. 10/972,588, 12 pages.

Similar documents

Display in a list (/api/control?client=&similarSearch=_)

The similarity check is done by matching this document to all documents in the database. Only the title and abstract text are matched, resulting in a score. The 10 highest-scoring documents are displayed here.

☐ All

☐ 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

Publication number	US20060090064A1
Publication date	2006-04-27
Application date	2004-10-25

According to one embodiment, a method comprises detecting loss of lockstep (LOL) for a processor in a multi-processor system. The method further comprises 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. In another embodiment, a method comprises system firmware determining that an LOL is detected for a lockstep pair of

processors that are assigned the role of boot processor in a system. The method further comprises determining one of the lockstep pair of processors that is not the cause of the LOL, and copying the state of the determined one of the lockstep pair of processors that is not the cause of the LOL to a spare processor. The method further comprises switching the role of boot processor to the spare processor.

☐ 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

Publication number US7624302B2
Publication date 2009-11-24
Application date 2004-10-25

According to one embodiment, a method comprises detecting loss of lockstep (LOL) for a processor in a multi-processor system. The method further comprises 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. In another embodiment, a method comprises system firmware determining that an LOL is detected for a lockstep pair of processors that are assigned the role of boot processor in a system. The method further comprises determining one of the lockstep pair of processors that is not the cause of the LOL, and copying the state of the determined one of the lockstep pair of processors that is not the cause of the LOL to a spare processor. The method further comprises switching the role of boot processor to the spare processor.

☐ System and method for maintaining in a multi-processor system a spare processor that is in lockstep for use in recovering from loss of lockstep for another processor

Publication number US7366948B2
Publication date 2008-04-29
Application date 2004-10-25

According to one embodiment, a method comprises assigning a first processor of a multi-processor system a role of spare processor for at least a second processor, and responsive to detecting loss of lockstep (LOL) for any of the at least a second processor, the first processor replaces the processor for which the LOL is detected. The method further comprises reestablishing lockstep for the processor for which the LOL is detected, and assigning the processor having its lockstep reestablished the role of spare processor for at least the first processor.

☐ System and method for maintaining in a multi-processor system a spare processor that is in lockstep for use in recovering from loss of lockstep for another processor

Publication number US20060107106A1
Publication date 2006-05-18
Application date 2004-10-25

According to one embodiment, a method comprises assigning a first processor of a multi-processor system a role of spare processor for at least a second processor, and responsive to detecting loss of lockstep (LOL) for any of the at least a second processor, the first processor replaces the processor for which the LOL is detected. The method further comprises reestablishing lockstep for the processor for which the LOL is detected, and assigning the processor having its lockstep reestablished the role of spare processor for at least the first processor.

☐ System and method for reestablishing lockstep for a processor module for which loss of lockstep is detected

Publication number US20060107116A1
Publication date 2006-05-18
Application date 2004-10-25

According to one embodiment, a method comprises, responsive to detection of loss of lockstep (LOL) for a processor module in a system, firmware requesting an operating system to idle the processor module. The method further comprises the operating system idling the processor module and returning control of the processor module to the firmware. According to one embodiment, a method comprises an operating system idling a processor module for which A loss of lockstep (LOL) is detected, and system firmware receiving control of the processor module. The method further comprises the system firmware determining whether a LOL was detected for the processor module, and if determined that LOL was detected for the processor module, the system firmware resetting the processor module to reestablish lockstep for the processor module.

☐ System and method for configuring lockstep mode of a processor module

Publication number US7308566B2
Publication date 2007-12-11
Application date 2004-10-25

A system comprises a processor module that supports lockstep mode of operation. The system further comprises non-volatile data storage having stored thereto configuration information specifying whether the processor module is desired to operate in lockstep mode. A method comprises storing configuration information to non-volatile data storage of a system, wherein the configuration information specifies whether lockstep mode of operation is desired to be enabled or disabled for a processor module of the system. The method further comprises causing, by the system, the processor module to have its lockstep mode enabled or disabled as specified by the configuration information.

☐ System and method for configuring lockstep mode of a processor module

Publication number	US20060107117A1
Publication date	2006-05-18
Application date	2004-10-25

A system comprises a processor module that supports lockstep mode of operation. The system further comprises non-volatile data storage having stored thereto configuration information specifying whether the processor module is desired to operate in lockstep mode. A method comprises storing configuration information to non-volatile data storage of a system, wherein the configuration information specifies whether lockstep mode of operation is desired to be enabled or disabled for a processor module of the system. The method further comprises causing, by the system, the processor module to have its lockstep mode enabled or disabled as specified by the configuration information.

☐ System and method for using information relating to a detected loss of lockstep for determining a responsive action

Publication number	US20060107114A1
Publication date	2006-05-18
Application date	2004-10-25

According to one embodiment, a method comprises detecting a loss of lockstep (LOL) for a processor module. The method further comprises determining a type of LOL that is detected, and, based at least in part on the determined type of LOL, determining a responsive action to take for the LOL. According to one embodiment, a method comprises detecting a loss of lockstep (LOL) for a processor module. The method further comprises using information identifying at least one of type of the detected LOL and source of the detected LOL to determine a responsive action to take for the LOL.

☐ System and method for using information relating to a detected loss of lockstep for determining a responsive action

Publication number	US7516359B2
Publication date	2009-04-07
Application date	2004-10-25

According to one embodiment, a method comprises detecting a loss of lockstep (LOL) for a processor module. The method further comprises determining a type of LOL that is detected, and, based at least in part on the determined type of LOL, determining a responsive action to take for the LOL. According to one embodiment, a method comprises detecting a loss of lockstep (LOL) for a processor module. The method further comprises using information identifying at least one of type of the detected LOL and source of the detected LOL to determine a responsive action to take for the LOL.

☐ System and method for reintroducing a processor module to an operating system after lockstep recovery

Publication number	US20060107111A1
Publication date	2006-05-18
Application date	2004-10-25

According to one embodiment, a method comprises, responsive to detection of loss of lockstep (LOL) for a processor module in a system, setting status information stored to the system for the processor module to indicate that the processor module has an error. The method further comprises reestablishing lockstep for the processor module without shutting down the system's operating system, and updating the status information for the processor module to indicate that the processor module no longer has the error. The method further comprises causing the system's operating system to recognize that the processor module having its lockstep reestablished is available for processing.

1
1



User Name: Karen Ward
Date and Time: Thursday, September 7, 2023 9:17:00AM EDT
Job Number: 205231206

Document (1)

1. *1:23cv640, Foras Technologies Ltd. V. Nissan Motor Company, Ltd. Et Al*
   Client/Matter: -None-
   Search Terms: US PAT 7627781 or 7,627,781
   Search Type: Terms and Connectors
   Narrowed by:

Content Type	Narrowed by
Dockets	-None-

1:23cv640, Foras Technologies Ltd. V. Nissan Motor Company, Ltd. Et Al

US District Court Docket

United States District Court, Texas Western

(Austin)

This case was retrieved on 09/07/2023

Header

Case Number: 1:23cv640
Date Filed: 06/06/2023
Assigned To: Judge Robert Pitman
Nature of Suit: Patent (830)
Cause: Patent Infringement
Lead Docket: None
Other Docket: 6:22cv00336
Jurisdiction: Federal Question

Class Code: Open
Statute: 35:271
Jury Demand: Plaintiff
Demand Amount: $0
NOS Description: Patent

Participants

Litigants

Foras Technologies Ltd.
Plaintiff

Attorneys

Jonathan Randy Yim
LEAD ATTORNEY;ATTORNEY TO BE NOTICED
BC Law Group, P.C.
200 Madison Avenue 24th Floor
New York, NY 10016
USA
516-359-9668 Email:Jyim@b-Clg.Com

Drew B Hollander
ATTORNEY TO BE NOTICED
BC Law Group, P.C.
200 Madison Avenue 24th Floor
New York, NY 10016
USA
212-951-0100 Email:Dhollander@b-Clg.Com

Seth Raymond Hasenour
ATTORNEY TO BE NOTICED
BC Law Group, P.C.
607 Terrell Hill Drive Ste Unit B
Austin, TX 78704
USA
516-359-9668 Email:Shasenour@b-Clg.Com

Brett E Cooper
ATTORNEY TO BE NOTICED
BC LAW GROUP, P.C.
200 Madison Avenue, 24th Floor
New York, NY 10016

Karen Ward

Litigants

Nissan Motor Company, Ltd.
Defendant
ZF Friedrichshafen AG
Defendant

Attorneys

USA
516-359-9668 Email:Bcooper@b-Clg.Com

Proceedings

#	Date	Proceeding Text	Source
1	06/06/2023	COMPLAINT (Filing fee $ 402 receipt number ATXWDC-17522685), filed by Foras Technologies Ltd.. (Attachments: # 1 Exhibit 1, # 2 Exhibit 2, # 3 Exhibit 3, # 4 Exhibit 4, # 5 Exhibit 5, # 6 Exhibit 6, # 7 Exhibit 7, # 8 Exhibit 8, # 9 Exhibit 9, # 10 Exhibit 10, # 11 Exhibit 11, # 12 Civil Cover Sheet)(Cooper, Brett) (Entered: 06/06/2023)	
2	06/06/2023	Notice of Filing of Patent/Trademark Form (AO 120). AO 120 forwarded to the Director of the U.S. Patent and Trademark Office. (Cooper, Brett) (Entered: 06/06/2023)	
3	06/06/2023	RULE 7 DISCLOSURE STATEMENT filed by Foras Technologies Ltd.. (Cooper, Brett) (Entered: 06/06/2023)	
4	06/06/2023	NOTICE Notice of Related Cases by Foras Technologies Ltd. (Cooper, Brett) (Entered: 06/06/2023)	
5	06/06/2023	REQUEST FOR ISSUANCE OF SUMMONS by Foras Technologies Ltd.. as to Nissan Motor Company, Ltd. (Cooper, Brett) (Entered: 06/06/2023)	
6	06/06/2023	REQUEST FOR ISSUANCE OF SUMMONS by Foras Technologies Ltd.. as to ZF Friedrichshafen AG (Cooper, Brett) (Entered: 06/06/2023)	
	06/06/2023	Case assigned to Judge Robert Pitman. CM WILL NOW REFLECT THE JUDGE INITIALS AS PART OF THE CASE NUMBER. PLEASE APPEND THESE JUDGE INITIALS TO THE CASE NUMBER ON EACH DOCUMENT THAT YOU FILE IN THIS CASE. (pg) (Entered: 06/09/2023)	
	06/06/2023	If ordered by the court, all referrals and consents in this case will be assigned to Magistrate Judge Howell (pg) (Entered: 06/09/2023)	
7	06/09/2023	Summons Issued as to Nissan Motor Company, Ltd. (pg) (Entered: 06/09/2023)	
8	06/09/2023	Summons Issued as to ZF Friedrichshafen AG. (pg) (Entered: 06/09/2023)	
9	06/09/2023	Report on Patent/Trademark sent to U.S. Patent and Trademark Office with Complaint. (pg) (Entered: 06/09/2023)	

Patents

Number	Title	Issued	Class	Subclass
7,502,958	System and method for providing firmware recoverable lockstep protection	03/10/2009	714	12
7,624,302	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	11/24/2009	714	12
7,627,781	System and method for establishing a spare processor for recovering from loss of lockstep in a boot processor	12/01/2009	714	13

Karen Ward

Copyright © LexisNexis CourtLink, Inc. All Rights Reserved.
*** THIS DATA IS FOR INFORMATIONAL PURPOSES ONLY ***

End of Document



User Name: Karen Ward
Date and Time: Thursday, September 7, 2023 9:18:00AM EDT
Job Number: 205231264

Document (1)

1. *2:23cv150 Foras Technologies Ltd V. Toyota Motor North America, Inc. Et Al*
   Client/Matter: -None-
   Search Terms: US PAT 7627781 or 7,627,781
   Search Type: Terms and Connectors
   Narrowed by:

Content Type	Narrowed by
Dockets	-None-

2:23cv150, Foras Technologies Ltd. V. Toyota Motor North America, Inc. Et Al

US District Court Docket

United States District Court, Texas Eastern

(Marshall)

This case was retrieved on 09/07/2023

Header

Case Number: 2:23cv150
Date Filed: 04/05/2023
Assigned To: District Judge Rodney Gilstrap
Nature of Suit: Patent (830)
Cause: Patent Infringement
Lead Docket: None
Other Docket: 2:22cv00219, 2:23cv00314
Jurisdiction: Federal Question

Class Code: Open
Statute: 35:145
Jury Demand: Both
Demand Amount: $0
NOS Description: Patent

Participants

Litigants

Foras Technologies Ltd.
Plaintiff

Attorneys

Drew Bennett Hollander
ATTORNEY TO BE NOTICED
BC Law Group, P.C.
200 Madison Avenue 24th Floor
New York, NY 10016
USA
212-951-0100 Email:Dhollander@b-Clg.Com

Jonathan Randy Yim
ATTORNEY TO BE NOTICED
BC Law Group, P.C.
200 Madison Avenue 24th Floor
New York, NY 10016
USA
516-359-9668 Email:Jyim@b-Clg.Com

Justin Kurt Truelove
ATTORNEY TO BE NOTICED
Truelove Law Firm
207 N Wellington P O Box 1409
Marshall, TX 75671
USA
903-938-8321 Fax: 903-215-8510
Email:Kurt@trueloveiawfirm.Com

Seth Raymond Hasenour
ATTORNEY TO BE NOTICED

Karen Ward

Litigants	Attorneys
	BC Law Group, P.C. 607 Terrell Hill Drive Ste Unit B Austin, TX 78704 USA 516-359-9668 Email:Shasenour@b-Clg.Com Brett E Cooper ATTORNEY TO BE NOTICED BC Law Group, P.C. 200 Madison Avenue 24th Floor New York, NY 10016 USA 516-359-9668 Email:Bcooper@b-Clg.Com
Toyota Motor North America, Inc. [Terminated: 07/12/2023] **Defendant**	Steven Routh LEAD ATTORNEY;ATTORNEY TO BE NOTICED [Terminated: 07/12/2023] Orrick, Herrington & Sutcliffe LLP - Washington DC Columbia Center 1152 15th Street, Nw Washington, DC 20005-1706 USA 202-339-8509 Fax: 202-339-8500 Email:Srouth@orrick.Com
Denso Corporation **Defendant**	Steven Routh LEAD ATTORNEY;ATTORNEY TO BE NOTICED Orrick, Herrington & Sutcliffe LLP - Washington DC Columbia Center 1152 15th Street, Nw Washington, DC 20005-1706 USA 202-339-8509 Fax: 202-339-8500 Email:Srouth@orrick.Com Andrew Thompson (Tom) Gorham ATTORNEY TO BE NOTICED Gillam & Smith LLP 102 N College, Suite 800 Tyler, TX 75702 USA 9039348450 Fax: 9039349257 Email:Tom@gillamsmithlaw.Com Melissa Richards Smith ATTORNEY TO BE NOTICED Gillam & Smith, LLP 303 South Washington Avenue Marshall, TX 75670 USA 903-934-8450 Fax: 903-934-9257 Email:Melissa@gillamsmithlaw.Com Sten Anker Jensen ATTORNEY TO BE NOTICED Orrick, Herrington & Sutcliffe LLP - Washington DC Columbia Center 1152 15th Street, Nw Washington, DC 20005-1706 USA 202-339-8436 Fax: 202-339-8500 Email:Sjensen@orrick.Com
Kia Corporation Consolidated Civil Action 2:23cv219 | **Consol Defendant**	Douglas E Lumish LEAD ATTORNEY;ATTORNEY TO BE NOTICED Latham & Watkins LLP - Menlo Park 140 Scott Drive

Litigants	Attorneys
	Menlo Park, CA 94025-1008 USA 650-328-4600 Fax: 650-463-2600 Email:Doug.Lumish@lw.Com Adam Michael Greenfield ATTORNEY TO BE NOTICED Latham & Watkins LLP - D.C. 555 Eleventh Street, Nw Suite 1000 Washington, DC 20004 USA 202.637.2200 Fax: 202.637.2201 Email:Adam.Greenfield@lw.Com Andrew Thompson (Tom) Gorham ATTORNEY TO BE NOTICED Gillam & Smith LLP 102 N College, Suite 800 Tyler, TX 75702 USA 9039348450 Fax: 9039349257 Email:Tom@gillamsmithlaw.Com Jacob Vannette ATTORNEY TO BE NOTICED Latham & Watkins LLP 555 Eleventh Street, Nw Suite 1000 Washington, DC 20004 USA 202-637-2200 Email:Jake.Vannette@lw.Com Melissa Richards Smith ATTORNEY TO BE NOTICED Gillam & Smith, LLP 303 South Washington Avenue Marshall, TX 75670 USA 903-934-8450 Fax: 903-934-9257 Email:Melissa@gillamsmithlaw.Com
Kia America, Inc. Consolidated Civil Action 2:23cv219 | **Consol Defendant**	Douglas E Lumish LEAD ATTORNEY;ATTORNEY TO BE NOTICED Latham & Watkins LLP - Menlo Park 140 Scott Drive Menlo Park, CA 94025-1008 USA 650-328-4600 Fax: 650-463-2600 Email:Doug.Lumish@lw.Com Adam Michael Greenfield ATTORNEY TO BE NOTICED Latham & Watkins LLP - D.C. 555 Eleventh Street, Nw Suite 1000 Washington, DC 20004 USA 202.637.2200 Fax: 202.637.2201 Email:Adam.Greenfield@lw.Com Andrew Thompson (Tom) Gorham ATTORNEY TO BE NOTICED

Litigants

Attorneys

Gillam & Smith LLP
102 N College, Suite 800
Tyler, TX 75702
USA
9039348450 Fax: 9039349257
Email:Tom@gillamsmithlaw.Com

Jacob Vannette
ATTORNEY TO BE NOTICED
Latham & Watkins LLP
555 Eleventh Street, Nw Suite 1000
Washington, DC 20004
USA
202-637-2200 Email:Jake.Vannette@lw.Com

Melissa Richards Smith
ATTORNEY TO BE NOTICED
Gillam & Smith, LLP
303 South Washington Avenue
Marshall, TX 75670
USA
903-934-8450 Fax: 903-934-9257
Email:Melissa@gillamsmithlaw.Com

Toyota Motor North America, Inc.
Consolidated Civil Action 2:23-321 |
Consol Defendant

Steven Routh
ATTORNEY TO BE NOTICED
Orrick, Herrington & Sutcliffe LLP - Washington DC
Columbia Center 1152 15th Street, Nw
Washington, DC 20005-1706
USA
202-339-8509 Fax: 202-339-8500 Email:Srouth@orrick.Com

Proceedings

#	Date	Proceeding Text	Source
1	04/05/2023	COMPLAINT against Denso Corporation, Toyota Motor North America, Inc. (Filing fee $ 402 receipt number ATXEDC-9437538.), filed by Foras Technologies Ltd.. (Attachments: # 1 Exhibit 1, # 2 Exhibit 2, # 3 Exhibit 3, # 4 Exhibit 4, # 5 Exhibit 5, # 6 Exhibit 6, # 7 Exhibit 7, # 8 Exhibit 8, # 9 Exhibit 9, # 10 Civil Cover Sheet)(Cooper, Brett) (Entered: 04/05/2023)	
2	04/05/2023	Notice of Filing of Patent/Trademark Form (AO 120). AO 120 mailed to the Director of the U.S. Patent and Trademark Office. (Cooper, Brett) (Entered: 04/05/2023)	
3	04/05/2023	CORPORATE DISCLOSURE STATEMENT filed by Foras Technologies Ltd. (Cooper, Brett) (Entered: 04/05/2023)	
	04/05/2023	Case Assigned to District Judge Rodney Gilstrap. (nkl,) (Entered: 04/05/2023)	
	04/05/2023	In accordance with the provisions of 28 USC Section 636(c), you are hereby notified that a U.S. Magistrate Judge of this district court is available to conduct any or all proceedings in this case including a jury or non-jury trial and to order the entry of a final judgment. The form Consent to Proceed Before Magistrate Judge is available on our website. All signed consent forms, excluding pro se parties, should be filed electronically using the event Notice Regarding Consent to Proceed Before Magistrate Judge. (nkl,) (Entered: 04/05/2023)	

#	Date	Proceeding Text	Source
4	04/05/2023	SUMMONS Issued as to Denso Corporation c/o Denso International America, Inc.c/o Corporation Service Company d/b/a CSC Lawyers Incorporating Service Company, Toyota Motor North America, Inc. c/o CT Corporation System. (Attachments: # 1 Summons(es))(nkl,) (Entered: 04/05/2023)	
5	04/11/2023	SUMMONS Returned Executed by Foras Technologies Ltd.. Toyota Motor North America, Inc. served on 4/6/2023, answer due 4/27/2023. (Cooper, Brett) (Entered: 04/11/2023)	
6	04/12/2023	SUMMONS Returned Executed by Foras Technologies Ltd.. Denso Corporation served on 4/6/2023, answer due 4/27/2023. (Cooper, Brett) (Entered: 04/12/2023)	
7	04/27/2023	***DEFICIENT DOCUMENT***Joint MOTION for Extension of Time to File Response to Complaint by Denso Corporation. (Attachments: # 1 Text of Proposed Order)(Routh, Steven) Modified on 4/27/2023 (ch,). (Entered: 04/27/2023)	
8	04/27/2023	***DEFICIENT DOCUMENT***Joint MOTION for Extension of Time to File Response to Complaint by Toyota Motor North America, Inc.. (Attachments: # 1 Text of Proposed Order)(Routh, Steven) Modified on 4/27/2023 (ch,). (Entered: 04/27/2023)	
	04/27/2023	NOTICE of Deficiency regarding the 7 submitted Needs Certificate of Conference. Correction should be made by one business day (ch,) (Entered: 04/27/2023)	
	04/27/2023	NOTICE of Deficiency regarding the 8 submitted Needs Certificate of Conference. Correction should be made by one business day (ch,) (Entered: 04/27/2023)	
9	04/27/2023	Joint MOTION for Extension of Time to File Response to Complaint by Denso Corporation. (Attachments: # 1 Text of Proposed Order)(Routh, Steven) (Entered: 04/27/2023)	
10	04/27/2023	Joint MOTION for Extension of Time to File Response to Complaint by Toyota Motor North America, Inc.. (Attachments: # 1 Text of Proposed Order)(Routh, Steven) (Entered: 04/27/2023)	
11	04/28/2023	ORDER granting 9 Motion for Extension of Time to File Response to Complaint. Signed by District Judge Rodney Gilstrap on 4/28/2023. (nkl,) (Entered: 05/01/2023)	
12	04/28/2023	ORDER granting 10 Motion for Extension of Time to File Response to Complaint. Signed by District Judge Rodney Gilstrap on 4/28/2023. (nkl,) (Entered: 05/01/2023)	
13	05/23/2023	NOTICE of Attorney Appearance by Drew Bennett Hollander on behalf of Foras Technologies Ltd. (Hollander, Drew) (Entered: 05/23/2023)	
14	05/23/2023	NOTICE of Attorney Appearance by Seth Raymond Hasenour on behalf of Foras Technologies Ltd. (Hasenour, Seth) (Entered: 05/23/2023)	
15	05/23/2023	NOTICE of Attorney Appearance by Jonathan Randy Yim on behalf of Foras Technologies Ltd. (Yim, Jonathan) (Entered: 05/23/2023)	
16	06/13/2023	CONSOLIDATION ORDER - The above-captioned cases are hereby ORDERED to be CONSOLIDATED for all pretrial issues with the LEAD CASE, Case No. 2:23-cv-00150. All parties are instructed to file any future filings in the LEAD CASE. Individual cases remain active for trial.. Signed by District Judge Rodney Gilstrap on 6/13/2023. (nkl,) (Entered: 06/13/2023)	
17	06/15/2023	ORDER setting Scheduling Conference for 7/20/2023 at 9:30 AM before District Judge Rodney Gilstrap. Signed by District Judge Rodney Gilstrap on 6/15/23. (tkd,) (Entered: 06/16/2023)	
18	06/22/2023	NOTICE of Attorney Appearance by Melissa Richards Smith on behalf of Kia America, Inc., Kia Corporation (Smith, Melissa) (Entered: 06/22/2023)	

#	Date	Proceeding Text	Source
19	06/22/2023	Joint MOTION for extension of time for Toyota and DENSO to respond to the complaint and for Foras to serve infringement contentions by Foras Technologies Ltd.. (Attachments: # 1 Proposed Order)(Cooper, Brett) (Entered: 06/22/2023)	
20	06/23/2023	ORDER granting 19 Joint MOTION for extension of time for Toyota and DENSO to respond to the complaint and for Foras to serve infringement contentions. Denso Corporation answer due 7/21/2023; Toyota Motor North America, Inc. answer due 7/21/2023.. Signed by District Judge Rodney Gilstrap on 6/23/2023. (ch,) (Entered: 06/23/2023)	
21	06/27/2023	NOTICE of Attorney Appearance by Douglas E Lumish on behalf of Kia America, Inc., Kia Corporation (Lumish, Douglas) (Entered: 06/27/2023)	
22	06/27/2023	NOTICE of Attorney Appearance by Adam Michael Greenfield on behalf of Kia America, Inc., Kia Corporation (Greenfield, Adam) (Entered: 06/27/2023)	
23	06/27/2023	NOTICE of Attorney Appearance by Jacob Vannette on behalf of Kia America, Inc., Kia Corporation (Vannette, Jacob) (Entered: 06/27/2023)	
24	06/27/2023	NOTICE of Attorney Appearance by Justin Kurt Truelove on behalf of Foras Technologies Ltd. (Truelove, Justin) (Entered: 06/27/2023)	
25	06/28/2023	NOTICE of Attorney Appearance by Justin Kurt Truelove on behalf of Foras Technologies Ltd. (Truelove, Justin) (Entered: 06/28/2023)	
26	07/03/2023	Joint MOTION FOR EXTENSIONS OF TIME REGARDING INFRINGEMENT AND INVALIDITY CONTENTIONS by Kia America, Inc., Kia Corporation. (Attachments: # 1 Proposed Order)(Smith, Melissa) (Entered: 07/03/2023)	
27	07/07/2023	ORDER granting 26 Motion for Extensions of Time Regarding Infringement and Invalidity Contentions. Signed by District Judge Rodney Gilstrap on 7/6/2023. (nkl,) (Entered: 07/07/2023)	
28	07/10/2023	ANSWER to 1 Complaint, by Toyota Motor North America, Inc..(Routh, Steven) (Entered: 07/10/2023)	
29	07/10/2023	Fed. R. Civ. P. 7.1(a)(1) Disclosure Statement filed by Toyota Motor North America, Inc. identifying Corporate Parent Toyota Motor Corporation for Toyota Motor North America, Inc.. (Routh, Steven) (Entered: 07/10/2023)	
30	07/10/2023	ANSWER to 1 Complaint, by Denso Corporation.(Routh, Steven) (Entered: 07/10/2023)	
31	07/10/2023	Fed. R. Civ. P. 7.1(a)(1) Disclosure Statement filed by Denso Corporation identifying Corporate Parent Toyota Motor Corporation for Denso Corporation. (Routh, Steven) (Entered: 07/10/2023)	
32	07/10/2023	Joint MOTION to Sever and Stay Claims Against Toyota by Denso Corporation, Toyota Motor North America, Inc.. (Attachments: # 1 Proposed Order)(Routh, Steven) (Additional attachment(s) added on 7/11/2023: # 2 Revised Proposed Order) (nkl,). (Entered: 07/10/2023)	
	07/10/2023	In accordance with the provisions of 28 USC Section 636(c), you are hereby notified that a U.S. Magistrate Judge of this district court is available to conduct any or all proceedings in this case including a jury or non-jury trial and to order the entry of a final judgment. The form Consent to Proceed Before Magistrate Judge is available on our website. All signed consent forms, excluding pro se parties, should be filed electronically using the event Notice Regarding Consent to Proceed Before Magistrate Judge. (nkl,) (Entered: 07/10/2023)	

#	Date	Proceeding Text	Source
33	07/12/2023	ORDER granting 32 Motion to Sever and Stay Claims Against Toyota. Signed by District Judge Rodney Gilstrap on 7/11/2023. (nkl,) (Entered: 07/12/2023)	
34	07/12/2023	NOTICE of Attorney Appearance by Melissa Richards Smith on behalf of Denso Corporation (Smith, Melissa) (Entered: 07/12/2023)	
35	07/19/2023	NOTICE of Attorney Appearance by Andrew Thompson (Tom) Gorham on behalf of Denso Corporation, Kia America, Inc., Kia Corporation (Gorham, Andrew) (Entered: 07/19/2023)	
	07/20/2023	Minute Entry for proceedings held before District Judge Rodney Gilstrap (Lead 2:23-cv-150; Members 2:23-cv-219; 2:23-cv-321_stayed): Scheduling Conference/Case Management Conference held on 7/20/2023. Counsel for the parties appeared. Court asked whether they consented to trial before the U.S. Magistrate Judge. Court then gave counsel Claim Construction and Jury Selection/Trial dates. (Court Reporter Shawn McRoberts) (aeb) (Entered: 07/24/2023)	
36	07/21/2023	NOTICE by Foras Technologies Ltd. of Compliance Regarding Disclosure of Asserted Claims and Infringement Contentions (Cooper, Brett) (Entered: 07/21/2023)	
37	07/24/2023	ANSWER to Complaint by Kia America, Inc., Kia Corporation.(Smith, Melissa) (Entered: 07/24/2023)	
38	07/24/2023	Fed. R. Civ. P. 7.1(a)(1) Disclosure Statement filed by Kia America, Inc., Kia Corporation (Smith, Melissa) (Entered: 07/24/2023)	
39	07/24/2023	DEMAND for Trial by Jury by Kia America, Inc., Kia Corporation. (Smith, Melissa) (Entered: 07/24/2023)	
	07/24/2023	In accordance with the provisions of 28 USC Section 636(c), you are hereby notified that a U.S. Magistrate Judge of this district court is available to conduct any or all proceedings in this case including a jury or non-jury trial and to order the entry of a final judgment. The form Consent to Proceed Before Magistrate Judge is available on our website. All signed consent forms, excluding pro se parties, should be filed electronically using the event Notice Regarding Consent to Proceed Before Magistrate Judge. (nkl,) (Entered: 07/24/2023)	
40	08/03/2023	Joint MOTION for Entry of Agreed Docket Control Order by Foras Technologies Ltd.. (Attachments: # 1 Proposed Order)(Cooper, Brett) (Entered: 08/03/2023)	
41	08/03/2023	Joint MOTION for Entry of Agreed Discovery Order by Foras Technologies Ltd.. (Attachments: # 1 Proposed Order)(Cooper, Brett) (Entered: 08/03/2023)	
42	08/09/2023	DOCKET CONTROL ORDER re 40 Joint MOTION for Entry of Agreed Docket Control Order filed by Foras Technologies Ltd.., Pretrial Conference set for 12/2/2024 at 09:00 AM before District Judge Rodney Gilstrap., Amended Pleadings due by 5/2/2024., Jury Selection set for 1/6/2025 at 09:00AM before District Judge Rodney Gilstrap., Markman Hearing set for 7/18/2024 at 09:00 AM before District Judge Rodney Gilstrap., Motions in Limine due by 11/12/2024., Proposed Pretrial Order due by 11/25/2024. Signed by District Judge Rodney Gilstrap on 8/8/2023. (nkl,) (Entered: 08/09/2023)	
43	08/09/2023	DISCOVERY ORDER re 41 Motion for Entry of Agreed Discovery Order. Signed by District Judge Rodney Gilstrap on 8/8/2023. (nkl,) (Entered: 08/09/2023)	
44	08/09/2023	Joint MOTION to Amend Docket Control Order re 42 Order,,, Terminate Motions,,, Scheduling Order,, by Foras Technologies Ltd.. (Attachments: # 1 Proposed Order)(Cooper, Brett) (Entered: 08/09/2023)	

#	Date	Proceeding Text	Source
45	08/10/2023	NOTICE of Attorney Appearance by Sten Anker Jensen on behalf of Denso Corporation (Jensen, Sten) (Entered: 08/10/2023)	
46	08/11/2023	FIRST AMENDED DOCKET CONTROL ORDER re 44 Motion to Amend Docket Control Order. Signed by District Judge Rodney Gilstrap on 8/11/2023. (nkl,) (Entered: 08/11/2023)	
47	08/17/2023	Joint MOTION for Entry of Agreed Protective Order by Foras Technologies Ltd.. (Attachments: # 1 Proposed Order)(Cooper, Brett) (Entered: 08/17/2023)	
48	08/17/2023	NOTICE by Foras Technologies Ltd. of Compliance Regarding Initial Disclosures (Cooper, Brett) (Entered: 08/17/2023)	
49	08/17/2023	NOTICE of Discovery Disclosure by Denso Corporation (Routh, Steven) (Entered: 08/17/2023)	
50	08/18/2023	NOTICE by Kia America, Inc., Kia Corporation of Compliance Regarding Initial and Additional Disclosures (Smith, Melissa) (Entered: 08/18/2023)	
51	08/21/2023	PROTECTIVE ORDER. Signed by District Judge Rodney Gilstrap on 8/21/2023. (nkl,) (Entered: 08/21/2023)	

Patents

Number	Title	Issued	Class	Subclass
7,502,958	System and method for providing firmware recoverable lockstep protection	03/10/2009	714	12
7,624,302	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	11/24/2009	714	12
7,627,781	System and method for establishing a spare processor for recovering from loss of lockstep in a boot processor	12/01/2009	714	13

Copyright © LexisNexis CourtLink, Inc. All Rights Reserved.
*** THIS DATA IS FOR INFORMATIONAL PURPOSES ONLY ***

End of Document

EXHIBIT
1001

US007627781B2

(12) **United States Patent**
Michaelis et al.

(10) **Patent No.: US 7,627,781 B2**
(45) **Date of Patent: *Dec. 1, 2009**

(54) **SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR**

(75) Inventors: **Scott L. Michaelis**, Plano, TX (US); **Anurupa Rajkumari**, Round Rock, TX (US)

(73) Assignee: **Hewlett-Packard Development Company, L.P.**, Houston, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 520 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/973,077**

(22) Filed: **Oct. 25, 2004**

(65) **Prior Publication Data**

US 2006/0107112 A1 May 18, 2006

(51) **Int. Cl.**
G06F 11/00 (2006.01)

(52) **U.S. Cl.** **714/13**; 714/11

(58) **Field of Classification Search** 714/10–12, 714/13, 31; 712/31

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,358,823	A	11/1982	McDonald et al.	
5,226,152	A	7/1993	Klug et al.	
5,249,188	A	9/1993	McDonald	
5,537,655	A	7/1996	Truong	
5,675,794	A	10/1997	Meredith	
5,751,932	A	5/1998	Horst et al.	
5,758,058	A	5/1998	Milburn	
5,764,660	A	6/1998	Mohat	
5,896,523	A	4/1999	Bissett et al.	
5,915,082	A *	6/1999	Marshall et al.	714/11
5,991,900	A	11/1999	Garnett	
6,065,135	A	5/2000	Marshall et al.	
6,141,770	A	10/2000	Fuchs	
6,148,348	A	11/2000	Garnett et al.	
6,220,581	B1	4/2001	Mueller	
6,263,452	B1	7/2001	Jewett	
6,330,356	B1	12/2001	Sundareswaran	
6,438,687	B2 *	8/2002	Klein	713/1
6,473,869	B2	10/2002	Bissett et al.	
6,560,682	B1	5/2003	Miller et al.	
6,604,177	B1	8/2003	Kondo et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

WO	WO 01 45421	6/2001

OTHER PUBLICATIONS

"System and Method for Maintaining in a Multi-Processor System a Spare Processor That Is in Lockstep for Use in Recovering From Loss of Lockstep for Another Processor".

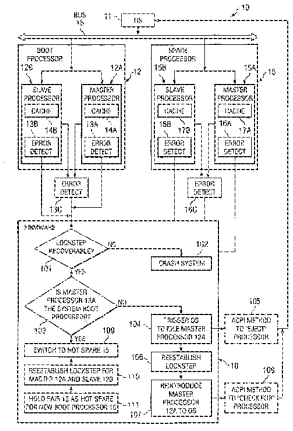
(Continued)

Primary Examiner—Dieu-Minh Le

(57) **ABSTRACT**

A system comprises a plurality of processors, and data storage storing information that assigns a role of boot processor to one of the plurality of processors and assigns a role of spare processor to another of the plurality of processors. The system further comprises logic operable, responsive to detecting loss of lockstep for the boot processor, for transferring, during system runtime, the role of boot processor to the spare processor.

30 Claims, 4 Drawing Sheets



U.S. PATENT DOCUMENTS

6,625,749	B1 *	9/2003	Quach	714/10
6,643,387	B1	11/2003	Sethuraman	
6,675,324	B2	1/2004	Marisetty et al.	
6,687,851	B1	2/2004	Somers et al.	
6,735,253	B1	5/2004	Chang	
6,754,787	B2	6/2004	Miller et al.	
6,920,581	B2 *	7/2005	Bigbee et al.	714/10
6,948,092	B2	9/2005	Kondo et al.	
6,983,018	B1	1/2006	Chih-Lung	
7,003,691	B2	2/2006	Safford et al.	
7,020,800	B2	3/2006	Fu et al.	
7,042,943	B2	5/2006	Haskell	
7,085,959	B2 *	8/2006	Safford	714/11
7,134,047	B2	11/2006	Quach	
7,155,721	B2	12/2006	Safford et al.	
7,191,292	B2	3/2007	Chaudhry et al.	
7,225,355	B2	5/2007	Yamazaki et al.	
7,237,144	B2	6/2007	Safford et al.	
7,251,746	B2 *	7/2007	Fox et al.	714/13
2002/0144177	A1	10/2002	Kondo et al.	
2002/0152420	A1	10/2002	Chaudhry et al.	
2003/0051190	A1	3/2003	Marisetty et al.	
2003/0070050	A1	4/2003	Miller et al.	
2003/0126498	A1	7/2003	Bigbee	
2003/0135711	A1	7/2003	Shoemaker	
2003/0140281	A1	7/2003	Fu et al.	
2004/0006722	A1	1/2004	Safford	
2004/0019771	A1	1/2004	Quach	
2004/0078650	A1	4/2004	Safford et al.	
2004/0078651	A1 *	4/2004	Safford et al.	714/11
2004/0153857	A1	8/2004	Yamazaki et al.	
2005/0172164	A1 *	8/2005	Fox et al.	714/13
2005/0240806	A1	10/2005	Bruckert et al.	
2005/0240811	A1 *	10/2005	Safford et al.	714/11
2005/0240829	A1	10/2005	Safford et al.	
2006/0085677	A1	4/2006	Safford et al.	
2006/0136672	A1	6/2006	Chaudhry et al.	
2006/0248384	A1 *	11/2006	Safford	714/11
2006/0248684	A1	11/2006	Vosbikian et al.	

OTHER PUBLICATIONS

"System and Method for Reintroducing a Processor Module to an Operating System After Lockstep Recovery".
"System and Method for Using Information Relating to a detected Loss of Lockstep for Determining a Responsive Action".
"System and Method for Providing Firmware Recoverable Lockstep Protection".
"System and Method for Configuring Lockstep Mode of a Processor Module".
"System and Method for Reestablishing Lockstep for a Processor Module for Which Loss of Lockstep Is Detected".
"System and Method for System Firmware Causing an Operating System to Idle a Processor".
"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".
Notice of Allowance dated Feb. 12, 2009 for U.S. Appl. No. 10/972,588 in the United States.
Notice of Allowance dated Sep. 23, 2008 for U.S. Appl. No. 10/972,588 in the United States.
Office Action dated Mar. 10, 2008 for U.S. Appl. No. 10/972,588 in the United States.
Office Action dated Sep. 12, 2007 for U.S. Appl. No. 10/972,588 in the United States.
Notice of Allowance for U.S. Appl. No. 10/972,835, dated Nov. 20, 2008 in the United States.
Notice of Allowance for U.S. Appl. No. 10/972,835, dated Aug. 1, 2008 in the United States.
Office Action for U.S. Appl. No. 10/972,835, dated Mar. 3, 2008 in the United States.
Office Action for U.S. Appl. No. 10/972,835, dated Jul. 12, 2007 in the United States.
Office Action for U.S. Appl. No. 10/973,003, dated Mar. 5, 2008 in the United States.
Office Action for U.S. Appl. No. 10/973,003, dated Sep. 12, 2007 in the United States.
Notice of Allowance for U.S. Appl. No. 10/973,076, dated Nov. 28, 2008 in the United States.
Notice of Allowance for U.S. Appl. No. 10/973,076, dated Jul. 16, 2008 in the United States.
Notice of Allowance for U.S. Appl. No. 10/973,076, dated Jun. 3, 2008 in the United States.
Final Office Action for U.S. Appl. No. 10/973,076, dated Feb. 27, 2008 in the United States.
Office Action for U.S. Appl. No. 10/973,076, dated Sep. 12, 2007 in the United States.
Notice of Allowance for U.S. Appl. No. 10/972,888, dated Nov. 14, 2007 in the United States.
Office Action for U.S. Appl. No. 10/972,888, dated Jul. 18, 2007 in the United States.
Notice of Allowance for U.S. Appl. No. 10/973,004, dated Aug. 1, 2007 in the United States.
Office Action for U.S. Appl. No. 10/973,004, dated Apr. 10, 2007 in the United States.
Final Office Action for U.S. Appl. No. 10/973,075, dated May 12, 2009 in the United States.
Office Action for U.S. Appl. No. 10/973,075, dated Oct. 29, 2008 in the United State.
Office Action for U.S. Appl. No. 10/973,075, dated Mar. 12, 2008 in the United States.
Office Action for U.S. Appl. No. 10/973,075, dated Jul. 2, 2007 in the United States.
Notice of Allowance for U.S. Appl. No. 10/972,796, dated Dec. 19, 2007 in the United States.
Office Action for U.S. Appl. No. 10/972,796, dated Jul. 12, 2007 in the United States.
Shoichi Araki, Real-Time Tracking of Multiple Moving Object Contours in a Moving Camera Image Sequence, vol. E83-D IEICE Trans. Inf. & Syst, Jul. 7, 2000, 9 pages.
Dong Kyu Kim, Active Object Tracking Using Image Mosaic Background, 4 pages.
Jiang-Bin Zheng, Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, Nov. 4-5, 2002, 4 pages.
Jong Ryul Kim, Automatic Localization and Trackign of Moving Objects Using Adaptive Snake Algorithm, ICICS-PCM 2003, Dec. 15-18, 2003, 5 pages, Singapore.
Jianhoa Meng, Cveps-A Compressed Video Editing and Parsing System, Nov. 18, 1996, pp. 1-11.
Final Office Action for U.S. Appl. No. 10/972,835, dated Nov. 21, 2007 in the United States.
Office Action for U.S. Appl. No. 10/973,075, dated Dec. 13, 2007 in the United States.
Notice of Allowance, dated Jul. 14, 2009, U.S. Appl. No. 10/972,588, 12 pages.

* cited by examiner

FIG. 1

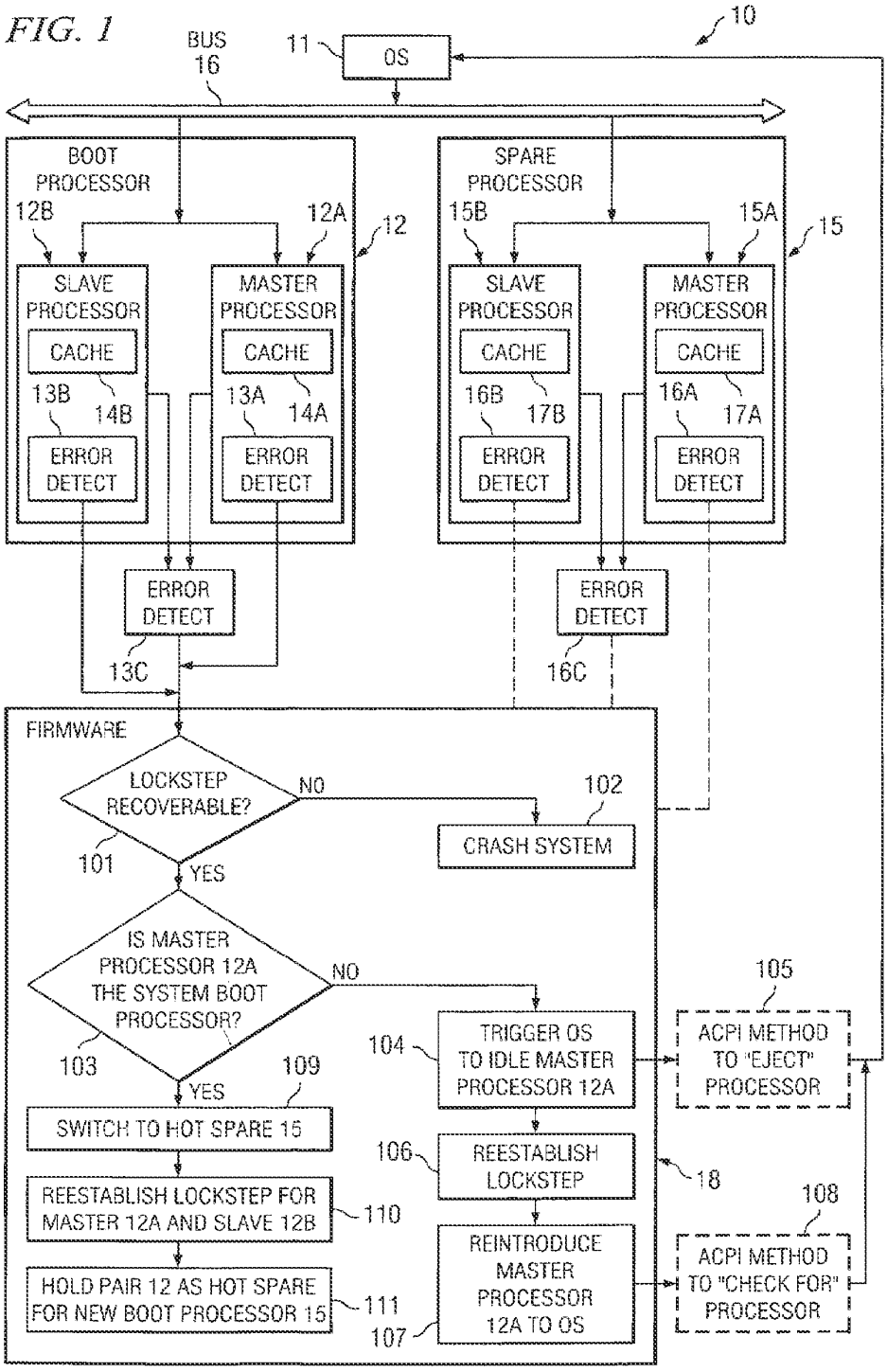

10
BUS
16
11
OS
BOOT
PROCESSOR
12B
12A
12
SPARE
PROCESSOR
15B
15A
15
SLAVE
PROCESSOR
CACHE
13B
14B
ERROR
DETECT
MASTER
PROCESSOR
CACHE
13A
14A
ERROR
DETECT
SLAVE
PROCESSOR
CACHE
16B
17B
ERROR
DETECT
MASTER
PROCESSOR
CACHE
16A
17A
ERROR
DETECT
ERROR
DETECT
13C
ERROR
DETECT
16C
FIRMWARE
LOCKSTEP
RECOVERABLE?
NO
102
CRASH SYSTEM
101
YES
IS MASTER
PROCESSOR 12A
THE SYSTEM BOOT
PROCESSOR?
NO
105
103
YES
109
104
TRIGGER OS
TO IDLE MASTER
PROCESSOR 12A
ACPI METHOD
TO "EJECT"
PROCESSOR
SWITCH TO HOT SPARE 15
106
REESTABLISH
LOCKSTEP
18
REESTABLISH LOCKSTEP FOR
MASTER 12A AND SLAVE 12B
110
108
REINTRODUCE
MASTER
PROCESSOR
12A TO OS
ACPI METHOD
TO "CHECK FOR"
PROCESSOR
HOLD PAIR 12 AS HOT SPARE
FOR NEW BOOT PROCESSOR 15
111
107

FIG. 2

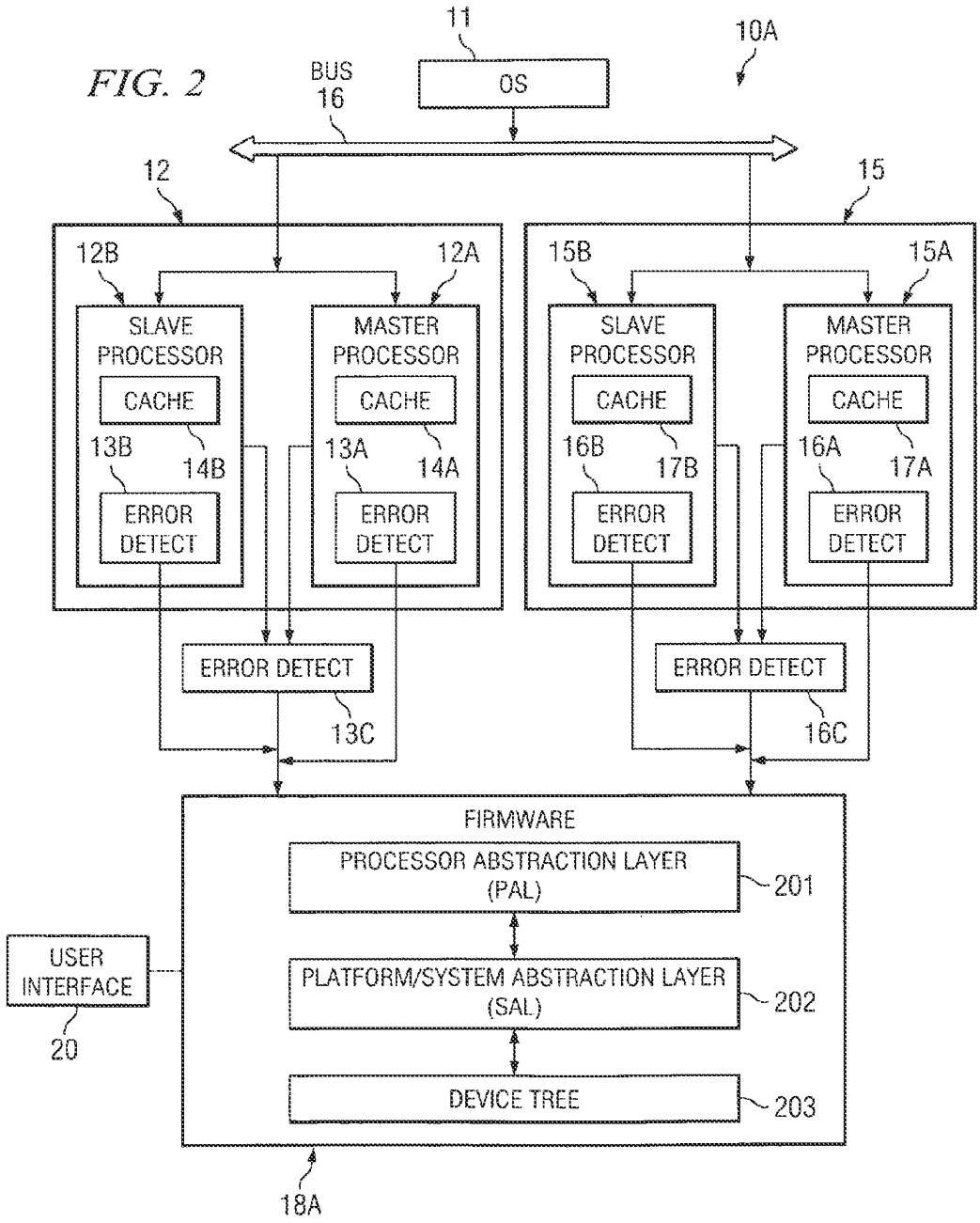

11
OS
10A
BUS
16
12
15
12B
12A
15B
15A
SLAVE PROCESSOR
CACHE
13B
14B
ERROR DETECT
MASTER PROCESSOR
CACHE
13A
14A
ERROR DETECT
SLAVE PROCESSOR
CACHE
16B
17B
ERROR DETECT
MASTER PROCESSOR
CACHE
16A
17A
ERROR DETECT
ERROR DETECT
13C
ERROR DETECT
16C
FIRMWARE
PROCESSOR ABSTRACTION LAYER (PAL)
201
PLATFORM/SYSTEM ABSTRACTION LAYER (SAL)
202
DEVICE TREE
203
USER INTERFACE
20
18A

FIG. 3

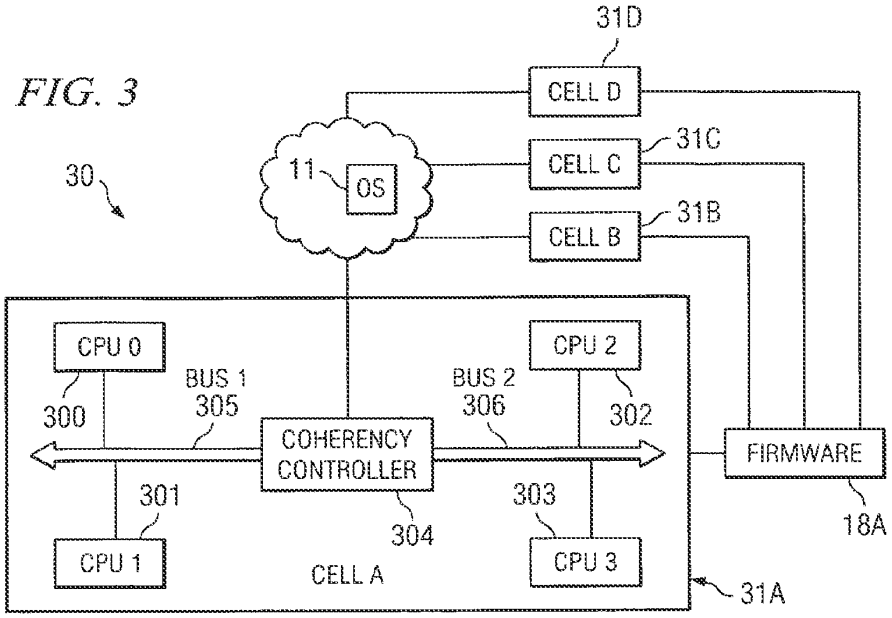


FIG. 4

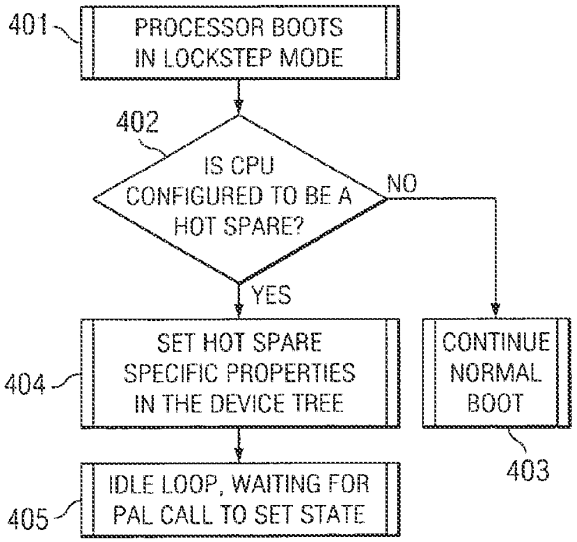


FIG. 6

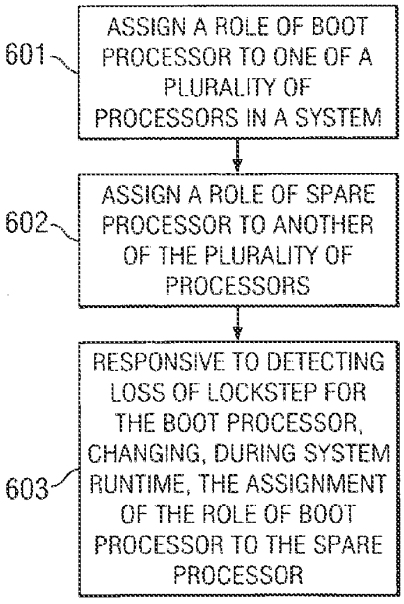
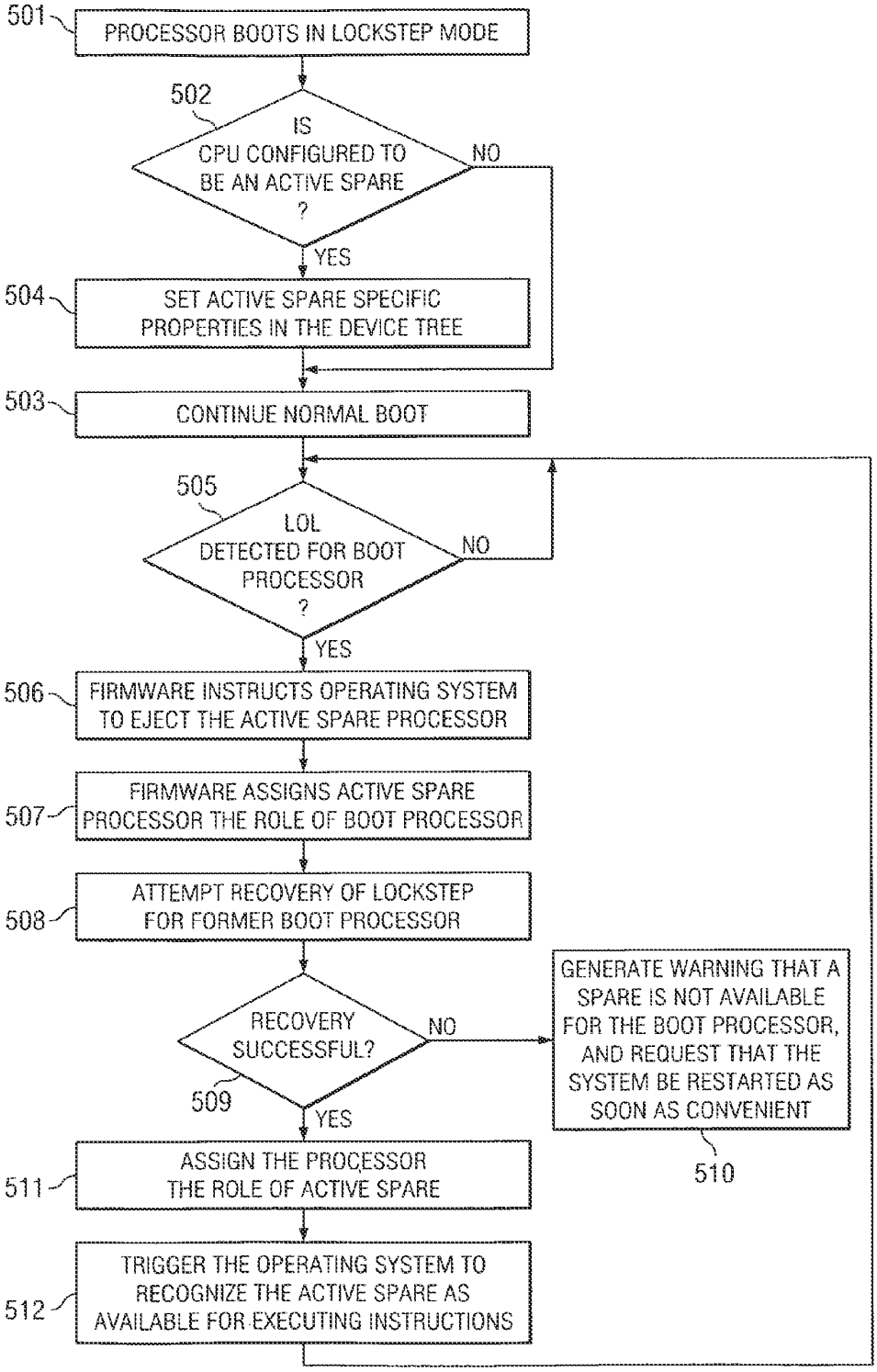


FIG. 5


501
PROCESSOR BOOTS IN LOCKSTEP MODE
502
IS CPU CONFIGURED TO BE AN ACTIVE SPARE ?
NO
YES
504
SET ACTIVE SPARE SPECIFIC PROPERTIES IN THE DEVICE TREE
503
CONTINUE NORMAL BOOT
505
LOL DETECTED FOR BOOT PROCESSOR ?
NO
YES
506
FIRMWARE INSTRUCTS OPERATING SYSTEM TO EJECT THE ACTIVE SPARE PROCESSOR
507
FIRMWARE ASSIGNS ACTIVE SPARE PROCESSOR THE ROLE OF BOOT PROCESSOR
508
ATTEMPT RECOVERY OF LOCKSTEP FOR FORMER BOOT PROCESSOR
RECOVERY SUCCESSFUL?
NO
509
YES
GENERATE WARNING THAT A SPARE IS NOT AVAILABLE FOR THE BOOT PROCESSOR, AND REQUEST THAT THE SYSTEM BE RESTARTED AS SOON AS CONVENIENT
510
511
ASSIGN THE PROCESSOR THE ROLE OF ACTIVE SPARE
512
TRIGGER THE OPERATING SYSTEM TO RECOGNIZE THE ACTIVE SPARE AS AVAILABLE FOR EXECUTING INSTRUCTIONS

SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application is related to the following concurrently filed and commonly assigned U.S. Patent Applications: U.S. patent application Ser. No. 10/972,835 (U.S. Published Patent Application 20060107114) titled "SYSTEM AND METHOD FOR USING INFORMATION RELATING TO A DETECTED LOSS OF LOCKSTEP FOR DETERMINING A RESPONSIVE ACTION"; U.S. patent application Ser. No. 10/973,004 (U.S. Published Patent Application 20060107117, now U.S. Pat. No. 7,308,566) titled "SYSTEM AND METHOD FOR CONFIGURING LOCKSTEP MODE OF A PROCESSOR MODULE"; U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION"; U.S. patent application Ser. No. 10/972,588 (U.S. Published Patent Application 20060090064) titled "SYSTEM AND METHOD FOR SWITCHING THE ROLE OF BOOT PROCESSOR TO A SPARE PROCESSOR RESPONSIVE TO DETECTTON OF LOSS OF LOCKSTEP IN A BOOT PROCESSOR"; U.S. patent application Ser. No. 10/973,003 (U.S. Published Patent Application 20060107116) titled "SYSTEM AND METHOD FOR REESTABLISHING LOCKSTEP FOR A PROCESSOR MODULE FOR WHICH LOSS OF LOCKSTEP IS DETECTED"; U.S. patent application Ser. No. 10/972,888 (U.S. Published Patent Application 20060107115) titled "SYSTEM AND METHOD FOR SYSTEM FIRMWARE CAUSING AN OPERATING SYSTEM TO IDLE A PROCESSOR"; U.S. patent application Ser. No. 10/973,075 (U.S. Published Patent Application 20060107111) titled "SYSTEM AND METHOD FOR REINTRODUCING A PROCESSOR MODULE TO AN OPERATING SYSTEM AFTER LOCKSTEP RECOVERY"; and U.S. patent application Ser. No. 10/972,796 (U.S. Published Patent Application 20060107106) titled "SYSTEM AND METHOD FOR MAINTAINING IN A MULTI-PROCESSOR SYSTEM A SPARE PROCESSOR THAT IS IN LOCKSTEP FOR USE IN RECOVERING FROM LOSS OF LOCKSTEP FOR ANOTHER PROCESSOR", the disclosures of which are hereby incorporated herein by reference.

DESCRIPTION OF RELATED ART

Silent Data Corruption ("SDC") is a difficult problem in the computing industry. In general, SDC refers to data that is corrupt, but which the system does not detect as being corrupt. SDCs primarily occur due to one of two factors: a) a broken hardware unit or b) a "cosmic" event that causes values to change somewhere in the system. Broken hardware means that a "trusted" piece of hardware is silently giving wrong answers. For example, the arithmetic unit in a processor is instructed to add 1+1 and it returns the incorrect answer 3 instead of the correct answer 2. An example of a cosmic event is when a charged particle (e.g., alpha particle or cosmic ray) strikes a region of a computing system and causes some bits to change value (e.g., from a 0 to a 1 or from a 1 to a 0).

Numerous techniques have been developed for detecting SDC to prevent the SDC from remaining "silent" or "undetected" within a system, as well as preventing such SDC from propagating through the system. Examples of these techniques include parity-based mechanisms and error correcting codes (ECCs) on buses and memory locations, as well as checksums and/or cyclic redundancy checks (CRC) over regions of memory. Parity-based mechanisms are often employed in processors, wherein a parity bit is associated with each block of data when it is stored. The parity bit is set to one or zero according to whether there is an odd or even number of ones in the data block. When the data block is read out of its storage location, the number of ones in the block is compared with the parity bit. A discrepancy between the values indicates that the data block has been corrupted. ECCs are parity-based mechanisms that track additional information for each data block. The additional information allows the corrupted bit(s) to be identified and corrected.

Parity/ECC mechanisms have been employed extensively for caches, memories, and similar data storage arrays. In the remaining circuitry on a processor, such as data paths, control logic, execution logic, and registers (the "execution core"), it is more difficult to apply parity/ECC mechanisms for SDC detection. Thus, there is typically some unprotected area on a processor in which data corruption may occur and the parity/ECC mechanisms do not prevent the corrupted data from actually making it out onto the system bus. One approach to SDC detection in an execution core (or other unprotected area of the processor chip) is to employ "lockstep processing." Generally, in lockstep processing two processors are paired together, and the two processors perform exactly the same operations and the results are compared (e.g., with an XOR gate). If there is ever a discrepancy between the results of the lockstep processors, an error is signaled. The odds of two processors experiencing the exact same error at the exact same moment (e.g., due to a cosmic event occurring in both processors at exactly the same time or due to a mechanical failure occurring in each processor at exactly the same time) is nearly zero.

A pair of lockstep processors may, from time to time, lose their lockstep. "Loss of lockstep" (or "LOL") is used broadly herein to refer to any error in the pair of lockstep processors. One example of LOL is detection of data corruption (e.g., data cache error) in one of the processors by a parity-based mechanism and/or ECC mechanism. Another example of LOL is detection of the output of the paired processors not matching, which is referred to herein as a "lockstep mismatch." It should be recognized that in some cases the data in the cache of a processor may become corrupt (e.g., due to a cosmic event), which once detected (e.g., by a parity-based mechanism or ECC mechanism of the processor) results in LOL. Of course, unless such corrupt data is acted upon by the processor, the output of that processor will not fail to match the output of its paired processor and thus a "lockstep mismatch" will not occur. For example, suppose that a value of "1" is stored to first location of cache in each of a pair of lockstep processors and a value of "1" is also stored to a second location of cache in each of the pair of lockstep processors. Further suppose that a cosmic event occurs for a first one of the processors, resulting in the first location of its cache being changed from "1" to "0", and thus corrupted. This data corruption in the first processor is a LOL for the pair. An error detection mechanism of this first processor may detect the data corruption, thus detecting the LOL. If the processors are instructed to act on the data of their first cache locations, then a lockstep mismatch will occur as the output of each of the processors will not match. For instance, if the processors each add the data stored to the first location of their respective cache with the data stored to the second location of their respective cache, the first processor (having the corrupt data) will output a

result of "1" (0+1=1) while the second processor outputs a result of "2" (1+1=2), and thus their respective outputs will not match.

By employing such techniques as parity-based error detection mechanisms and output comparisons for lockstep paired processors, SDC detection can be enhanced such that practically no SDC occurring in a processor goes undetected (and thus such SDC does not remain "silent") but instead results in detection of LOL. However, the issue then becomes how best for the system to respond to detected LOL. The traditional response to detected LOL has been to crash the system to ensure that the detected error is not propagated through the system. That is, LOL in one pair of lockstep processors in a system halts processing of the system even if other processors that have not encountered an error are present in the system. However, with the increased desire for many systems to maintain high availability, crashing the system each time LOL is detected is not an attractive proposition. This is particularly unattractive for large systems having many processors because cosmic events typically occur more frequently as the processor count goes up, which would result in much more frequent system crashes in those large systems. High availability is a major desire for many customers having large, multi-processor systems, and thus having their system crash every few weeks is not an attractive option. Of course, permitting corrupt data to propagate through the system is also not a viable option.

Prior solutions attempting to resolve at least some detected SDCs without requiring the system to be crashed have been Operating System ("OS") centric. That is, in certain solutions the OS has been implemented in a manner to recover from a detected LOL without necessarily crashing the system. This OS-centric type of solution requires a lot of processor and platform specific knowledge to be embedded in the OS, and thus requires that the OS provider maintain the OS up-to-date as changes occur in later versions of the processors and platforms in which the OS is to be used. This is such a large burden that most commonly used OSs do not support lockstep recovery.

Certain solutions have attempted to recover from a LOL without involving the OS in such recovery procedure. For instance, in one technique upon LOL being detected, firmware is used to save the state of one of the processors in a lockstep pair (the processor that is considered "good") to memory, and then both processors of the pair are reset and reinitialized. Thereafter, the state is copied from the memory to each of the processors in the lockstep pair. This technique makes the processors unavailable for an amount of time without the OS having any knowledge regarding this unavailability, and if the amount of time required for recovery is too long, the system may crash. That is, typically, if a processor is unresponsive for X amount of time, the OS will assume that the processor is hung and will crashdump the system so that the problem can be diagnosed. Further, in the event that a processor in the pair cannot be reset and reinitialized (e.g., the processor has a physical problem and fails to pass its self-test), this technique results in crashing the system.

BRIEF SUMMARY OF THE INVENTION

According to at least one embodiment, a system comprises a plurality of processors, and data storage storing information that assigns a role of boot processor to one of the plurality of processors and assigns a role of spare processor to another of the plurality of processors. The system further comprises logic operable, responsive to detecting loss of lockstep for the boot processor, for transferring, during system runtime, the role of boot processor to the spare processor.

According to at least one embodiment, a method comprises assigning a role of boot processor to one of a plurality of processors in a system; assigning a role of spare processor to another of the plurality of processors; and responsive to detecting loss of lockstep for the boot processor, changing, during system runtime, the assignment of the role of boot processor to the spare processor.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows an example multi-processor system in which firmware utilizes a spare processor for recovering from LOL for the system's boot processor;

FIG. **2** shows a block diagram of one embodiment implemented for the IA-64 processor architecture;

FIG. **3** shows an example system having multi-processor cells in which an embodiment for using firmware for recovering from LOL may be employed;

FIG. **4** shows an operational flow diagram for establishing a hot spare for the boot processor according to one embodiment;

FIG. **5** shows an operational flow diagram for establishing an active spare for the boot processor and assigning such active spare the role of boot processor responsive to detecting LOL for the boot processor in accordance with one embodiment; and

FIG. **6** shows a general flow diagram for operation of certain embodiments.

DETAILED DESCRIPTION

As described further herein and in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION," the disclosure of which is incorporated herein by reference, certain techniques are provided for recovering from LOL detected for a boot processor in a multi-processor system, in which a spare processor is utilized for such recovery. For instance, upon LOL being detected for the system's boot processor, the spare processor assumes the role of boot processor, and then recovery of lockstep for the original boot processor is attempted. Accordingly, for such lockstep recovery technique, a spare processor is used for attempting to recover from LOL detected for the boot processor. Embodiments are provided herein for establishing a spare processor for the boot processor. That is, embodiments described herein provide techniques for establishing, in a multi-processor system, a spare processor that assumes the role of boot processor upon LOL being detected for the system's boot processor.

In certain embodiments, information (referred to herein as a "device tree") is stored to the multi-processor system, wherein such information identifies the processor in the system that is the boot processor and identifies the processor that is the spare for the boot processor. The information is stored such that it is accessible by the system's firmware, and the firmware, 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. The firmware then attempts to recover lockstep for the processor for which LOL was detected. If 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. Thus, 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).

In some implementations described herein, the spare processor for the boot processor is a "hot" spare, in which it is idled and not made available to the OS for processing instructions during normal system operation. That is, the hot spare processor is held in reserve, idling until such time as it assumes the role of boot processor responsive to detected LOL for the boot processor. In other implementations, the spare processor is not held in reserve, but is instead an "active" processor that is made available to the OS for executing instructions during normal system operation. Upon LOL being detected for the boot processor, the designated spare processor is dynamically transformed into a hot spare (i.e., it is idled) such that it can assume the role of the boot processor. For instance, upon the system's firmware detecting a LOL for the boot processor, the firmware accesses the device tree and determines the designated spare processor, and then the firmware instructs the OS to idle the designated spare processor (i.e., stop scheduling instructions for the processor) to make the "active" spare a "hot" spare, which then assumes the role of boot processor.

Concurrently filed and commonly assigned U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION," discloses a technique in which lockstep can be recovered for processors (particularly, non-boot processors) without requiring use of a spare processor. Rather, upon LOL being detected for a lockstep pair of processors, the firmware instructs the OS to idle (or "eject") the lockstep pair of processors for which LOL was detected and then the firmware attempts to reestablish lockstep (e.g., by resetting the processors). If successful in reestablishing lockstep, the firmware triggers the OS to recognire the lockstep pair of processors as again being available for servicing instructions.

However, for various reasons, in certain system architectures problems arise in attempting to idle (or eject) the boot processor from the system. Thus, in certain embodiments, a spare processor is used for recovering from LOL for the system's boot processor. Again, embodiments are provided herein for establishing/defining such a spare processor for the boot processor. In certain embodiments, the spare processor is used for recovering from LOL only in the case in which LOL is detected for the boot processor. For instances in which LOL is detected for non-boot processors, the lockstep may be recovered without the use of a spare by using the recovery technique described in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION," for example.

Various techniques are provided herein for designating the spare for a boot processor. In certain embodiments, a user interface is provided for receiving input specifying the boot processor and/or its spare. In certain embodiments, the system's firmware selects the boot processor and/or its spare. In accordance with one embodiment provided herein, the multiprocessor system includes multiple buses, wherein certain processors are communicatively coupled to a first bus and certain other processors are communicatively coupled to a different bus. In such an architecture, the spare processor is designated as a processor that is coupled to the same bus as the boot processor. Thus, if a user is allowed to select the boot processor and the spare, once the boot processor is selected, the user interface may present the other processor(s) that are coupled to the same bus as the selected boot processor for selection of a spare therefrom. Alternatively, once the boot processor is selected by a user, the system firmware may select a spare as a processor that is coupled to the same bus as the boot processor. In certain embodiments, the system firmware may select the boot processor, as well as a spare for the boot processor.

Turning to FIG. **1**, an example embodiment of a system **10** that establishes a spare processor for use in recovering from detected loss of lockstep (LOL) for a system boot processor is shown. System **10** includes OS **11**, as well as master processor **12**A and slave processor **12**B (collectively referred to as a lockstep processor pair **12**). In certain implementations the lockstep processor pair **12** may be implemented on a single silicon chip, which is referred to as a "dual core processor" in which master processor **12**A is a first core and slave processor **12**B is a second core. Further, lockstep processor pair **12** may be referred to as a processor or CPU "module" because it includes a plurality of processors (**12**A and **12**B) in such module. As described further herein, to certain portions of system **10**, such as its OS **11**, the processor module **12** may appear as a single processor because slave processor **12**B is merely used for redundant processing for checking the output of master processor **12**A. In this example, lockstep processor pair **12** is designated as the boot processor for system **10**. As described further herein, in certain embodiments lockstep processor pair **12** is designated as the system boot processor by storing information to the system that specifies such designation.

Master processor **12**A includes cache **14**A, and slave processor **12**B includes cache **14**B. OS **11** and lockstep processor pair **12** are communicatively coupled to bus **16**. Typically, master processor **12**A and slave processor **12**B are coupled to bus **16** via an interface that allows each of such processors to receive the same instructions to process, but such interface only communicates the output of master processor **12**A back onto bus **16**. The output of slave processor **12**B is used solely for checking the output of mater processor **12**A.

In this example, master processor **12**A includes error detect logic **13**A, and slave processor **12**B includes error detect logic **13**B. While shown as included in each of the processors **12**A and **12**B in this example, in certain embodiments the error detect logic **13**A and **13**B may be implemented external to processors **12**A and **12**B. Error detect logic **13**A and **13**B include logic for detecting errors, such as data cache errors, present in their respective processors **12**A and **12**B. Examples of error detect logic **13**A and **13**B include known parity-based mechanisms and ECC mechanisms. Error detect logic **13**C is also included, which may include an XOR (exclusive OR) gate, for detecting a lockstep mismatch between master processor **12**A and slave processor **12**B. As mentioned above, a lockstep mismatch refers to the output of master processor **12**A and slave processor **12**B failing to match. While shown as external to the lockstep processor pair **12** in this example, in certain embodiments error detect logic **13**C may be implemented on a common silicon chip with processors **12**A and **12**B.

Lockstep mismatch is one way of detecting a LOL between the master processor **12**A and slave processor **12**B. A detection of an error by either of error detect logic **13**A and **13**B also provides detection of LOL in the processors **12**A and **12**B. Because the detection of LOL by error detect logic **13**A and **13**B may occur before an actual lockstep mismatch occurs, the detection of LOL by error detect logic **13**A and **13**B may be referred to as a detection of a "precursor to

lockstep mismatch". In other words, once an error (e.g., corrupt data) is detected by error detect logic **13**A or **13**B, such error may eventually propagate to a lockstep mismatch error that is detectable by error detect logic **13**C.

System **10** further includes a second lockstep processor pair **15** that is communicatively coupled to bus **16**. Lockstep processor pair **15** includes master processor **15**A and slave processor **15**B. Master processor **15**A includes cache **17**A, and slave processor **15**B includes cache **17**B. As with error detect logic **13**A-**13**C implemented for lockstep processor pair **12**, lockstep processor pair **15** has error detect logic **16**A-**16**C. While only two lockstep processor pairs **12** and **15** are shown for simplicity in the example of FIG. **1**, system **10** may include any number of such lockstep processor pairs. As one specific example, in one implementation system **10** includes 64 lockstep processor pairs, wherein the master processors of the pairs perform parallel processing for the system. In the specific example of FIG. **1**, lockstep processor pair **15** is designated as a spare for boot processor **12**, as described further herein.

In certain embodiments described herein, techniques for designating a spare processor for a boot processor are provided, wherein the spare processor is used for recovery when LOL is detected for the system's boot processor. That is, the designated spare processor assumes the role of boot processor in the event that LOL is detected for the boot processor. In accordance with certain embodiments, upon LOL being detected for a boot processor, a hot spare processor (i.e., an idling processor that is available in the system) is transferred the role of boot processor, and then the old boot processor having LOL is reset to reestablish its lockstep. Thus, in certain embodiments, a "hot" spare is designated for the boot processor.

In other embodiments, 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 processing resource made available to OS **11** for processing instructions that OS **11** schedules for such "active" spare processor. Upon LOL being detected for the boot processor, the designated spare processor is idled (thereby, transforming such active processor into a hot spare), and is transferred the role of boot processor. Then, the former boot processor for which the LOL was detected is reset to reestablish its lockstep. Thus, in certain embodiments, a designated spare processor is not required to be held in reserve, but is instead an available resource for use within the system and is only transformed into an actual spare processor when needed for recovering from LOL for the boot processor.

Firmware **18** is also included in system **10**, which in this embodiment is invoked upon an error being detected by any of the error detect logics **13**A-**13**C and **16**A-**16**C. In certain embodiments, processors **12**A, **12**B, **15**A, and **15**B are processors from the Itanium Processor Family (IPF). IPF is a 64-bit processor architecture co-developed by Hewlett-Packard Company and Intel Corporation, which is based on Explicitly Parallel Instruction Computing (EPIC). IPF is a well-known family of processors. IPF includes processors such as those having the code names of MERCED, MCKINLEY, and MADISON. In addition to supporting a 64-bit processor bus and a set of 128 registers, the 64-bit design of IPF allows access to a very large memory (VLM) and exploits features in EPIC. While a specific example implementation of one embodiment is described below for the IPF architecture, embodiments described herein for establishing a spare processor for recovering from LOL detected for a boot processor are not limited in application to an IPF architecture, but may be applied as well to other architectures (e.g., 32-bit processor architectures, etc.).

Processor architecture generally comprises corresponding supporting firmware, such as firmware **18** of system **10**. For example, as described further below in conjunction with the specific example of FIG. **2**, the IPF processor architecture comprises such supporting firmware as Processor Abstraction Layer (PAL), System Abstraction Layer (SAL), and Extended Firmware Interface (EFI). Such supporting firmware may enable, for example, the OS to access a particular function implemented for the processor. For instance, the OS may query the PAL as to the size of the cache implemented for the processor, etc. Other well-known functions provided by the supporting firmware (SAL, EFI) include, for example: (a) performing I/O configuration accesses to discover and program the I/O Hardware (SAL_PCI_CONFIG_READ and SAL_PCI_CONFIG-WRITE); (b) retrieving error log data from the platform following a Machine Check Abort (MCA) event (SAL_GET_STATE_INFO); (c) accessing persistent store configuration data stored in non-volatile memory (EFI variable services: GetNextVariableName, GetVariable and SetVariable); and accessing the battery-backed real-time clock/calendar (EFI GetTime and SetTime). Accordingly, the supporting firmware, such as the PAL, is implemented to provide an interface to the processor(s) for accessing the functionality provided by such processor(s). Each of those interfaces provide standard, published procedure calls that are supported.

In the example system **10** of FIG. **1**, firmware **18** utilizes the above-mentioned hot spare technique for recovering from LOL for a boot processor. That is, processor **15** is designated as a hot spare for boot processor **12**, and such processor **15** is held as an idle, spare processor and used in the manner described further herein for assuming the role of boot processor in the event that a LOL is detected for boot processor **12**. Of course, as described further herein, the above-mentioned technique in which a designated spare is not held in reserve but is instead dynamically transformed into a spare processor may be implemented by firmware **18** in alternative embodiments. As described further herein, in certain embodiments, a device tree is stored to a data storage device (e.g., SRAM) such that it is accessible to the firmware **18**, wherein the device tree includes information identifying processor **12** as the system's boot processor and identifying processor **15** as the spare. The information specifying the boot processor and/or spare may be received as input from a user in certain embodiments. For instance, in certain embodiments a user interface is provided that enables a system administrator to designate the boot processor and/or spare processor in the multi-processor system. In other embodiments, the firmware **18** selects the boot processor and/or spare.

Suppose that during the system boot-up procedure, master processor **12**A assumes the role of boot processor, as shown in the example of FIG. **1**. In this example embodiment, lockstep processor pair **15** is held as a hot spare for recovering from a LOL that may be detected for the boot processor **12**A. Of course, additional lockstep processor pairs may be included in system **10** (not specifically shown in the example of FIG. **1**), and those additional lockstep processor pairs, which are not the system boot processor, may recover from LOL in a different manner that does not utilize a hot spare processor. Thus, in accordance with certain embodiments hereof, hot spare processor pair **15** is not needed for recovering from LOL detected for any non-boot processor, but may

instead be used for recovery of LOL only if such LOL is detected for the boot processor (processor **12**A in this example).

Firmware **18** is included in this example, and upon detection of LOL by any of error detect logics **13**A-**13**C, firmware **18** determines, in operational block **101**, whether the detected LOL is a recoverable LOL. That is, firmware **18** determines in block **101** whether the detected LOL is of a type from which the firmware can recover lockstep for the lockstep processor pair **12** without crashing the system. As described further herein, lockstep is recoverable for certain detected LOLs (which may be referred to as "recoverable LOLs"), while lockstep is not recoverable for other detected LOLs (which may be referred to as "non-recoverable LOLs"). If the lockstep is not recoverable from the detected LOL, then in the example of FIG. **1** firmware **18** crashes the system in block **102**.

If the lockstep is recoverable, then operation advances to block **103** whereat the firmware determines whether the processor for which the LOL was detected is the system's boot processor. Because the LOL is detected for the lockstep processor pair **12** in this example, firmware **18** determines whether master processor **12**A is the system's boot processor. This can be determined, for example, by accessing the device tree **203** of FIG. **2** (described further below). The device tree may, in certain implementations, include a field for each processor indicating whether such processor is the system's boot processor. This field in the device tree may be set by the firmware during the boot-up process to identify the corresponding processor that is used as the system's boot processor or it may be pre-set by a system administrator (e.g., via a user interface), as examples. If determined in block **103** that the master processor **12**A is not the system's boot processor, then operation advances in this example to block **104**, and a lockstep recovery process is performed that does not utilize hot spare **15**.

In the specific example shown in FIG. **1**, if the lockstep is determined to be recoverable in block **101** and the processor for which the LOL is detected is determined in block **103** to not be the system boot processor, firmware **18** cooperates with OS **11** via standard OS methods to recover the lockstep. For instance, in the example embodiment of FIG. **1**, Advanced Configuration and Power Interface (ACPI) methods are used by firmware **18** to cooperate with OS **11**. Accordingly, no processor or platform specific knowledge is required to be embedded in OS **11**, but instead any ACPI-compatible OS may be used, including without limitation HP-UX and Open VMS operating systems. Example techniques that may be used for recovering from LOL for non-boot processors of a system are described further in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/973,076 (U.S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION," the disclosure of which is incorporated herein by reference.

In the example embodiment of FIG. **1**, if determined in block **103** that the processor for which the LOL is detected is not the system boot processor, firmware **18** triggers OS **11** to idle the master processor **12**A in operational block **104**. In this embodiment, firmware **18** utilizes an ACPI method **105** to "eject" master processor **12**A, thereby triggering OS **11** to idle the master processor **12**A (i.e., stop scheduling tasks for the processor). Of course, by idling master processor **12**A, slave processor **12**B will in turn be idled. Thus, idling master processor **12**A results in idling the lockstep processor pair **12**. In this example embodiment, OS **11** is not aware of the presence of slave processor **12**B, but is instead aware of master processor **12**A. The interface of lockstep processor pair **12** to bus **16** manages copying to slave processor **12**B the instructions that are directed by OS **11** to master processor **12**A. Thus, firmware **18** need not direct OS **11** to eject slave processor **12**B, as OS **11** is not aware of such slave processor **12**B in this example implementation. Again, by idling master processor **12**A, slave processor **12**B is also idled as it merely receives copies of the instructions directed to master processor **12**A. Of course, if in a given implementation OS **11** is aware of slave **12**B, firmware **18** may be implemented to also direct OS **11** to idle such slave processor **12**B in a manner similar to that described for idling master processor **12**A.

Firmware **18** then attempts to recover lockstep for the lockstep processor pair **12** in block **106**. For instance, firmware **18** resets the processor pair **12**. During such reset of processor pair **12**, system **10** can continue to operate on its remaining available processors. Once the processor pair **12** is reset and lockstep is recovered, firmware **18** reintroduces master processor **12**A to OS **11** in operational block **107**. In this embodiment, firmware **18** updates the ACPI device table information for master processor **12**A to indicate that such master processor **12**A is "present, functioning and enabled." As discussed in the ACPI 2.0 specification for the _STA status method of a device, the _STA (status) object returns the status of a device, which can be one of the following: enabled, disabled, or removed. In this respect, in the result code (bitmap) bit **0** is set if the device is present; bit **1** is set if the device is enabled and decoding its resources; bit **2** is set if the device should be shown in the UI; bit **3** is set if the device is functioning properly (cleared if the device failed its diagnostics); bit **4** is set if the battery is present; and bits **5**-**31** are reserved. A device can only decode its hardware resources if both bits **0** and **1** are set. If the device is not present (bit **0** cleared) or not enabled (bit **1** cleared), then the device must not decode its resources. Bits **0**, **1** and **3** are the "present, enabled and functioning" bits mentioned above. Firmware **18** utilizes an ACPI method **108** to trigger OS **11** to "check for" master processor **12**A, thereby reintroducing the master processor **12**A to OS **11**. As a result of checking for master processor **12**A, OS **11** will recognize that such master processor **12**A is again available and will thus begin scheduling tasks for master processor **12**A once again.

In view of the above, a recovery technique is provided that does not require utilizing a spare processor for recovering from LOL detected for non-boot processors. For various reasons, in certain system architectures problems arise in attempting to idle (or eject) the boot processor from the system. Thus, in certain embodiments, a hot spare processor is used for recovering from LOL for the system's boot processor. For instance, in the example of FIG. **1**, if determined in block **103** that the master processor **12**A is the system's boot processor, operation advances to block **109** whereat the state of the "good" processor in the lockstep processor pair **12** is copied over to each of processors **15**A and **15**B in the spare lockstep processor pair **15**. Example techniques that may be utilized by firmware **18** for switching the state of a "good" processor in the lockstep processor pair 12 in accordance with certain embodiments is described further in co-pending U.S. patent application Ser. No. 10/187,833 (published as publication no. US 2004/0006722 A1) filed Jul. 3, 2002 (now U.S. Pat. No. 7,085,959) and titled "METHOD AND APPARATUS FOR RECOVERY FROM LOSS OF LOCK STEP" and in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/972,588 (U.S. Published Patent Application 20060090064) 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", the disclosures of which are each hereby incorporated herein by reference.

Thus, operational block **109** essentially makes the spare processor pair **15** the system's boot processor, and then in block **110** firmware **18** resets the lockstep processor pair **12** and reestablishes its lockstep. Once lockstep is reestablished for lockstep processor pair **12**, that pair is held in operational block **111** as a hot spare for the new boot processor pair **15**. That is, firmware **18** updates the device tree **203** of FIG. **2** to reflect that lockstep processor pair **12** is a hot spare for the boot processor pair **15**. Thus, should a recoverable LOL be detected for the new boot processor **15**, the above process may be used by firmware **18** to make the spare lockstep processor pair **12** the boot processor and then recover lockstep for pair **15**. An example technique for resetting the lockstep processor pair **12** and reestablishing its lockstep in which the lockstep processor pair **12** is then held as a hot spare is described further in concurrently filed and commonly assigned U.S. patent application Ser. No. 10/972,796 U.S. Published Patent Application 20060107106) titled "SYSTEM AND METHOD FOR MAINTAINING IN A MULTI-PROCESSOR SYSTEM A SPARE PROCESSOR THAT IS IN LOCKSTEP FOR USE IN RECOVERING FROM LOSS OF LOCKSTEP FOR ANOTHER PROCESSOR", the disclosure of which is hereby incorporated herein by reference.

Embodiments described herein enable a spare processor to be designated for a boot processor, such as spare processor **15** in the above example. In certain implementations, a system administrator can designate through a user interface the boot processor, and, in some cases, further designate the corresponding spare for such boot processor. In other implementations, a system administrator may designate the boot processor and firmware selects the appropriate spare processor. In other embodiments, the firmware may select a boot processor and the corresponding spare processor for such boot processor. The information identifying the boot processor and the spare processor to be used for recovery from LOL detected for the boot processor is stored to the system in a data storage device (e.g., SRAM) that is accessible by the system's firmware to enable the firmware to identify those processors.

FIG. **2** shows a block diagram of one embodiment of the above system **10**, which is implemented for the IPF processor architecture and is labeled as system $\mathbf{10}_A$. The quintessential model of the traditional IPF architecture is given in the *Intel IA-64 Architecture Software Developer's Manual, Volume 2: IA-64 System Architecture*, in section 11.1 *Firmware Model*, the disclosure of which is hereby incorporated herein by reference. Accordingly, in this example embodiment of system $\mathbf{10}_A$, firmware **18**, labeled as firmware $\mathbf{18}_A$, includes processor abstraction layer (PAL) **201** and platform/system abstraction layer (SAL) **202**. In general, PAL **201** is firmware provided by Intel for its processors, and SAL **202** is developed by an original equipment manufacturer (OEM) for the specific system/platform in which the processors are to be employed. PAL **201**, SAL **202**, as well as an extended firmware interface (EFI) layer (not shown), together provide, among other things, the processor and system initialization for an OS boot in an IPF system.

The boot-up process of a traditional IPF system, for example, proceeds as follows: When the system is first powered on, there are some sanity checks (e.g., power on self-test) that are performed by microprocessors included in the system platform, which are not the main system processors that run applications. After those checks have passed, power and clocks are given to a boot processor (which may, for example, be master processor **12**A). The boot processor begins executing code out of the system's Read-Only Memory (ROM) (not specifically shown in FIG. **2**). The code that executes is the PAL **201**, which gets control of system **10**. PAL **201** executes to acquire all of the processors in system $\mathbf{10}_A$ (recall that there may be many lockstep processor pairs, such as pairs **12** and **15**) such that the processors begin executing concurrently through the same firmware.

After it has performed its duty of initializing the processor(s), PAL **201** passes control of system $\mathbf{10}_A$ to SAL **202**. It is the responsibility of SAL **202** to discover what hardware is present on the system platform, and initialize it to make it available for the OS **11**. When main memory is initialized and functional, the firmware $\mathbf{18}_A$ is copied into the main memory. Then, control is passed to EFI (not shown), which is responsible for activating boot devices, which typically includes the disk. The EFI reads the disk to load a program into memory, typically referred to as an operating system loader. The EFI loads the OS loader into memory, and then passes it control of system $\mathbf{10}_A$ by branching the boot processor into the entry point of such OS loader program.

The OS loader program then uses the standard firmware interfaces to discover and initialize system $\mathbf{10}_A$ further for control. One of the things that the OS loader typically has to do in a multi-processor system is to retrieve control of the other processors (those processors other than the boot processor). For instance, at this point in a multi-processor system, the other processors may be executing in do-nothing loops. In an ACPI-compatible system, OS **11** makes ACPI calls to parse the ACPI tables to discover the other processors of a multi-processor system in a manner as is well-known in the art. Then OS **11** uses the firmware interfaces to cause those discovered processors to branch into the operating system code. At that point, OS **11** controls all of the processors and the firmware $\mathbf{18}_A$ is no longer in control of system $\mathbf{10}_A$.

As OS **11** is initializing, it has to discover from the firmware $\mathbf{18}_A$ what hardware is present at boot time. And in the ACPI standards, it also discovers what hardware is present or added or removed at run-time. Further, the supporting firmware (PAL, SAL, and EFI) are also used during system runtime to support the processor. For example, OS **11** may access a particular function of master processor **12**A via the supporting firmware $\mathbf{18}_A$, such as querying PAL **201** for the number, size, etc., of the processor's cache **14**A. Some other well-known firmware functions that OS **11** may employ during runtime include: (a) PAL **201** may be invoked to configure or change processor features such as disabling transaction queuing (PAL_BUS_SET_FEATURES); (b) PAL **201** may be invoked to flush processor caches (PAL_CACHE_FLUSH); (c) SAL **202** may be invoked to retrieve error logs following a system error (SAL_GET_STATE_INFO, SAL_CLEAR_STATE_INFO); (d) SAL **202** may be invoked as part of hot-plug sequences in which new I/O cards are installed into the hardware (SAL_PCI_CONFIG_READ, SAL_PCI_CONFIG_WRIT); (e) EFI may be invoked to change the boot device path for the next time the system reboots (SetVariable); (f) EFI may be invoked to change the clock/calendar hardware settings; and (g) EFI may be invoked to shutdown the system (ResetSystem).

A "device tree" is provided, which is shown as device tree **203** in this example. Device tree **203** is stored in SRAM (Scratch RAM) on the cell, which is RAM that is reinitialized. Firmware **18**A builds the device tree **203** as it discovers what hardware is installed in the system. Firmware then converts this information to the ACPI tables format and presents it to OS **11** so that OS **11** can know what is installed in the system. The ACPI device tables (not shown) are only consumed by OS **11** at boot time, so they are never updated as things

change. For OS **11** to find the current status, it calls an ACPI "method" to discover the "current status". The _STA method described above is an example of such an ACPI method. When _STA is called, the AML can look for properties on the device specified in the firmware device tree and convert that into the Result Code bitmap described above. So, if lockstep has been lost on a processor, firmware **15**A will set the device tree property that indicates loss of lockstep, then when OS **11** calls _STA for that device, the "lockstep lost" property directs the AML code to return to "0" in the "functioning properly" bit so that OS **11** can know there is a problem with that processor.

If a lost lockstep is recovered in accordance with the recovery technique described herein, firmware **18**A can indicate that lockstep has been recovered in the device tree **203**. Then when _STA is called on that device responsive to the OS receiving the "check for device" ACPI method, the present, enabled and functioning bits will all be set and OS **11** will know the CPU is safe to use.

Further, device tree **203** includes information designating which processor is the boot processor, as well as a spare processor for the boot processor. A simple example of device tree **203** is shown below in Table 1:

TABLE 1

Device	Status	Lockstep Enabled	Boot Processor?	Spare for Boot Processor?
Processor A	Present, Enabled, and Functioning	Yes	Yes	No
Processor B	N/A	Yes	No	Yes
Processor C	Present, Enabled, and Functioning	Yes	No	No
. . .	. . .	. . .	. . .	. . .

In the above example of Table 1, 3 processors "Processor A," "Processor B," and "Processor C" are identified in the device tree as being included in the multi-processor system. The status of each of processors Processor A and Processor B is identified as "Present, Enabled, and Functioning." In this example Processor B is a hot spare, so it is held in reserve and does not show up in the static ACPI tables that are built when the system boots. Accordingly the OS does not know about Processor B and does not have a device handle to call the _STA method on. As described above, the _STA method is a known ACPI method for checking the status of the corresponding device. Therefore, the status of Processor B is designated N/A (not applicable) in this example because the _STA method will not be called for this processor as long as it is the hot spare. Of course, upon such spare assuming the role of boot processor, its status is updated to "Present, Enabled, and Functioning." In other embodiments in which Processor B is an active spare, the OS will be aware of such processor and accordingly its status may be also indicated as "Present, Enabled, and Functioning."

Further, lockstep is identified in this example as enabled for these processors. In certain embodiments, the system administrator can designate, via user interface **20**, whether lockstep mode is to be enabled for the system's processors. Further, Processor A is designated as the system's boot processor in the example device tree shown in Table 1, and Processor B is designated as the spare processor. Thus, upon detection of LOL for one of the processors in the system, the firmware can access this device tree to determine if the processor for which LOL is detected is the system's boot processor; and if determined that the processor for which LOL is detected is the system's boot processor, the firmware can further determine from this device tree the designated spare processor to use in the lockstep recovery process.

In the example of FIG. **2**, a user interface **20** is included, which allows a user (e.g., system administrator) to designate the system's boot processor and spare processor. In certain embodiments, the user interface **20** further allows the user to designate other information, such as whether lockstep mode is enabled for the processors. User interface **20** may be implemented as a command line interface or as a graphical user interface (GUI), as examples. As one example, the user interface **20** may be implemented as an EFI application that provides a command line interface for inputting commands. Such command line interface may support commands for assigning the boot processor and/or spare processor, as well as performing such tasks as configuring and deconfiguring processors, deconfiguring bad memory, specifying which OS to boot, etc. This command line interface implements a typical command <argument> format and has hooks into the firmware to change settings in NVRAM. As another example, the user interface **20** may be implemented within "parmgr", which is an HP-UX application that controls how partitions are configured, memory interleave, CPU configuration, etc. Such parmgr provides a GUI interface and a command line interface. The GUI interface presents a "worksheet" that has the current values filled in and allows the system administrator to update/change those values and then "apply" the changes. At that time, parmgr pushes the changes into NVRAM so they will be honored on the next reset. Various other implementations of a user interface **20** may be employed in alternative embodiments.

One example system in which firmware $\mathbf{18}_A$ of FIG. **2** may be employed is shown in FIG. **3**. FIG. **3** shows system **30** that includes multi-processor cells A-D, labeled $\mathbf{31}_A$-$\mathbf{31}_D$, respectively. Cell A $\mathbf{31}_A$ is shown in more detail, and should be understood that cells B-D $\mathbf{31}_B$-$\mathbf{31}_D$ have substantially the same architecture as that of cell A $\mathbf{31}_A$ in this example. The architecture of cell A $\mathbf{31}_A$ described hereafter corresponds to that of Hewlett-Packard's Superdome™ systems. As shown, multi-processor cell A $\mathbf{31}_A$ includes coherency controller **304** that is communicatively coupled to two buses, labeled **305** and **306**, respectively. A plurality of processor modules are included within cell A $\mathbf{31}_A$, shown as CPU **0**, CPU **1**, CPU **2** and CPU **3** (and labeled **300**-**303**, respectively). More particularly, a plurality of processors modules are communicatively coupled to each of the buses **305** and **306**. In this example, CPU **0** (**300**) and CPU **1** (**301**) are each communicatively coupled to the first bus **305**, and CPU **2** (**302**) and CPU **3** (**303**) are each communicatively coupled to the second bus **306**. It should be understood that each of CPU **0** (**300**), CPU **1** (**301**), CPU **2** (**302**), and CPU **3** (**303**) is in actuality a lockstep processor pair, such as the lockstep processor pair **12** of FIGS. **1**-**3**. Thus, for instance, CPU **0** (**300**) includes both a master processor and a slave processor, as with processors **12**A and **12**B included in the lockstep processor pair **12** of FIGS. **1**-**2**.

Coherency controller **304** decodes the address an interrupt is targeted towards and determines which bus (**305** or **306**) the interrupt packet should be delivered to. Each CPU on the destination bus sees the interrupt and compares the interrupt target address with its internal "LID" register to determine if it is the targeted CPU. If the match is correct, the CPU responds to the interrupt. If the match is not correct, the CPU ignores the interrupt packet.

As shown, the example firmware $\mathbf{18}_A$ described above with FIGS. **1**-**2** may be employed to manage the recovery from detected LOLs for the processors of cells A-D ($\mathbf{31}_A$-$\mathbf{31}_D$). In

this example architecture, when designating a spare for the boot processor, such spare is selected to be another processor communicatively coupled to the same bus as the boot processor. For instance, suppose that during the boot-up of system **30**, CPU **2** (**302**) is designated as the boot processor. Accordingly, firmware 18_A will identify another processor that is communicatively coupled to bus **306** to designate as a spare for such boot processor, such as CPU **3** (**303**). During runtime if a LOL is detected for CPU **2** (**302**), then firmware 18_A may operate according to the process described above in FIG. **1** to switch the designation of boot processor to a designated spare, CPU **3** (**303**). In performing this switch the LID register of the CPU that lost lockstep is copied into the LID register of the spare CPU and the spare CPU pair will then assume the ID of the failed CPU and start responding to interrupts directed to that ID. The LID of the failed CPU is quickly re-written to a different value so that it will no longer respond to interrupts directed to that ID. CPU **2** (**302**) can then be reset to recover its lockstep, and such CPU **2** (**302**) may then be designated as a spare for the new boot processor CPU **3** (**303**).

It should be recognized that in the example system of FIG. **3**, if a hot spare were used for recovering from LOL for all processors, rather than just for the boot processor, much of the potential processing resources would be required to be held in reserve as spare resources. For instance, a hot spare would be required on each of the buses of each cell. Because there are two processors coupled to each bus in this example (i.e., CPU **0** and CPU **1** coupled to bus **305**, and CPU **2** and CPU **3** coupled to bus **306**), reserving a hot spare processor for each bus would result in holding half of the total processing resources in reserve. This is an undesirably expensive and wasteful solution. Thus, according to the illustrated embodiment, a hot spare processor is reserved only for the boot processor. Thus, continuing with the above example in which CPU **2** is the boot processor and CPU **3** is held as a "hot" spare, both of CPU **0** and CPU **1** are active (i.e., neither are held in spare). Further, none of the processors of cells B-D need to be held in reserve as hot spares. In the event of a recoverable LOL occurring for any of non-boot processors of the cells, the lockstep recovery process of FIG. **1** (i.e., operational blocks **104**-**108**) can be utilized without requiring a spare for lockstep recovery for those non-boot processors.

While in the above example embodiment, a hot spare CPU is maintained for recovery from LOL for the system's boot processor, in an alternative embodiment a hot spare need not be held as such, but rather a processor can be dynamically made a "spare" when needed for recovery from LOL encountered for the system boot processor. That is, assuming as in the above example of FIG. **3** that CPU **2** is the system's boot processor, CPU **3** does not have to be held in reserve (idled) as a hot spare for CPU **2**, but instead CPU **3** can be active and dynamically idled and made a "spare" upon detection of LOL in CPU **2**. Accordingly, CPU **3** may be designated as a spare for the boot processor, but such CPU **3** need not be held in reserve during normal system operation. Instead, the CPU **3** may be dynamically (during system run-time) transformed into a spare processor (e.g., by causing the OS to idle such processor) in the event that a LOL is detected for the boot processor.

Thus, the resource of this spare CPU need not be wasted during normal runtime, but can be dynamically turned into a spare processor that is available for recovering from LOL encountered on CPU **2** in the manner described above. As an example of this embodiment, 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. The system boot processor role would be assumed by the idled healthy processor pair, just as described above in the example in which the healthy CPU **3** is held idle for the system boot processor, CPU **2**. The only difference being that now the firmware had to take some initial action to turn the healthy processor pair into a "hot spare" by idling it and making it appear to the OS as though it was not functioning. Thus, the LIDs would be swapped. That is, the LID register of the boot processor that lost lockstep is copied into the LID register of the now spare processor pair (CPU **3** in this example), and the LID register of this spare processor pair would be copied into the LID register of the original boot processor. Lockstep is then reestablished on the original boot processor, and it can be reintroduced to the OS with the ID of the original non-boot processor that replaced it as the system boot processor. The processor that was formerly the boot processor may be designated, in device tree **203**, as the spare for the now boot processor.

Turning to FIG. **4**, an operational flow diagram for establishing a hot spare for the boot processor according to one embodiment is shown. The operational loop of FIG. **4** is performed for each processor in a multi-processor system. In block **401**, the processor boots in lockstep mode. As described above, in certain embodiments, whether the processors are to operate in lockstep mode is configurable in the device tree, and thus the firmware may consult the device tree to determine whether to boot each processor in lockstep mode. In operational block **402**, the firmware determines whether the processor is configured to be a hot spare for the system's boot processor. As described above, this determination can be made by accessing the device tree maintained in the firmware. If the processor is not configured to be a hot spare, then operation advances to block **403** whereat the normal boot process continues for this processor. If, on the other hand, the processor is configured to be a hot spare, then operation advances to block **404** whereat hot spare specific properties are set in the device tree. If configured to be a hot spare, the firmware will hide the processor from the system's OS by setting properties on the node associated with this CPU in the firmware device tree. Typically, nodes of type "CPU" are reported to the OS in the ACPI tables. In this case, the "type" will be set in the device tree to "hot_spare" for the designated spare processor so that this processor is not reported in the ACPI namespace. Thus, when the system firmware builds its ACPI tables, it looks for devices of type "CPU", and the hot spare is not placed in the ACPI tables because its type is set to "hot_spare" in this example. Accordingly, the OS will not be aware of the hot spare processor.

In operational block **405**, the hot spare processor executes in an idle loop waiting for a PAL call to set state. Such PAL call to "set state" copies the saved state from the processor that experienced LOL (i.e., the state of the "good" one of the lockstep pair of processors in the processor module that experienced LOL) into the hot spare processor. Specifically, PAL copies all the needed information from memory into the internal registers of the hot spare processor so that the hot spare processor becomes a new active processor, transforming the role of the hot spare to boot processor. This call does not change the device tree settings. Accordingly, with the spare processor hidden from the OS (i.e., the OS is unaware of the hot spare), the hot spare sits in an idle loop until it is needed for recovering from LOL detected for the boot processor, at which time the system firmware interrupts its loop and assigns it the role of boot processor, as described above.

FIG. **5** shows an operational flow diagram for establishing an active spare for the boot processor and assigning such active spare the role of boot processor responsive to detecting LOL for the boot processor in accordance with one embodiment. Operational blocks **501-504** of FIG. **5** are performed for each processor in a multi-processor system for establishing an active spare in the system. In block **501**, the processor boots in lockstep mode. As described above, in certain embodiments, whether the processors are to operate in lockstep mode is configurable in the device tree, and thus the firmware may consult the device tree to determine whether to boot each processor in lockstep mode. In operational block **502**, the firmware determines whether the processor is configured to be an active spare for the system's boot processor. As described above, this determination can be made by accessing the device tree maintained in the firmware. If the processor is not configured to be an active spare, then operation advances to block **503** whereat the normal boot process continues for this processor. If, on the other hand, the processor is configured to be an active spare, then operation advances to block **504** whereat active spare specific properties are set in the device tree. For instance, in this example, the device tree would include a property called "Active Spare" and this property would be set to TRUE for the active spare processor. If LOL is detected on the boot processor, firmware searches the devices in the device tree for the "Active Spare=TRUE" condition to find the designated active spare processor. Thereafter, operation returns to operational block **503**, whereat the processor continues its normal boot and is introduced to the OS, rather than being hidden from the OS as the above-described hot spare.

Operational block **505** represents the system's normal operation until such time as a LOL is detected for the boot processor. Upon error detect logic detecting a LOL for a processor, the error detect logic invokes the system's firmware. For instance, the LOL may invoke the PAL, which in turn may report the LOL to the SAL. The firmware (e.g., SAL) determines if the processor for which the LOL is detected is designated (e.g., in the device tree) as the boot processor. If so, then the firmware determines in block **505** that LOL is detected for the boot processor. Thus, in operational block **506**, the firmware determines (e.g., from the device tree) 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. In operational block **507**, the firmware assigns the active spare processor the role of boot processor. The state of the "good" one of the lockstep processor pair of the boot processor for which the LOL is detected is copied into both of the lockstep processors of the active spare processor. Further, 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.

In block **508**, the firmware attempts to recover lockstep for the former boot processor for which LOL was detected in block **505**. In operational block **509**, the firmware determines whether the lockstep recovery for the former boot processor is successful. In certain embodiments, if the lockstep cannot be recovered in operational block **508** (e.g., the processor fails its self-test when being reset), then that processor is simply not reintroduced back to the OS and in operational block **510** a message is generated by the system to provide notice of the loss of this processor and that the boot processor no longer has a spare. If the lockstep cannot be reestablished in block **508**, then if a LOL occurs in the new boot processor there will not be a spare processor for taking over its role (and thus the system may crash). If desired, the system administrator can schedule an orderly shutdown and re-boot in order to establish a new boot processor that has a hot spare available and/or schedule service of the system (e.g., for replacing the failed processor).

If the lockstep recovery is successful, then the former boot processor is assigned the role of active spare (e.g., by designating it as such in the device tree) in block **511**, and the firmware triggers the OS to recognize the active spare as available for executing instructions. Operation then returns to block **505**, and if a LOL is detected for the current boot processor, the procedure of blocks **506-512** may be repeated.

Turning to FIG. **6**, a general flow diagram for operation of certain embodiments is shown. In block **601**, the role of boot processor is assigned to one of a plurality of processors in a system. As described above, the role of boot processor may be so assigned in a device tree. Such role of boot processor may be assigned by a user (e.g., via a user interface) in certain embodiments, and assigned by the system's firmware in other embodiments.

In block **602**, the role of spare processor is assigned to another one of the processors. As described above, in certain embodiments the spare processor may be a hot spare, while in other embodiments the spare processor may be an active spare. As also described above, the role of spare processor may be so assigned in a device tree. Such role of spare processor may be assigned by a user (e.g., via a user interface) in certain embodiments, and assigned by the system's firmware in other embodiments.

In operational block **603**, the system firmware, responsive to detecting LOL for the boot processor, changes the assignment of the role of boot processor to the spare processor. This change in the role of boot processor occurs dynamically during system runtime, rather than requiring that the system be interrupted (e.g., shutdown and restarted, or crashed).

Certain systems include a plurality of partitions, wherein each partition includes a boot processor and OS. It should be understood that the above techniques for establishing a spare processor for use in assuming the role of boot processor in the event that a LOL is detected for the boot processor may be employed on each partition. Thus, a device tree may designate a boot processor and spare (e.g., active or hot) for each partition.

What is claimed is:

1. A system comprising:

a plurality of processors;

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; and

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, wherein said another of said plurality of processors that is assigned said role of spare processor is held in reserve as a hot spare processor until said loss of lockstep is detected for said boot processor.

2. The system of claim **1** wherein said data storage is non-volatile data storage.

3. The system of claim **1** further comprising:

a user interface operable to receive input specifying said one of said plurality of processors to be assigned said role of boot processor.

4. The system of claim 1 further comprising:
a user interface operable to receive input specifying said another of said plurality of processors to be assigned said role of spare processor.

5. The system of claim 1 wherein each of said plurality of processors is a processor module that includes a master processor and a slave processor.

6. The system of claim 1 wherein said another of said plurality of processors that is assigned said role of spare processor is an active spare before being transferred said role of boot processor.

7. The system of claim 1 further comprising:
an operating system, wherein, responsive to the logic transferring the role of boot processor to the spare processor, the operating system recognizes the spare processor as the boot processor.

8. The system of claim 1 wherein said plurality of processors are Itanium Processor Family (IPF) processors.

9. A method comprising:
assigning a role of boot processor to one of a plurality of processors in a system;
assigning a role of spare processor to another of said plurality of processors;
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; and
responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system.

10. The method of claim 9 wherein said assigning said role of boot processor comprises storing information to non-volatile data storage that identifies said one of said plurality of processors as said boot processor.

11. The method of claim 9 wherein said assigning said role of spare processor comprises storing information to non-volatile data storage that identifies said another of said plurality of processors as said spare processor.

12. The method of claim 9 further comprising:
determining at least one processor that is coupled to a common communication bus with said one of said plurality processors assigned the role of boot processor; and
selecting said another of said plurality of processors to be assigned the role of spare processor from said determined at least one processor.

13. The method of claim 9 further comprising:
during system boot-up, system firmware identifying the spare processor and holding said spare processor in reserve as a hot spare.

14. The method of claim 9 further comprising:
during system boot-up, system firmware booting the spare processor as an active spare.

15. The method of claim 9 wherein system firmware performs said changing operation.

16. A system comprising:
a plurality of lockstep processor modules that each includes a master processor and a slave processor that is in lockstep with said master processor;
a first communication bus with which at least a first two of said plurality of lockstep processor modules are communicatively coupled;
a second communication bus with which at least a second two of said plurality of lockstep processor modules are communicatively coupled; and
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, wherein said spare processor is communicatively coupled to the same one of said first and second communication buses as said boot processor.

17. The system of claim 16 further comprising:
system firmware operable, responsive to detecting loss of lockstep for said boot processor, changing the assignment of the role of boot processor to said spare processor.

18. The system of claim 17 wherein said system firmware performs said changing during system runtime without requiring an operating system of said system to be shutdown.

19. The system of claim 16 further comprising:
a user interface for receiving information selecting said one of said plurality of processors to be assigned the role of boot processor.

20. The system of claim 16 further comprising:
a user interface for receiving information selecting said another of said plurality of processors to be assigned the role of spare processor.

21. The system of claim 20 wherein said user interface presents for selection as said spare processor at least one processor that is communicatively to the same one of said first and second communication buses as said boot processor.

22. The system of claim 16 further comprising:
system firmware that, during system boot-up, identifies the spare processor and holds said spare processor in reserve as a hot spare.

23. The system of claim 16 further comprising:
system firmware that, during system boot-up, boots the spare processor as an active spare.

24. The system of claim 16 wherein said plurality of lockstep processor modules include Itanium Processor Family (IPF) processors.

25. A method comprising:
selecting a first one of a plurality of lockstep processor modules in a system as a boot processor, wherein each of said plurality of lockstep processor modules includes a master processor and a slave processor that is in lockstep with said master processor, and wherein said boot processor is communicatively coupled to a first of a plurality of communication buses;
selecting a second one of said plurality of lockstep processor modules as a spare processor, wherein said spare processor is selected as one that is communicatively coupled to the same communication bus as said boot processor; and
responsive to detecting loss of lockstep for said boot processor, said system causing said spare processor to dynamically assume the role of boot processor.

26. The method of claim 25 wherein said system causing said spare processor to dynamically assume the role of boot processor comprises:
said system causing said spare processor to dynamically assume the role of boot processor without shutting down an operating system of said system.

27. The method of claim 25 further comprising:
receiving input to said system specifying said first one of said plurality of lockstep processor modules to be said boot processor.

28. The method of claim 25 further comprising:
receiving input to said system specifying said second one of said plurality of lockstep processor modules to be said spare processor.

29. The method of claim 25 further comprising:
storing information identifying said selected first one of said plurality of lockstep processor modules as said boot

processor and identifying said selected second one of said plurality of lockstep processor modules as said spare processor.

30. The method of claim 29 wherein said system causing said spare processor to dynamically assume the role of boot processor comprises:

system firmware determining said spare processor from said stored information; and

said system firmware causing an operating system to recognize said spare processor as said boot processor.

* * * * *

EXHIBIT
1014



Intel® IA-64 Architecture Software Developer's Manual

Volume 2: IA-64 System Architecture

Revision 1.1

July 2000



Document Number: 245318-002

Firmware Overview 24

The IA-64 architecture defines three firmware layers: Processor Abstraction Layer (PAL), System Abstraction Layer (SAL), and Extensible Firmware Interface (EFI).

The PAL, SAL and EFI layers work together to handle the reset abort event. The reset abort handling performs processor and system initialization for operating system (OS) boot and provides a legacy-free API to the operating system loader. The PAL and SAL firmware layers work together to handle machine check aborts (MCA), initialization events (INIT), and platform management interrupt (PMI) handling. All three firmware layers also provide runtime procedure calls to abstract processor and platform functions that may vary across implementations.

This chapter will provide an overview of the firmware layers and how the firmware layers interact with each other as well as with the operating system. For the full architecture specifications of the PAL firmware please refer to Section 11, "IA-64 Processor Abstraction Layer". For full architecture specifications on SAL and EFI firmware layers please refer to Section 12.2, ".Related Documents" on page 12-3.

The PAL layer is developed by Intel Corporation and delivered with the processor. The SAL and EFI firmware is developed by the platform manufacturer and provide a means of supporting value added platform features from different vendors.

The interaction of the various functional firmware blocks with the processor, platform and operating system is shown in Figure 24-1, "Firmware Model" on page 24-2.

24.1 Processor Boot Flow Overview

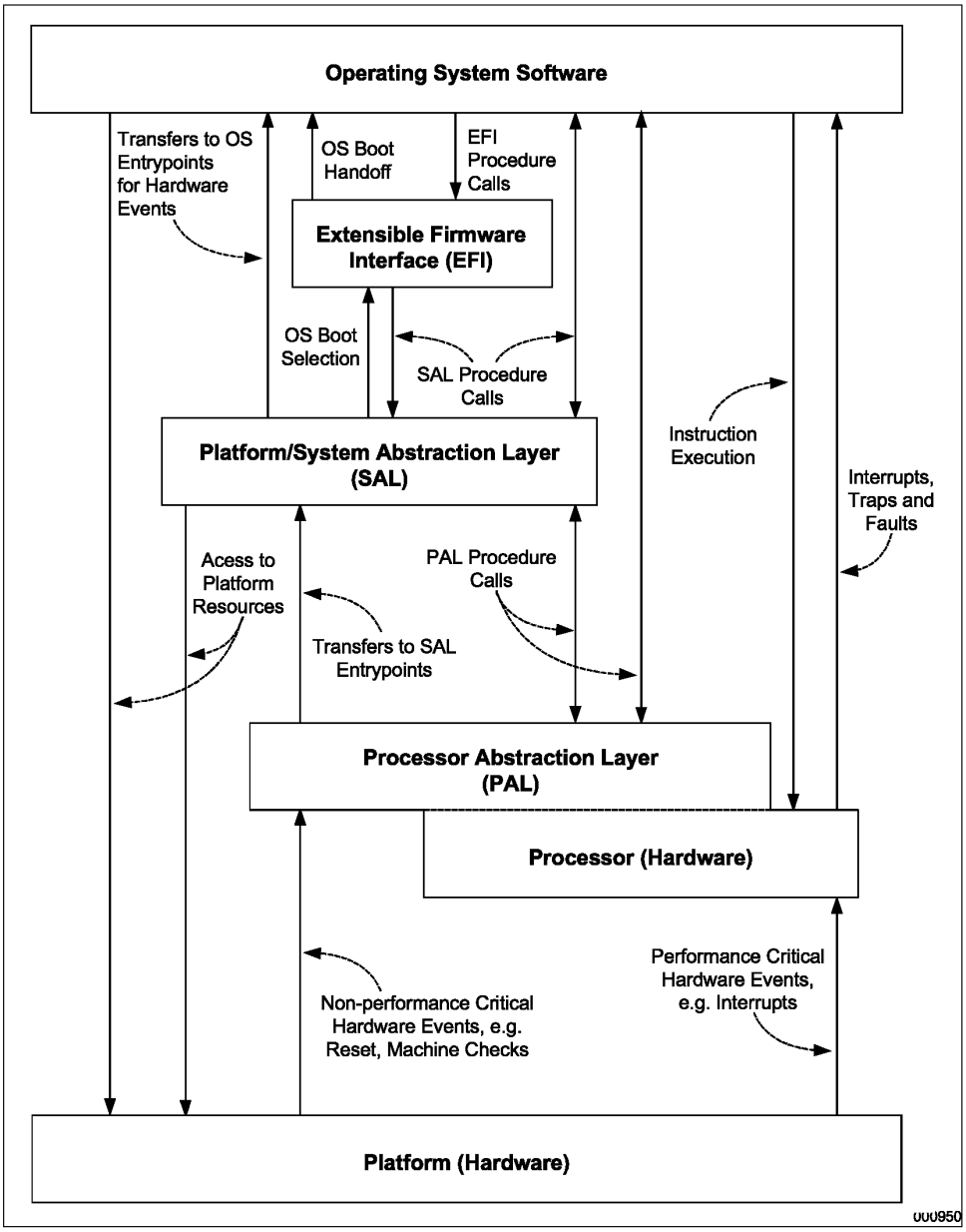
24.1.1 Firmware Boot Flow

Upon detection of a reset event on an IA-64 processor, execution begins at an architected entry point inside of PAL. This PAL code will verify the integrity of the PAL code and may perform some basic processor testing. PAL will then branch to an entry point within the SAL firmware. This first branch to SAL is to determine if a firmware update is needed requiring re-programming of the firmware code. If no firmware update is needed SAL will branch back to PAL.

PAL now performs additional processor testing and initialization. These first processor tests are performed without platform memory. PAL indicates the outcome of the testing and branches to an entry point within SAL firmware for the second time. SAL will now begin platform testing and initialization.

The order of steps within the SAL firmware is platform implementation dependent and may vary. In general, the SAL firmware selects a Bootstrap processor (BSP) in multi-processor (MP) configurations early in the boot sequence. Next, SAL will find and initialize memory and invoke PAL procedures to conduct additional processor tests to ensure the health of the processors. SAL then initializes the system fabric and platform devices. SAL will display the progress of the boot on the video output device and permit the user to change the system configuration.

Figure 24-1. Firmware Model



The SAL firmware layer hands off control to the EFI firmware layer which incorporates a Boot Manager. The EFI firmware specification [EFI] enables booting from a variety of mass storage devices such as hard disk, CD, DVD as well as remote boot via a network. At a minimum, one of the mass storage devices contains an EFI system partition.

The EFI Boot Manager displays the list of operating system choices and permits the user to select the operating system for booting. To support this functionality, the OS setup program stores the

boot paths of the OS loaders and boot options in non-volatile storage managed by the EFI. The EFI reserves the environment variables Boot#### (#### represents values 0000 to 0xFFFF) for this purpose. The OS setup program must also store the OS loader binary images within the EFI System Partition. The EFI Boot Manager will also allow the user to add boot options, delete boot options, launch an EFI application, and set the auto-boot time out value.

The EFI System Partition also contains EFI drivers that will be loaded by the EFI firmware prior to transfer of control to an IA-64 OS loader. The floating-point software assist (FPSWA) library is included in these EFI drivers. The FPSWA library may be invoked by the OS during floating-point exception faults and traps. Please see Section 19.1.1, "The Software Assistance Exceptions (Faults and Traps)" on page 19-1 for more information on the usage of this library.

If the user elects to boot an IA-32 operating system, the EFI will load 512 bytes of the first level boot code (Master Boot Record in the case of disk devices) at location 0x7C00. Next, EFI will remove its memory footprint and returns to the SAL firmware. The SAL will then invoke a PAL procedure to set up the IA-32 System environment and jump to the boot code at 0x7C00. The boot code will load an IA-32 OS loader which, in turn, loads and transfers control to the IA-32 OS kernel.

If the user elects to boot an IA-64 operating system, the EFI loads the appropriate OS loader from the EFI System Partition and passes control to it. The OS loader will load other files including the OS kernel from an OS partition using the EFI boot services which provides a legacy free API interface to the OS loader. EFI uses SAL to access low level platform resources. The interfaces between EFI and SAL are platform firmware implementation dependent and not relevant for the OS loader developers.

The OS loader can obtain information about the memory map usage of the firmware by making the EFI procedure call GetMemoryMap(). This procedure provides information related to the size and attributes of the memory regions currently used by firmware.

The OS loader will then jump to the OS kernel that takes control of the system. Until this point, SAL retained control of key system resources such as the Interrupt Vector Table and provided the necessary interrupt, trap and fault handlers.

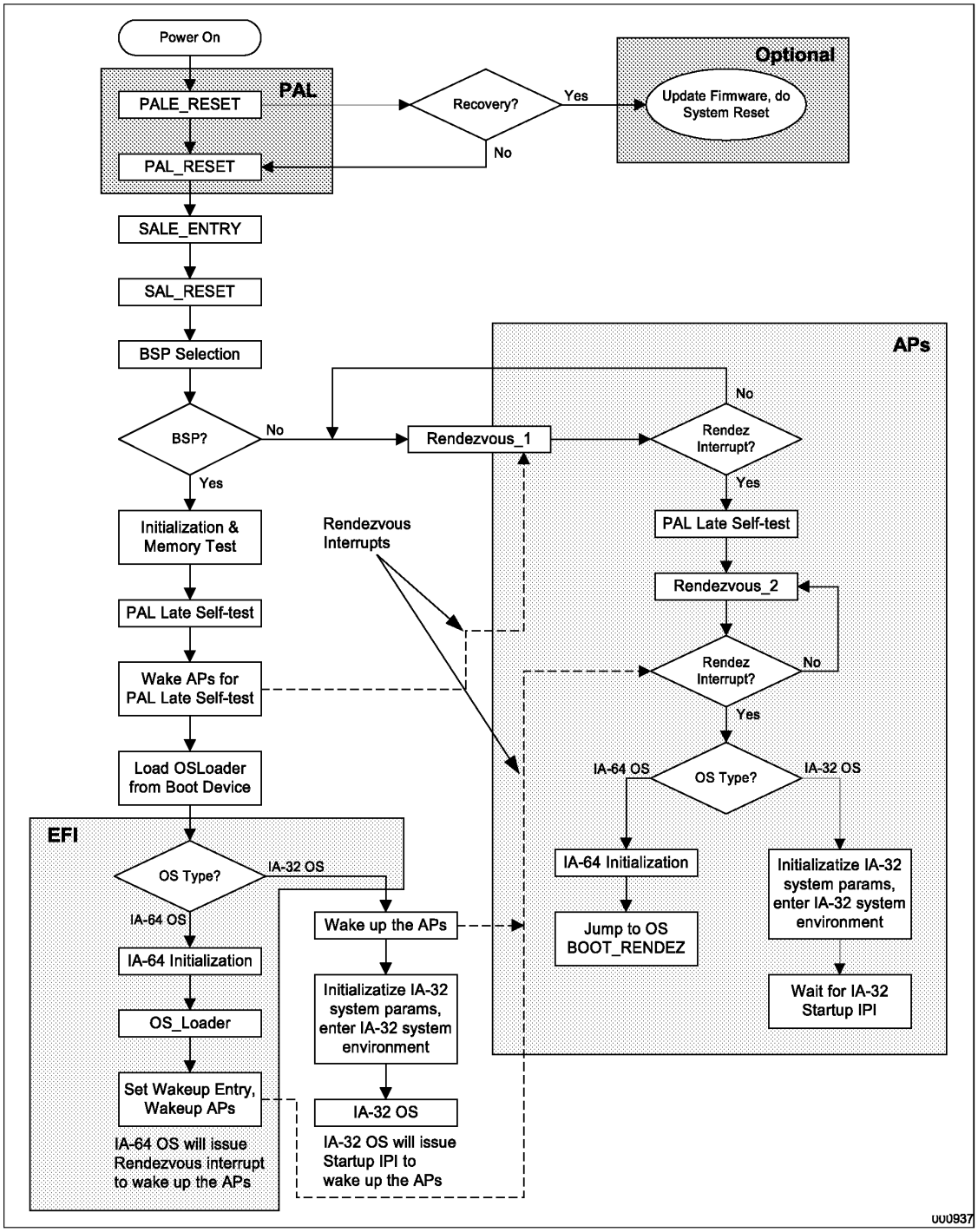
Figure 24-2, "Control Flow of Boot Process in a Multi-processor Configuration" on page 24-4 depicts the booting steps in a MP configuration.

24.1.2 Operating System Boot Steps

The firmware will initialize the processor(s) and platform to a specific state before handing off to the operating system boot loader. The boot loader is then responsible for copying the operating system from some storage medium into memory for running. Once this is done the operating system will need to initialize some key registers before entering into a higher level language code such as C. This section will describe code that an OS will need to execute in order to initialize system registers for preparing an OS to run in virtual mode and handle interrupts. Appendix B, "Code Examples" provides the IA-64 sample assembly code described in this section.

Assuming the specific operating system boot loader hands off to the OS kernel in physical mode, the operating system should first disable interrupts and interrupt collection via the PSR. This is done to avoid taking external interrupts from timers, etc and also prepares for writing specific system registers that require PSR.ic to be 0 when written.

Figure 24-2. Control Flow of Boot Process in a Multi-processor Configuration



Next the operating system startup code invalidates the ALAT via the `invala` instruction. The invala in complete form will invalidate all entries in the ALAT.

The register stack should be invalidated. This can be done by setting the Register Stack Configuration Register (RSC) to zero followed by a loadrs instruction. Setting the RSC to zero will put the register stack in enforced lazy mode and set the RSC.loadrs, load distance to tear point, to zero. The loadrs will invalidate all stacked registers outside current frame.

The region registers and protection key registers are then initialized with operating system implementation dependent values. For example, the OS will initialize the region register with a preferred page size. It would also disable the VHPT until it was ready for it. In the example, all region registers are initialized with an 8-KB page size.

An OS must setup a kernel stack pointer and backing store pointer for the register stack. The stack pointer (GR12) is set to the OS kernel stack area with scratch space to cover calling conventions. AR.RSC must be set to enforced lazy mode before writing to the bspstore register. Initializing the bspstore has effects on all 3 RSE pointers (BSP, BSPSTORE, and RSE.BspLoad).

In order for the operating systems to handle interruptions, the operating system interrupt vector table base address must be set up. The size of the vector table is 32K bytes and is 32K byte aligned. Setting the location of the table is accomplished by moving the address into CR.IVA.

Operating systems setup system address translations for the kernel text and data by using the translation insertion format described in Section 4.1.1.5, "Translation Insertion Format". A combination of a general register, Interruption TLB Insertion Register (ITIR), and the Interruption Faulting Address register (IFA) are used to insert entries into the TLB. To void TLB faults on specific text and data areas the operating system can lock critical virtual memory translations in the TLB by use of Translation Register (TR) section of the TLB. The entries are placed into a TR via the Insert Translation Register (itr) instruction. The translation will remain unless the software issues the Purge Translation (ptr) instruction. Other important areas might be locked also, such as entries for memory mapped I/O, etc.

After the initial translations have been entered, the OS can make final preparations for enabling virtual addressing. The OS needs to set several important bits in the IPSR, such as data address translation (dt), register stack translation (rt), instruction address translation (it), enabling interruption collection (ic), and setting the specific register bank (bn).

The Default Control Register (DCR) specifies the default parameters for PSR values on interruption, some additional global controls, and whether speculative load faults can be deferred. The example defers all speculation faults. Also, if the operating system is utilizing the performance monitors then the DCR.pp bit should be set so that on interruption the PSR.pp bit will be set.

The global pointer (GR1) should point to the global data area. It must be setup properly before using higher level languages such as C. The startup code should also set the following registers to zero, the Interruption Function State (CR.IFS, to set frame marker to zero), and AR.RNAT (to make sure no NaT bits are set before OS kernel begins using the RSE.

Before enabling virtual addressing, the Interruption Instruction Bundle Pointer (IIP) is set to point a virtual address. This is done so when the return from interruption instruction (`rfi`) is executed the instruction fetched will have a virtual address. The rfi will switch modes based on IPSR values which are moved into the PSR. The IIP value becomes the new IP.

EXHIBIT
1012

UNIX Administration

A Comprehensive Sourcebook for Effective Systems and Network Management

© 2002 by CRC Press LLC

4

System Startup and Shutdown

4.1 Introductory Notes

UNIX systems run continuously under normal circumstances. Shutting down and powering-off a UNIX system should be done rarely, usually only when a hardware upgrade is being performed or a system is being allocated, or occasionally when another action requiring a system shutdown is performed. In real life, system shutdown is more frequent, because unpredictable situations always occur.

Power-cycling a UNIX system is not the only way the system can be shut down. Rebooting is also a familiar task for any UNIX administrator; UNIX administrators know well how system rebooting can be healthy for overall system maintenance.

Nevertheless, keeping the UNIX system running is the most visible task of a system administrator. If the system crashes, everyone will complain, your phone will ring constantly, and you will find yourself anxiously trying to fix the problem and bring the system back into production. Quickly you will learn how important the system you are in charge of really is, and how many users depend on it. Even more important, you will learn how crucial a smooth, fast, proper system startup can be.

This chapter covers topics related to normal UNIX system startup and shutdown procedures. Invoking a system startup and shutdown is quite simple; the main requirement is to be the superuser on the system (an easy task for an administrator). On the other hand, making the system behave correctly, especially during startup, requires a great deal of knowledge and administrative skill. Proper system startup is supposed to customize and set the myriad of existing system configuration files that will control each portion of the UNIX system. Some of these files include system-related configuration data, but there are also site-added applications; the bottom line is that the system should be fully operational after any system startup.

Given the complexity of properly configuring the system startup, this chapter could easily be located at the end of the overall text, rather than at its beginning. However, discussing the administration of a running UNIX system without knowing how that system came to be running seems strange; it is as though we are talking about administering a nonexistent UNIX system. So this material remains in the beginning by design; it will focus on the topic of global system startup and shutdown, and we will return to individual startup and shutdown issues later, whenever it is appropriate in discussing specific UNIX topics.

© 2002 by CRC Press LLC

From an administrative standpoint, system shutdown is the simpler procedure; at the end of the procedure a system must terminate all running processes, dismount all filesystems, and stop any other system activity. System shutdown works even if we never touch the default shutdown procedure — or perhaps it is better to say it *mostly* works, because the author of this text has witnessed a UNIX system that could not be shut down from the command line, and the only choice was to power-cycle the system. Our administrative task is to provide a graceful system shutdown. Everything must be stopped in a regular way, or the administrator will have to use the brute force method of power-cycling. System startup, on the other hand, must be done properly or the system will never come up. Obviously, more attention should be paid to system startup, and we will spend much more time discussing the startup procedure than the shutdown process.

System startup is often referred to as ***system booting***. Although "booting" specifies only one phase in the overall system startup, the two terms are commonly interchanged, as you will see in this chapter. Strictly speaking, system startup has a broader meaning than system booting.

All UNIX systems must be shut down in a regular way before any further action can be taken. You should never directly power-off UNIX systems (such as DOS-based PCs); the shutdown procedure must be implemented, otherwise disk data integrity can be corrupted (a UNIX filesystem could be damaged). The corruption can range between a relatively benign loss of data to heavy filesystem damage, which in the worst case scenario can leave a system unbootable.

The two major UNIX platforms BSD and System V have different startup and shutdown procedures, with, of course, the main differences occurring in startup. Among existing commercial UNIX flavors, the System V approach is more common; it provides more flexibility and some other administrative advantages. However, the BSD approach is somewhat easier to understand, and we will start our discussion with the BSD startup/shutdown procedure. Once the startup/shutdown concept is well understood, it will be easy to continue with the System V procedure.

4.2 System Startup

The system startup procedure is a continuous process that a UNIX system goes through, from its initial hardware-determined stage until the final production-ready stage. However, this unique system journey passes through several distinct phases, and each of these phases has its specific characteristics. The startup phases, listed in the order they occur, are:

- Bootstrap program execution
- Kernel execution
- *rc* system initialization
- Terminal line initialization

It is easier to understand the system startup procedure when the whole process is divided into several phases and each of the phases is analyzed separately, so this is the approach we will take. Although each of the listed phases is equally important for successful system startup, the ***system initialization*** phase requires the most administrative attention, so most of the following discussion will address this phase.

© 2002 by CRC Press LLC

In each of the startup phases, the system learns enough to execute the next phase. Each phase contributes a bit to the overall system startup. At the very beginning, the system does not know very much; at the very end, the system is ready for multi-user operations.

4.2.1 The Bootstrap Program

The origin of the word *boot* (as in, "to boot the system") is *bootstrapping*, which is the process of bringing a computer system to life and ready for use. ("Bootstrapping" is actually the "nerd word" for starting up a computer.) The computer system itself is only a collection of hardware resources (registers, arithmetic/logical unit, program counter, memories, etc.) capable of executing a sequence of instructions that make a program. The program, stored in the computer's memory (any kind of memory: ROM, RAM, external magnetic memory, etc.), defines the system's activity at every moment, including its first steps during the system startup.

An initial program, the ***bootstrap program***, must be stored in the fast nonvolatile memory directly accessible by a processor, or CPU (CPU stands for central processing unit and is another term for a processor). This portion of the computer memory is known as internal read-only memory (ROM). The execution of the bootstrap program is always automatically initiated when the system is powered-on or when a system hardware reset is applied. It is also initiated when the system is rebooted from the system console. Only the initial part of the bootstrap program, the part sufficient to bring the system into a workable state to deal with other memory types, must be stored in ROM. Once this level is achieved, the bootstrap program execution can be continued from another nonvolatile media such as a hard disk, a floppy disk, a tape, or a CD-ROM, or even through the network from a boot-server (in the case of diskless workstations). For UNIX systems, regular system booting is commonly executed from a hard disk, while first-time UNIX OS installation is performed from a CD-ROM (not long ago, a tape was used).

The system has to learn enough from the ROM to be able to access a disk to continue the bootstrap program, but it still assumes a simple flat data structure on the disk. A complex disk data organization such as the UNIX filesystem data structure is still too complicated for the system at this stage; more learning is needed to deal with a filesystem. That is why the rest of the bootstrap program is stored in a special part of the disk known as the *boot partition* (sometimes also known as the *boot segment*). The main characteristic of the boot partition is its easy access and flat data structure, so the system is able to continue with the bootstrap program execution, and further learning.

The ratio of the percentage of the bootstrap program stored in the ROM versus the disk boot partition varied through time. In the early days of UNIX when only low capacity, expensive ROM was available, the first part of the bootstrap program was reduced to the bare minimum size. Today, systems include high-density ROM sufficient to store quite sophisticated bootstrap programs; this makes boot partitions less important, although they are still a part of every system startup.

Once the bootstrap program is completely executed, the system is knowledgeable enough to continue with the kernel execution.

Traditionally UNIX presents the only OS running on underlying hardware; and traditionally this is a proprietary hardware for that UNIX flavor. This fact makes a booting process unique and quite straightforward. However, once PC hardware also became common in the UNIX arena, a more flexible booting, with UNIX as one of several choices, emerged as a preferable system characteristic. Linux is an example.

© 2002 by CRC Press LLC

On the Linux platform, the three most common booting mechanisms are:

- To boot Linux from the floppy, and leave hard drive for other OSs
- To use the ***Linux loader (LILO)***, the most common case
- To run ***Loadlin***, an MS-DOS program that boots Linux from within DOS

What is exceptional with LILO is the possibility of configuring this loader in different ways to match different needs; multiple-choice booting, including a non-UNIX startup, is also possible. The configured loader should then be installed in the boot sector of the first disk, known as MBR (master boot record). When the system is started, the PC BIOS transfers control to MBR and triggers a corresponding LILO booting. Linux provides an easy LILO configuration through its */etc/lilo.conf* configuration file, and the command ***lilo*** for its installation as MBR.

4.2.2 The Kernel Execution

The bootstrap program is responsible for loading the *UNIX kernel* into the system memory. The *kernel image*, originally named ***unix*** under System V, or ***vmunix*** under BSD, is intentionally located in the *root filesystem*, because the root filesystem is the first filesystem the system mounts to access data. Mounting is a UNIX-specific procedure that makes data on the disk accessible. (We discuss this issue in great detail in following chapters.) In the past, the kernel image was located in the root directory for easier access, but today, it usually resides in a separate subdirectory. We do not refer to "mounting" the kernel; we usually just say that the kernel image was loaded into the system memory and its execution was started.

The *kernel* manages all system hardware; all hardware drivers are part of the kernel, and the only OS access to the system hardware is through the kernel. Therefore the system hardware will be available upon the completion of this phase.

Once control passes to the *kernel*, it prepares itself to run the system by initializing its internal tables and completing the hardware diagnostics that are part of the boot process. The level of diagnostics implemented varies from one UNIX flavor to another. At the very end, the kernel verifies the integrity of the root filesystem and remounts it, and starts three programs that create three basic processes. Two of them, named ***kernel processes***, function wholly within the kernel in the kernel's privileged execution mode. They are actually portions of the kernel itself, only "dressed" like processes for scheduling reasons.

On BSD systems, the two processes are:

1. ***Swapper*** (process #0), responsible for the "*swapping*" — to schedule the transfer of whole processes between the main system memory and a mandatory swap partition on the primary disk when system resources are low
2. ***Pagedeamon*** (process #2), responsible for supporting the memory-management system regarding paging — a regular transfer of data in the pages between the main system memory and the disk

On System V systems, the processes are named differently: ***sched*** for the process #0, while the process #2 is replaced with *various memory handlers*.

The third process created by the system is the ***init*** process (process #1), which performs all administrative tasks during the system startup and shutdown. The ***init*** process is an extremely important process that enables the creation of all subsequent processes (in UNIX a process can be created only by another parent process). The ***init*** process has the PID=1, and it is the ancestor of all subsequent UNIX processes and the direct parent of each user's login shell.

© 2002 by CRC Press LLC

In the case of diskless workstations, the procedure is slightly different. Obviously, the kernel cannot be read from a nonexisting root filesystem; therefore, it must be downloaded from the network. Further kernel activities are adapted to the diskless environment.

The kernel is quite verbose and it prints messages on the console that report on the current execution status, total memory used and free, and some other information. However, the information reported varies among different UNIX flavors.

4.2.3 The Overall System Initialization

The ***init*** process does the rest of the work needed to bring the system into its final stage:

- Mounting the remaining local disk partitions
- Performing some filesystem cleanup
- Bringing on major UNIX subsystems (accounting, printing, etc.)
- Setting the system's name and time zone
- Starting the network
- Mounting remote filesystems
- Enabling user logins

4.2.3.1 rc Initialization Scripts

Most of the initialization activities are specified and carried out by means of the system ***rc initialization scripts*** stored in the ***/etc*** directory and its subdirectories. *Rc initialization scripts* are usually named in the way that they include the acronym **rc** as part of their names (as a prefix, a suffix, or in a fullpath name). **rc** stands for **run-command** and basically explains the purpose of the scripts. These mostly ***Bourne shell programs*** are organized differently on BSD and System V platforms, although their purpose is the same. As with any other script, rc initialization scripts are readable, so we can manage them in a very comprehensive way. Besides that, rc scripts are sufficiently verbose during execution, and this is a great help if the system hangs midway through the startup, or if there are any other related problems.

Main administrative activities are related to this phase. System site-related customization means editing the rc initialization scripts. Any system upgrade means to upgrade (or to add) rc initialization scripts; any startup modification means to do something with rc initialization scripts. The rest of this section exclusively addresses these issues. Afterward, a full picture of the necessary administration in this segment should be complete.

4.2.3.2 Terminal Line Initialization

The terminal line initialization is a part of the overall system initialization; however, the implemented initialization technique is quite different than that in the *rc system initialization*, which is sufficient reason to handle this topic separately. UNIX is extremely cautious with terminal line initialization — terminal lines are "gates" to the outside world. Users access the system via terminal lines, and the essence of UNIX existence is to serve users.

Once the initialization scripts have been executed, the system is fully operational, except for the fact that no one can log in to the system. In order to provide login via a particular terminal line, there must be a corresponding controlling process listening on it (usually the ***getty*** process, but the ***ttymon*** is used on the Solaris platform). At the final initialization phase, ***init*** spawns the ***getty*** processes to all indicated terminals and the startup procedure

© 2002 by CRC Press LLC

is completed. Today, users typically log in over a network using pseudo-terminals; however, the ***getty*** program is still doing its job.

Terminal line initialization is fully covered in Chapter 11.

4.2.4 System States

Once the initialization activities are completed, the UNIX system enters the ***multi-user mode***, and users may log in to the system. But ***init*** can also place the system in ***single-user mode*** instead of completing the initialization tasks required for multi-user mode. The single-user mode corresponds to a functionally reduced UNIX system. In single-user mode, a UNIX system looks very much like a personal computer. The single-user mode is primarily dedicated to administrative and maintenance activities that require complete control over the system. The user has all superuser privileges.

In some cases, the system will automatically enter single-user mode if there are any problems in the boot process that the system cannot handle on its own (for example, filesystem problems that **fsck** cannot fix), so the system administrator must resolve the problem. The ***init*** simply spawns the Bourne shell on the system's console and waits for it to terminate before continuing with the rest of the startup sequence. Entering <CTRL-D> or the exit command from the shell prompt can terminate the spawned single-user shell. Once this is done, the system may continue into multi-user mode.

Single-user mode represents a minimal system startup with no daemons running, so many UNIX facilities are disabled. Only the root filesystem is mounted (in the most common case) and a restricted number of commands are available (commands residing in the root filesystem). Under normal circumstances, other filesystems can be mounted by hand to access other commands.

Single-user mode can be a security problem for a system, because full control over the system is granted. On older UNIX systems, no password was required, but physical access to the system was required in the single-user mode. On some systems, a front panel lock with *normal* (*secure*) vs. *maintenance* (*service*) key positions enabled multi-user vs. single-user mode; the system protection was the key, and only authorized personnel could acquire the key. Modern UNIX systems usually require a root password to enter single-user mode. None of these approaches are perfect, and all of them have some disadvantage. A request for the root password could make difficulties under different circumstances, if the root password was forgotten.

While the BSD flavored system could be in one of three possible states — *off*, *single-user*, and *multi-user mode* — the System V platform explicitly defines a series of system states, called *run-levels* designated by a one-character name. System V run-levels are flavor dependent; an example is listed in the following table:

Run-Level	Name and Uses
0	Power-down state => safe to power-off the system
1	Administrative state
s or S	Single-user mode (on many systems same as 1)
2	Multi-user mode for stand-alone system
3	Multi-user mode for networked system, possibly sharing disks with other systems => via RFS, TCP/IP, and NFS, or some other protocol
4	Unused => can be user defined locally
5	Firmware state => for maintenance and running diagnostics, and for booting from an alternate not-root disk
6	Shutdown and reboot state => to reboot system from some running state (s, 2, 3, or 4); the system is taken down (to run-level 0) and then rebooted back

© 2002 by CRC Press LLC

To display the current system run-level, the following command is available:

```
$ who -r
. run-level   3   Mar   14   11:14   3   0   S
```

The system was taken to run-level 3, from run-level S, via run-level 0, on March 14, at 11:14. The leading dot is by default at the beginning of the line.

On the System V platform, movement between run-levels is managed by ***init***, and each run-level is controlled by its own set of initialization scripts.

4.2.5 The Outlook of a Startup Procedure

UNIX systems are configured to boot automatically when powered-on. If this is not possible, systems enter some form of the "ROM monitor mode" — a restricted ROM resident command interpreter that enables essential diagnostics, booting, and some other basic system activities. The ROM monitor mode is also the state that the system enters after being shut down; in that state, a system can be safely powered off. On some systems there is also a keystroke combination to enter this mode — for example on Sun Microsystems systems, the key (STOP-A) followed by the specific ROM monitor prompt "*OK*>."

The ROM monitor always provides the boot command, specified as "*b*" or "*boot*," among the other commands it provides. Certain options sufficient to control the system startup when problems are encountered (to boot the system from different media, into different modes, etc.) are also provided. The default booting media is the hard disk.

On old UNIX systems, manual booting from the ROM monitor was a two-stage procedure:

1. The boot command first loaded a boot program with a stand-alone shell (actually a mini-operating system).
2. A second command was then issued in a stand-alone shell to load UNIX kernel.

This two-step procedure looked like this:

```
>b
$$ unix
```

Different prompts specify two steps in the boot procedure. The technology available in the past limited the bootstrap program possibilities, which led to a more complicated startup procedure.

Today all UNIX flavors provide a relatively verbose system startup; a number of messages are directed to the console indicating the stage and status of the startup procedure. It is highly recommended that you monitor the system startup on the console. Otherwise, some trouble messages can remain undetected, which leads to a high probability for later surprises.

The startup sequences for two system user modes are presented in Figures 4.1 and 4.2. The UNIX system named "***atlas***" is running Solaris 2.x.; brief comments follow.

© 2002 by CRC Press LLC




```
---------- bootstrap program starts ----------
SPARCstation 20 (1 X 390Z50), Keyboard Present
ROM Rev. 2.19, 32 MB memory installed, Serial #7491530
Ethernet address 8:0:20:72:4f:ca, Host ID: 72724fca.

Rebooting with command:
Boot device: /iommu/sbus/espdma@f, 400000/esp@f, 800000/sd@3,0 File and args:
---------- bootstrap program ends, and kernel starts ----------
SunOS Release 5.4 Version generic [UNIX (R) System V Release 4.0]
Copyright (c) 1983-1984, Sun Microsystems, Inc.
---------- kernel ends, and rc initialization starts ----------
configuring network interfaces: le0.
Hostname: atlas
The system is coming up. Please wait.
checking ufs filesystems
/dev/rdsk/c0t2d0s6: is clean
/dev/rdsk/c0t3d0s7: is clean
/dev/rdsk/c0t2d0s0: is clean
Flushing routing table:
add net default: gateway 146.95.8.250
starting rpc services: rcpbind keyserv kerbd done.
Setting netmask of le0 to 255.255.255.0
Setting default interface for multicast: add net 244.0.0.0: gateway atlas.ph.hunter.cuny.edu
syslog service started.
Print services started.
volume management starting.
HTTP service starting.
The system is ready.
---------- rc initialization ends, and terminal line initialization starts ----------

atlas console login:
```

FIGURE 4.1
An illustration of a multiple-user startup sequence.

The Sun logo and first five lines are printed from the ***bootstrap*** program. These lines list basic system configuration and identification data, as well as the kind of boot device. The somewhat cryptic description of a boot device indicates an SCSI disk. The ***kernel*** prints only two identification lines that include the system version and release. Other lines are printed from initialization scripts invoked by the program ***init***. One of the lines indicates that the system was customized. The message that indicates the start of the HTTP service is not a part of a regular OS installation — obviously, this site has been customized to provide an Internet service. At the end, the login prompt is displayed upon the console initialization.

The startup procedure includes filesystem checking, one of the most important activities performed by the **fsck** utility (**fsck** is discussed in greater detail in Chapter 5). The filesystem verifications are different on BSD and System V platforms. BSD checks all filesystems on every boot; System V does not check filesystems if they were dismounted normally when the system last went down (the **fsstat** command is used for this purpose), and faster booting is enabled. Filesystem checking can result in the display of many messages depending on the current filesystem status. If more serious filesystem corruption is encountered, the system is left in single-user mode, and manual filesystem checking and repair by the administrator may be required.

A single-user startup sequence is much shorter, and it includes the *boot* and *kernel* lines. The next two lines about the network interface configuration and host's name are printed from corresponding initialization scripts involved in the system single-user startup. Finally, the console is activated and the user is informed of two possibilities:

© 2002 by CRC Press LLC

1. Enter the system in single-user mode by entering the root password
2. Or continue with multi-user startup by entering [Ctrl-D]

If [Ctrl-D] is entered, the system continues with the multi-user startup, as in the previous case.




```
---------- bootstrap program starts ----------
SPARCstation 20 (1 X 390Z50), Keyboard Present
ROM Rev. 2.19, 32 MB memory installed, Serial #7491530
Ethernet address 8:0:20:72:4f:ca, Host ID: 72724fca.

Rebooting with command: -s
Boot device: /iommu/sbus/espdma@f, 400000/esp@f, 800000/sd@3, 0 File and args: -s
---------- bootstrap program ends, and kernel starts ----------
SunOS Release 5.4 Version generic [UNIX (R) System V Release 4.0]
Copyright (c) 1983-1984, Sun Microsystems, Inc.
---------- kernel ends, and single-user rc initialization starts ----------
configuring network interfaces: le0.
Hostname: atlas

INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance):

>>>>>>>>>>>>>>> Since Ctrl-D is entered <<<<<<<<<<<<<<<
---------- rc initialization continues ----------

INIT: New run level: 3
The system is coming up. Please wait.
checking ufs filesystems
/dev/rdsk/c0t2d0s6: is clean
/dev/rdsk/c0t3d0s7: is clean
/dev/rdsk/c0t2d0s0: is clean
Flushing routing table:
add net default: gateway 146.95.8.250
starting rpc services: rcpbind keyserv kerbd done.
Setting netmask of le0 to 255.255.255.0
Setting default interface for multicast: add net 244.0.0.0: gateway atlas.ph.hunter.cuny.edu
syslog service started.
Print services started.
volume management starting.
HTTP service starting.
The system is ready.
---------- rc initialization ends, and terminal line initialization starts ----------

atlas console login:
```

FIGURE 4.2
An illustration of a single-user startup sequence.

4.2.6 Initialization Scripts

Once the ***init*** process is born, the system startup is determined by a series of rc initialization scripts which define a detailed procedure to bring the system into the multi-user mode. This is the most common case, although other system modes (run-levels) are also possible. These files control all custom-defined and site-dependent items (there are multiple rc initialization scripts), and they are executed sequentially. Generally, rc initialization scripts represent Bourne shell script files, executable at any time and on any UNIX platform. (The Bourne shell is the default shell, and it is available at the very early system stage on every UNIX platform.) The rc initialization scripts do not differ from any other shell script, except at the time of their execution. (This, by the way, is why the prefix "rc" is used in their description, as well as in the name.) However, they can also be executed from the

© 2002 by CRC Press LLC

command line at any time, and administrators can make full use of this opportunity: on System V, individual function-specific initialization scripts are often used to stop and start specific UNIX functions during regular system production. On modern UNIX platforms, sometimes Korn shell rc initialization scripts are also included (for example, on the HP-UX platform) which indicated the early availability of the Korn shell.

Understanding rc initialization scripts is a vital part of system administration — this is the place for system customization. A system administrator must be familiar with these files, their locations and, in many cases, their contents. Only then is full control over the system startup possible, and quick corrective action can follow any problem encountered during system boot time. Each modification in the initialization scripts must be done very carefully with respect for the basic administrative rule: save original script files before making any changes. If this rule is not followed, various problems can ensue.

Despite the fact that rc initialization scripts on both UNIX platforms BSD and System V serve the same purpose, the mechanisms by which they are initiated and executed are quite different. These differences require great attention, knowledge, and skills from system administrators working in a heterogeneous environment, which is very common today. Today, the System V rc approach prevails — the System V organization of the rc initialization scripts offers more flexibility and other administrative advantages. We will discuss System V initialization in greater detail after a quick survey of the BSD-style initialization.

4.3 BSD Initialization

4.3.1 The BSD *rc* Scripts

Originally, the BSD initialization was controlled only by two rc initialization scripts: ***/etc/rc*** and ***/etc/rc.local***. A general system initialization was supported by the */etc/rc* script, while the */etc/rc.local* script referred to a local site, i.e., to issues that should be customized (probably a more appropriate script name would be "*rc.site*" to avoid any possible confusion toward the logical association with a "network-local relationship"). During system booting to the multi-user mode, ***init*** executed the *rc* script, which in turn executed the *rc.local* script. If a single-user boot was performed, scripts were only partially executed; the remaining parts were executed when the single-user shell was exited.

Having only two rc initialization scripts would lead one to believe that system maintenance was easy, but in fact the reality is quite the opposite. The work required for system initialization remained the same, regardless of how many rc scripts were involved, and huge rc script files were more difficult to manage and more vulnerable to corruption during modification. It could be very difficult to find an appropriate control sequence, items were often doubled, and so on.

SunOS introduced additional script files: ***/etc/rc.boot*** and ***/etc/rc.single***. The program ***init*** invokes first *rc.boot* script and from there *rc.single* (regardless of whether the system is booting to single vs. multi-user mode); then the */etc/rc* and */etc/rc.local* files follow.

4.3.2 BSD Initialization Sequence

For a clearer picture, the block diagram of the SunOS execution sequence is presented in Figure 4.3 (it is assumed the system is booting from the local disk). The SunOS organization made a clear distinction between single and multiple-user modes; it was immediately easier to follow any problems that developed in the system booting.

© 2002 by CRC Press LLC

Electronic Acknowledgement Receipt	
EFS ID:	5416814
Application Number:	10972588
International Application Number:	
Confirmation Number:	1079
Title of Invention:	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
First Named Inventor/Applicant Name:	Scott L. Michaelis
Customer Number:	22879
Filer:	Robert Wayne Nelson/Kalyn Black
Filer Authorized By:	Robert Wayne Nelson
Attorney Docket Number:	200404973-1
Receipt Date:	28-MAY-2009
Filing Date:	25-OCT-2004
Time Stamp:	19:27:01
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	E-IDS.pdf	538632 987f6f6549b8b448c8f85502143088f6d95c7d78aa	no	1
Warnings:					
Information:					

2	Information Disclosure Statement (IDS) Filed (SB/08)	sb0008a-1.pdf	287377 a18fdc83b08748aeeb17a921c0a582acd8bs6fe12	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Foreign Reference	WO_01_45421.pdf	861429 c925ea9898094ac3b16c9377ac9869d0b7e0722b	no	21
Warnings:					
Information:					
		Total Files Size (in bytes):	1684438		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200404973-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Scott L. Michaelis	Confirmation No.: 1079
Application No.: 10/972,588	Examiner: Le, Dieu Minh T.
Filing Date: 10/25/2004	Group Art Unit: 2114

Title: 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

Mail Stop Amendment
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

This Information Disclosure Statement is submitted:

☒ under 37 CFR 1.97 (b), or
(Within three months of filing national application; or date of entry of national application; or before mailing date of first action on the merits; whichever occurs last)

☐ under 37 CFR 1.97 (c), together with either a:
- ☐ Statement under 37 CFR 1.97 (e), or
- ☐ a $180 fee under 37 CFR 1.17 (p), or

(After the CFR 1.97 (b) time period, but before final action or notice of allowance, whichever occurs first)

☐ under 37 CFR 1.97 (d), together with a:
- ☐ Statement under 37 CFR 1.97 (e)(1) or (2), and
- ☐ a $180 fee set forth in 37 CFR 1.17 (p)

(Filed after final action, a notice of allowance, on or before payment of the issue fee)

Please charge to Deposit Account **08-2025** the sum of $ 0.00 . At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account **08-2025** pursuant to 37 CFR 1.25.

Applicant(s) submit herewith form PTO/SB/08B - Information Disclosure Statement together with any required copies of patents, publications or other information of which applicant(s) are aware, which applicant(s) believe(s) may be material to the examination of this application and for which there may be a duty to disclose in accordance with 37 CFR 1.56.

☐ A concise explanation of the relevance of foreign language patents, foreign language publications and other foreign language information listed on PTO/SB/08B, as presently understood by the individual(s) designated in 37 CFR 1.56 (c) most knowledgeable about the content is given on the attached sheet, or where a foreign language patent is cited in a search report or other action by a foreign patent office in a counterpart foreign application, an english language version of the search report or action which indicates the degree of relevance found by the foreign office is listed on form PTO/SB/08B and is enclosed.

It is requested that the information disclosed herein be made of record in this application.

Respectfully submitted,
Scott L. Michaelis

By: /Robert W. Nelson/
Robert W. Nelson, Esq.
Attorney/Agent for Applicant(s)
Reg No. : 37,898
Date : May 28, 2009
Telephone : (303) 298-9888

Rev 10/08 (S-IDS)

PTO/SB/08b (03-09)
Approved for use through 04/30/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	10/972,588
Filing Date	10/25/2004
First Named Inventor	Scott L. Michaelis
Art Unit	2114
Examiner Name	LE, DIEU MINH T.
Attorney Docket Number	200404973-1

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Notice of Allowance dated 2-11-2009 for Serial Number 10/973,077 in the United States	
		Notice of Allowance dated 9-23-2008 for Serial Number 10/973,077 in the United States	
		Office Action dated 3-19-2008 for Serial Number 10/973,077 in the United States	
		Office Action dated 9-20-2007 for Serial Number 10/973,077 in the United States	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

Electronic Patent Application Fee Transmittal	
Application Number:	10972588
Filing Date:	25-Oct-2004
Title of Invention:	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
First Named Inventor/Applicant Name:	Scott L. Michaelis
Filer:	Robert Wayne Nelson/Joy Reinhart
Attorney Docket Number:	200404973-1
Filed as Large Entity	

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Request for continued examination	1801	1	810	810
		Total in USD ($)		**810**

Electronic Acknowledgement Receipt	
EFS ID:	5321771
Application Number:	10972588
International Application Number:	
Confirmation Number:	1079
Title of Invention:	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
First Named Inventor/Applicant Name:	Scott L. Michaelis
Customer Number:	22879
Filer:	Robert Wayne Nelson/Joy Reinhart
Filer Authorized By:	Robert Wayne Nelson
Attorney Docket Number:	200404973-1
Receipt Date:	12-MAY-2009
Filing Date:	25-OCT-2004
Time Stamp:	17:06:16
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Continued Examination (RCE)	RCE.pdf	94300 e53924fecc3809f2ef1392338f8ea9f7227164538	no	2

Warnings:

This is not a USPTO supplied RCE SB30 form.

Information:					
2	Information Disclosure Statement (IDS) Filed (SB/08)	sb0008b.pdf	221151 [redacted]	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	10973077_NOA_2-11-2009.pdf	2036476 [redacted]	no	33
Warnings:					
Information:					
4	NPL Documents	10973077_NOA_9-23-2008.pdf	1017594 [redacted]	no	20
Warnings:					
Information:					
5	NPL Documents	10973077_OA_3-19-2008.pdf	839166 [redacted]	no	25
Warnings:					
Information:					
6	NPL Documents	10973077_OA_9-20-2007.pdf	966565 [redacted]	no	30
Warnings:					
Information:					
7	Fee Worksheet (PTO-875)	fee-info.pdf	30791 [redacted]	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	5206043		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200404973-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Scott L. Michaelis	Confirmation No.: 1079
Application No.: 10/972,588	Examiner: Le, Dieu Minh T.
Filing Date: 10/25/2004	Group Art Unit: 2114

Title: SYSTEM AND METHOD FOR SWITCHING THE ROLE BOOT PROCESSOR TO A SPARE PROCESSOR RESPONSIVE TO DETECTION OF LOSS OF LOCKSTEP IN A BOOT PROCESSOR

Mail Stop RCE
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

REQUEST FOR CONTINUED EXAMINATION (RCE) 37 CFR 1.114

Subsection (b) of 35 U.S.C. 132, effective on May 29, 2000, provides for continued examination of a utility or plant application filed on or after June 8, 1995.
See The American Inventors Protection Act of 1999 (AIPA).

This is a Request for Continued Examination (RCE) under CFR 1.114 of the above-identified application.

NOTE: 37 CFR 1.114 is effective on May 29, 2000. If the above- application was filed prior to May 29, 2000, applicant may wish to consider filing a continued prosecution application (CPA) under CFR 1.53(d) (PTO/SB/29) instead of a RCE to be eligible for the patent term adjustment provisions of the AIPA. See Changes to Application Examination and Provisional Application Practice, Interim Rule, 65 Fed. Reg. 14865 (Mar. 20, 2000), 1233 off. Gaz. Pat. Office

Submission under 37 CFR 1.114

☐ Previously submitted

- ☐ Consider the amendment(s)/reply under 37 CFR 1.116 previously filed on _____
  (Any unentered amendment(s) referred to above will be entered).
- ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____
- ☐ Other _____

☒ Enclosed

- ☐ Amendment/Reply
- ☐ Affidavit(s)/Declaration(s)
- ☒ Information Disclosure Statement (IDS)
- ☐ Other _____

Miscellaneous

- ☐ Suspension of action is requested under 37 CFR 1.103(c) for a period of _____ months.
  The fee for this Suspension is (37 CFR 1.17(i)) $130.00
- ☐ Other _____

CONTINUED EXAMINATION TRANSMITTAL (RCE) (37 CFR 1.114) (continued)	PATENT APPLICATION ATTORNEY DOCKET NO. 200404973-1

☒ RCE filing fee $810.00

☐ A Petition for Extension of Time

☐ 1st Month $130 ☐ 2nd Month $490 ☐ 3rd Month $1110 ☐ 4th Month $1730

☒ Please charge to Deposit Account 08-2025 the sum of $ 810 . At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

Respectfully submitted,

Scott L. Michaelis

By: /Robert W. Nelson/

Robert W. Nelson, Esq.

Attorney/Agent for Applicant(s)

Reg No. : 37,898

Date : May 12, 2009

Telephone : (303) 298-9888

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 08/03/2009

ABOARDLE SALE #00000046 Mailroom Dt: 05/12/2009 082025 10972588
01 FC : 1801 810.00 DA

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

22879 7590 02/12/2009

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LE, DIEU MINH T	
ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 02/12/2009

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/972,588	10/25/2004	Scott L. Michaelis	200404973-1	1079

TITLE OF INVENTION: 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

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	05/12/2009

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

22879 7590 02/12/2009

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/972,588	10/25/2004	Scott L. Michaelis	200404973-1	1079

TITLE OF INVENTION: 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

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	05/12/2009

EXAMINER	ART UNIT	CLASS-SUBCLASS
LE, DIEU MINH T	2114	714-012000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ☐ Change of correspondence address (or Change of Correspondence Address Form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 .....
2 .....
3 .....

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
- ☐ Issue Fee
- ☐ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies .....

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number ..... (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ..... Date .....

Typed or printed name ..... Registration No. .....

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/972,588	10/25/2004	Scott L. Michaelis	200404973-1	1079

22879 7590 02/12/2009

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LE, DIEU MINH T	
ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 02/12/2009

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 529 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 529 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

Notice of Allowability	Application No.	Applicant(s)	
	10/972,588	MICHAELIS ET AL.	
	Examiner	Art Unit	
	Dieu-Minh Le	2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *the communication filed on 12/09/2008*.

2. ☒ The allowed claim(s) is/are *1-12, and 16-30 [now as 1-27]*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 12/09/2008
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/Dieu-Minh Le/
Primary Examiner, Art Unit 2114

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10972588	MICHAELIS ET AL.
	Examiner	Art Unit
	Dieu-Minh Le	2114

ORIGINAL		INTERNATIONAL CLASSIFICATION									
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED				
714	12	G	0	6	F	11 / 03 (2006.01.01)					

CROSS REFERENCE(S)					
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
714	11				

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	14	17												
2	2	15	18												
3	3	16	19												
4	4	17	20												
5	5	18	21												
6	6	19	22												
7	7	20	23												
8	8	21	24												
9	9	22	25												
10	10	23	26												
11	11	24	27												
12	12	25	28												
	13	26	29												
	14	27	30												
	15														
13	16														

NONE		Total Claims Allowed:	
(Assistant Examiner)	(Date)	27	
/Dieu-Minh Le/ Primary Examiner.Art Unit 2114	2/1/2009	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	1

U.S. Patent and Trademark Office — Part of Paper No. 20090201

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10972588	MICHAELIS ET AL.
	Examiner	Art Unit
	Dieu-Minh Le	2114

SEARCHED			
Class	Subclass	Date	Examiner
714	10, 11, 12, 13	1/31/09	dml
712	31	1/31/09	dml

SEARCH NOTES		
Search Notes	Date	Examiner
EAST search all files updated	1/31/09	dml
Inventor name search	1/31/09	dml
NPL search files [IEEE, Google Scholar]	2/1/09	dml

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
714	10-13	1/31/09	dml
712	31	1/31/09	dml
	See Interference search data file		

	/Dieu-Minh Le/ Primary Examiner.Art Unit 2114

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	34	("20020144177" | "20020152420" | "20030051190" | "20030070050" | "20030126498" | "20030135711" | "20030140281" | "20040006722" | "20040019771" | "20040078650" | "20040078651" | "20040153857" | "20050172164" | "20050240806" | "20050240811" | "20050240829" | "20060085677" | "20060248384" | "20060248684" | "4358823" | "5537655" | "5675794" | "5915082" | "5991900" | "6220581" | "6263452" | "6438687" | "6625749" | "6687851" | "6754787" | "6920581" | "7085959" | "7225355" | "7251746").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2009/01/31 22:53
L2	2532	lockstep lock-step	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:15
L3	25197	(loss adj L2) LOL	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:15
L4	25667	multi-processor "multi processor" and L3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:15

L5	1817	(boot$4 or initializ$4) adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:30
L6	302	spare adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:30
L7	20	L5 and L6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:30
L8	18	4 and 7	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:31
L9	169476	without same ("shut$4 down" stop$4 idle interrupt$4) same (OS "operating system")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:32
L10	7	L9 and 8	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:32
L11	0	1 and 10	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:37
L12	30	(michaelis-scott$ or myer-sylvia$ or mchardy-william$).in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:38
L13	836	(714/11).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/01/31 23:38
L14	589	(714/10).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/01/31 23:38

L15	396	(714/12).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/01/31 23:38
L16	726	(714/13).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/01/31 23:38
L17	395	(714/31).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/01/31 23:38
L18	4	(L13 L14 L15 L16 L17) and L12	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:38
L19	2	(L13 L14 L15 L16 L17) and 10	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/01/31 23:38
L20	7	("714"/$).ccls. and 8 and 9	US-PGPUB; USPAT; USOCR	OR	ON	2009/01/31 23:43
L21	0	("711"/$).ccls. and 8 and 9	US-PGPUB; USPAT; USOCR	OR	ON	2009/01/31 23:44
L22	0	("709"/$).ccls. and 8 and 9	US-PGPUB; USPAT; USOCR	OR	ON	2009/01/31 23:44
S1	14	(("7085959") or ("20050172164") or ("5675794") or ("6438687") or ("6920581") or ("5915082") or ("6625749")).PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/26 17:54
S2	2	("7085959").PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/08/27 15:05
S3	730	(714/11).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/27 15:06
S4	337	(714/12).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 15:32

S5	606	(714/13).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/27 15:07
S6	36	("20020144177" | "20030070050" | "20040006722" | "20040019771" | "20040153857" | "5226152" | "5249188" | "5751932" | "5758058" | "5764660" | "6065135" | "6148348" | "6473869" | "6560682" | "6604177" | "6615366" | "6625749" | "6687851" | "6754787").PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/27 15:08
S7	44	("20020144177" | "20030070050" | "20040006722" | "20040019771" | "20040153857" | "6625749" | "6687851" | "6754787")".PN"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/27 15:09
S9	2222	lockstep lock-step	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 15:21
S11	25008	(loss adj S9) LOL	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 15:22
S12	51	(detect$3 or track$4) adj S11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 15:23
S13	1547	(boot$4 or initializ$4) adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:41

S14	246	spare adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 15:27
S15	12	S12 and S13 and S14	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 15:30
S16	25	(michaelis-scott$ or myer-sylvia$ or mchardy-william$).in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:14
S18	632	(714/10).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 15:33
S19	95	(712/31).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:08
S20	52536	(switch$4 chang$4 failover fail-over "fail over") same (spare slave redundan$4 backup back-up "back up" second$4) same processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:54
S21	81	S20 same S13	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:28
S22	168219	without same ("shut$4 down" stop$4 idle interrupt$4) same (OS "operating system")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:33

S23	6	S21 and S22	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:33
S24	20342	multi-processor "multi processor" and S11 and S13	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:45
S25	176	S24 and S20 and S22	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:46
S26	7	S25 and S14	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:53
S27	87	(recover$4 reestablish$3 restart$3) with S9	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:54
S28	31827	(pair spare slave redundan$4 backup back-up "back up" second$4) adj2 processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:55
S29	101834	firmware	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:56
S30	21	S27 and S28 and S29	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 16:57

S31	15153	trigger$3 same ("operating system" OS operating-system) same (idle stop$4 interrupt$3 halt$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 14:49
S32	207	(loss lose los$3 lost$3) same (lockstep lock-step "lock step")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 14:52
S33	86	(detect$3 track$3 monitor$3) same S32	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 14:56
S34	3	S33 same S31	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 14:56
S35	3	S32 same S31	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:02
S36	10	S32 and S31	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:02
S37	33339	(pair spare slave redundan$4 backup back-up "back up" second$4) adj2 processor	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:05
S38	2323	lockstep lock-step	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:06

S39	92	(recover$4 reestablish$3 restart$3) with S38	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:06
S40	114677	firmware	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:06
S41	11	S37 same S39 same S40	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:06
S42	764	(714/11).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:07
S43	352	(714/12).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:07
S44	659	(714/13).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:07
S45	658	(714/10).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:07
S46	96	(712/31).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:07

S47	2	(S42 S43 S44 S45 S46) and S31 and S33	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:07
S48	2	(S42 S43 S44 S45 S46) and S31 and S32 and S37 and S40	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/14 15:08
S49	804	(714/11).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
S50	565	(714/10).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
S51	371	(714/12).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
S52	696	(714/13).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
S53	381	(714/31).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
S54	204	(loss lose los$3 lost$3) same (lockstep lock-step "lock step")	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:21
S55	1586	(boot$4 or initializ$4) adj processor	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:22
S56	52623	(switch$4 chang$4 failover fail-over "fail over") same (spare slave redundan$4 backup back-up "back up" second$4) same processor	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:22
S57	30192	(pair spare slave redundan$4 backup back-up "back up" second$4) adj2 processor	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:22
S58	82	(detect$3 track$3 monitor$3) same S54	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:23
S59	163830	without same ("shut$4 down" stop$4 idle interrupt$4) same (OS "operating system")	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:24
S60	32	S58 and S56 and S57	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:29
S61	7	S60 and S49	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:29

S62	3	S60 and S50 and S59	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:30
S63	7	S60 and (S51 S52 S53) and S59	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:31
S64	22	S60 and ("714"/$.ccls.) and S59	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:38
S65	12	(switch$4 chang$4) same role same S55	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:39
S66	7	S64 and S65	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:40

1/31/09 11:48:52 PM
C:\Documents and Settings\MLe\My Documents\EAST\Workspaces\10972588rce.wsp

IEEE Xplore® RELEASE 2.6

Home | Login | Logout | Access Information | Alerts | Purchase History | My Past | Settings | Help

Welcome United States Patent and Trademark Office

Celebrating 125 Years of Engineering the Future | IEEE

BROWSE | SEARCH | IEEE Xplore GUIDE | SUPPORT

Search Session History

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Sun, 1 Feb 2009, 2:32:38 PM EST

Search Query Display

Run Search | Reset

Recent Search Queries		Results
#1	(((spare processor)<in>metadata) <and> ((system identifier) <in>metadata))<and> ((shutting down os)<in>metadata)	0
#2	(((switching changing)<in>metadata) <and> ((boot processor) <in>metadata))<and> ((spare processor)<in>metadata)	0
#3	(((detecting tracking monitoring)<in>metadata) <and> (((loss of lockstep) lol)<in>metadata))<and> ((boot processor) <in>metadata)	0
#4	((((loss of lockstep) lol)<in>metadata) <and> ((boot processor) <in>metadata))<and> ((system identifier)<in>metadata)	0
#5	(((system firmware)<in>metadata) <and> ((lol)<in>metadata)) <and> ((error detection logic)<in>metadata)	0

Clear Session History

Help Contact Us Privacy & Security IEEE.org

© Copyright 2009 IEEE – All Rights Reserved

Indexed by Inspec®

Web Images Maps News Shopping Gmail more ▼

Google

detect (loss of lockstep) (switch recover) (boot processor) (spare processor) (wit Search Advanced Search Preferences

Sign in

Web Results 1 - 10 of about 290 for detect (loss of lockstep) (switch recover) (boot processor) (spare processor) (without) interrup OS. (0.14 seconds)

1. (WO/1998/055922) LOOSELY-COUPLED, SYNCHRONIZED EXECUTION

The **processor** services the maskable **interrupt** some number of instructions .... TS already was set prior to the task **switch** into the **lockstep** control task. ...
www.wipo.int/pctdb/en/wo.jsp?wo=1998055922&IA=US1998011423&DISPLAY=DESC - 93k - Cached - Similar pages

2. EIDE flaw : Java Glossary

Because **OS**/2 has already signalled EOI (End Of **Interrupt**) to the PIC (Programmable .... Support line at 1-800-628-8686 forhelp with your **processor** issues. ...
mindprod.com/jgloss/eideflaw.html - 62k - Cached - Similar pages

3. [PDF] Manual Cover.fm

File Format: PDF/Adobe Acrobat - View as HTML
interface and does not use any legacy I/O or **interrupt** resources. ..... temperature sensor, and a fast acting TCC, the **processor**, **without** ...
www.powerbridge.de/download/manual/cpci/cpci-cpu/CPCI-6910_UM_0705.pdf - Similar pages

4. [PDF] Software Optimization Guide for the AMD Hammer **Processor**

File Format: PDF/Adobe Acrobat
AMD-V processors automatically deliver a pending virtual **interrupt** to the guest when ..... The **processor** continues to combine writes to this buffer **without** ...
www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/40546.pdf - Similar pages

5. Architecture and implementation for a high reliability long-term ...

Another important parameter, the **processor** throughput, was ... **detection** of the CPU is to have two CPUs operate in **lock-step** ...
ieeexplore.ieee.org/iel2/1033/6953/00282118.pdf?arnumber=282118 - Similar pages
by D Gomez - 1992

6. [PDF] ATCA-4300/ATCA-4310 Compute Processing Module Reference

File Format: PDF/Adobe Acrobat
OS boot failure. Failure to **boot** from a PXE server or **detection** of a bad **boot** ..... The **switch** from PIC mode to **Processor** System Bus **interrupt** ...
www.radisys.com/files/support_downloads/02237-03_ATCA-43xx_CPM_Reference.pdf - Similar pages

7. [PDF] "EP000986784B1"

File Format: PDF/Adobe Acrobat - View as HTML
and directed to the **processor** through a specific **interrupt**. vector. ..... the task **switch** into the **lockstep** control task. If it was, ...
https://publications.european-patent-office.org/PublicationServer/document.pdf.jsp?iDocId=5130155&iebug=.pdf - Similar pages

8. [PDF] Fingerprinting: Hash-Based Error **Detection** in Microprocessors

File Format: PDF/Adobe Acrobat
In each run, the **processor boots** and starts the test program. ..... **recovery** and **OS**-transparent redundant execution **without lockstep**. ...
www.ece.cmu.edu/~jhoe/distribution/2008/jsmolens.pdf - Similar pages

9. [PDF] Intel 5000 Series Chipsets Server Board Family Datasheet

File Format: PDF/Adobe Acrobat
Setup Utility — Specific **Processor** Information Screen Display ..... The system can **boot without** a keyboard or mouse attached. If present, the BIOS will ...
www.cs.earlham.edu/~charliep/d38960004-s5000.pdf - Similar pages

10. [PS] Synthesis: An Ecient Implemeniation of Fundamental Operating ...

File Format: Adobe PostScript - View as HTML
symmetric multiprocessor, scheduling tasks on either **processor without** ..... An

interrupt causing a context **switch** will execute the current program's ...
www.cs.columbia.edu/~library/TR-repository/reports/reports-1992/cucs-039-92.ps.gz
- Similar pages
by H Massalin - 1992 - Cited by 131 - Related articles

Goooooooooogle
1 2 3 4 5 6 7 8 9 10 Next

detect (loss of lockstep) (switch recover) (boot processor) (spare processor) (wit Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve | Try Google Experimental

Google Home - Advertising Programs - Business Solutions - Privacy - About Google

PTO/SB/08a (09-08)
Approved for use through 10/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 2

Complete if Known	
Application Number	~~10/972,588~~ 10/972,588
Filing Date	October 25, 2004
First Named Inventor	Scott L. Michaelis
Art Unit	2114
Examiner Name	Le, Dieu Minh T.
Attorney Docket Number	200404973-1

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/M.L./		US- 2002/0152420	10-17-2002	Chaudhry et al.	
		US- 2003/0051190	03-13-2003	Marisetty et al.	
		US- 2003/0135711	07-17-2003	Shoemaker et al.	
		US- 2004/0078651	04-22-2004	Stafford et al.	
		US- 2005/0240811	10-27-2005	Stafford et al.	
		US- 2006/0248684	11-02-2006	Stafford	
		US- 2006/0085677	04-20-2006	Stafford et al.	
		US- 2006/0248384	11-9-2006	Vosbikian et al.	
		US- 4,358,823	11-9-1982	McDonald et al.	
		US- 5,537,655	07-16-1996	Truong	
		US- 5,675,794	10-07-1997	Meredith	
		US- 5,991,900	11-23-1999	Garnett	
		US- 6,220,581	04-24-2001	Mueller	
		US- 6,263,452	07-17-2001	Jewett et al.	
		US- 6,438,687	08-20-2002	Klein	
		US- 7,225,355	05-29-2007	Yamazaki et al.	
		US- 2003/0126498	07-03-2003	Bigbee et al.	
		US- 2004/0078650	04-22-2004	Stafford et al.	
		US- 2005/0240806	10-27-2005	Bruckert et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] Number[4] Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]

Examiner Signature	/Dieu Minh Le/	Date Considered	01/31/2009

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

PTO/SB/08a (11-06)
Approved for use through 12/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 2 of 2

Complete if Known	
Application Number	10/972,588
Filing Date	October 25, 2004
First Named Inventor	Scott L. Michaelis
Art Unit	2114
Examiner Name	Le, Dieu Minh T.
Attorney Docket Number	200404973-1

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/M.L../		US- 2006/0248684	11-09-2006	Vosbikian et al.	
		US- 5,896,523	04-20-1999	Bissett et al.	
		US- 6,141,770	10-31-2000	Fuchs et al.	
		US- 6,675,324	01-06-2004	Marisetty et al.	
		US- 6,948,092	09-20-2005	Kondo et al.	
		US- 7,003,691	02-21-2006	Stafford et al.	
		US- 7,020,800	03-28-2006	Fu et al.	
		US- 7,134,047	11-07-2006	Quach	
		US- 7,155,721	12-26-2006	Stafford et al.	
		US- 7,237,144	06-26-2007	Stafford et al.	
		US- 7,191,292	03-13-2007	Chaudhry et al.	
		US- 2006/0136672	06-22-2006	Chaudhry et al.	
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] Number[4] Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]

Examiner Signature	/Dieu Minh Le/	Date Considered	01/31/2009

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10972588	MICHAELIS ET AL.
	Examiner	Art Unit
	Dieu-Minh Le	2114

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	02/28/2008	09/11/2008	02/01/2009						
1	1	✓	=	=						
2	2	✓	=	=						
3	3	✓	=	=						
4	4	✓	=	=						
5	5	✓	=	=						
6	6	✓	=	=						
7	7	✓	=	=						
8	8	✓	=	=						
9	9	✓	=	=						
10	10	✓	=	=						
11	11	✓	=	=						
12	12	✓	=	=						
	13	✓	-	-						
	14	✓	-	-						
	15	✓	-	-						
13	16	✓	=	=						
14	17	✓	=	=						
15	18	✓	=	=						
16	19	✓	=	=						
17	20	✓	=	=						
18	21	✓	=	=						
19	22	✓	=	=						
20	23	✓	=	=						
21	24	=	=	=						
22	25	=	=	=						
23	26	=	=	=						
24	27	=	=	=						
25	28	=	=	=						
26	29	✓	=	=						
27	30	✓	=	=						

PTO/SB/08a (09-08)
Approved for use through 10/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 2

Complete if Known	
Application Number	10/973,588
Filing Date	October 25, 2004
First Named Inventor	Scott L. Michaelis
Art Unit	2114
Examiner Name	Le, Dieu Minh T.
Attorney Docket Number	200404973-1

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 2002/0152420	10-17-2002	Chaudhry et al.	
		US- 2003/0051190	03-13-2003	Marisetty et al.	
		US- 2003/0135711	07-17-2003	Shoemaker et al.	
		US- 2004/0078651	04-22-2004	Stafford et al	
		US- 2005/0240811	10-27-2005	Stafford et al.	
		US- 2006/0248684	11-02-2006	Stafford	
		US- 2006/0085677	04-20-2006	Stafford et al.	
		US- 2006/0248384	11-9-2006	Vosbikian et al.	
		US- 4,358,823	11-9-1982	McDonald et al.	
		US- 5,537,655	07-16-1996	Truong	
		US- 5,675,794	10-07-1997	Meredith	
		US- 5,991,900	11-23-1999	Garnett	
		US- 6,220,581	04-24-2001	Mueller	
		US- 6,263,452	07-17-2001	Jewett et al.	
		US- 6,438,687	08-20-2002	Klein	
		US- 7,225,355	05-29-2007	Yamazaki et al.	
		US- 2003/0126498	07-03-2003	Bigbee et al.	
		US- 2004/0078650	04-22-2004	Stafford et al.	
		US- 2005/0240806	10-27-2005	Bruckert et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] Number[4] Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

PTO/SB/08a (11-08)
Approved for use through 12/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 2 of 2

Complete if Known	
Application Number	10/972,588
Filing Date	October 25, 2004
First Named Inventor	Scott L. Michaelis
Art Unit	2114
Examiner Name	Le, Dieu Minh T.
Attorney Docket Number	200404973-1

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 2006/0248684	11-09-2006	Vosbikian et al.	
		US- 5,896,523	04-20-1999	Bissett et al.	
		US- 6,141,770	10-31-2000	Fuchs et al.	
		US- 6,675,324	01-06-2004	Marisetty et al.	
		US- 6,948,092	09-20-2005	Kondo et al.	
		US- 7,003,691	02-21-2006	Stafford et al.	
		US- 7,020,800	03-28-2006	Fu et al.	
		US- 7,134,047	11-07-2006	Quach	
		US- 7,155,721	12-26-2006	Stafford et al.	
		US- 7,237,144	06-26-2007	Stafford et al.	
		US- 7,191,292	03-13-2007	Chaudhry et al.	
		US- 2006/0136672	06-22-2006	Chaudhry et al.	
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

Electronic Patent Application Fee Transmittal	
Application Number:	10972588
Filing Date:	25-Oct-2004
Title of Invention:	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
First Named Inventor/Applicant Name:	Scott L. Michaelis
Filer:	Robert Wayne Nelson/Kalyn Black
Attorney Docket Number:	200404973-1
Filed as Large Entity	

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Request for continued examination	1801	1	810	810
		Total in USD ($)		**810**

Electronic Acknowledgement Receipt	
EFS ID:	4419258
Application Number:	10972588
International Application Number:	
Confirmation Number:	1079
Title of Invention:	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
First Named Inventor/Applicant Name:	Scott L. Michaelis
Customer Number:	22879
Filer:	Robert Wayne Nelson/Kalyn Black
Filer Authorized By:	Robert Wayne Nelson
Attorney Docket Number:	200404973-1
Receipt Date:	09-DEC-2008
Filing Date:	25-OCT-2004
Time Stamp:	11:59:52
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Miscellaneous Incoming Letter	E-RCE.pdf	573164 253f3fcc8657f5f54c0e8842981c27b338ed7dc ea041	no	2
Warnings:					
Information:					

2	Information Disclosure Statement (IDS) Filed (SB/08)	200404973-1_IDS.pdf	114650 ec317c6c071983948433aa0398ae7dd828887	no	2
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Fee Worksheet (PTO-06)	fee-info.pdf	30795 5925721c38507d46e8baa84022e82dc17bb	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	718609		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200404973-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Scott L. Michaelis

Confirmation No.: 1079

Application No.: 10/972,588

Examiner: Le, Dieu Minh T.

Filing Date: October 25, 2004

Group Art Unit: 2114

Title: 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

Mail Stop RCE
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

REQUEST FOR CONTINUED EXAMINATION (RCE) 37 CFR 1.114

Subsection (b) of 35 U.S.C. 132, effective on May 29, 2000, provides for continued examination of a utility or plant application filed on or after June 8, 1995.
See The American Inventors Protection Act of 1999 (AIPA).

This is a Request for Continued Examination (RCE) under CFR 1.114 of the above-identified application.

NOTE: 37 CFR 1.114 is effective on May 20, 2000. If the above- application was filed prior to May 29, 2000, applicant may wish to consider filing a continued prosecution application (CPA) under CFR 1.53(d) (PTO/SB/29) instead of a RCE to be eligible for the patent term adjustment provisions of the AIPA. See Changes to Application Examination and Provisional Application Practice, Interim Rule, 65 Fed. Reg. 14865 (Mar. 20, 2000), 1233 off. Gaz. Pat. Office

Submission under 37 CFR 1.114

☐ Previously submitted

- ☐ Consider the amendment(s)/reply under 37 CFR 1.116 previously filed on _____.
  (Any unentered amendment(s) referred to above will be entered).
- ☐ Consider the arguments in the Appeal Brief or Reply Brief prevously filed on _____.
- ☐ Other _____

☒ Enclosed

- ☐ Amendment/Reply
- ☐ Affidavit(s)/Declaration(s)
- ☒ Information Disclosure Statement (IDS)
- ☐ Other _____

Miscellaneous

- ☐ Suspension of action is requested under 37 CFR 1.103(c) for a period of _____ months.
  The fee for this Suspension is (37 CFR 1.17(i)) $130.00
- ☐ Other _____

Rev 10/08 (E-RCE)

CONTINUED EXAMINATION TRANSMITTAL (RCE) (37 CFR 1.114) (continued)	PATENT APPLICATION
	ATTORNEY DOCKET NO. 200404973-1

☒ RCE filing fee $810.00

☐ A Petition for Extension of Time

☐ 1st Month $130 ☐ 2nd Month $490 ☐ 3rd Month $1110 ☐ 4th Month $1730

☒ Please charge to Deposit Account 08-2025 the sum of $ 810 . At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that my regulate fees.

Respectfully submitted,

Scott L. Michaelis

By: /Robert W. Nelson/

Robert W. Nelson, Esq.

Attorney/Agent for Applicant(s)

Reg No. : 37,898

Date : December 9, 2008

Telephone : (303) 298-9888

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 08/03/2009

ABOARDLE	SALE	#00000047	Mailroom Dt:	12/09/2008	082025	10972588	
		01 FC : 1801		810.00 DA			

PATENT APPLICATION
Sheet 1 of 3

FORM PTO-1449 LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT'S INFORMATION DISCLOSURE STATEMENT (Use several sheets if necessary)	ATTY. DOCKET NO. 200404973-1	APPLICATION NO. 10/972,588	CONFIRMATION NO.
	APPLICANT Scott L. Michaelis et al.		
	FILING DATE	GROUP 2114	

REFERENCE DESIGNATION **U.S. PATENT DOCUMENTS**

EXAMINER INITIAL		DOCUMENT NUMBER	PUBLICATION DATE	NAME	Pages, Columns, Lines Where Relevant Passages or Figures Appear
AML	1A	5,226,152	07/06/1993	Klug et al.	
	1B	5,249,188	09/28/1993	McDonald	
	1C	5,751,932	05/12/1998	Horst et al.	
	1D	5,758,058	05/26/1998	Millburn	
	1E	5,764,660	06/09/1998	Mohat	
	1F	6,065,135	05/16/2000	Marshall et al.	
	1G	6,148,348	11/14/2000	Garnett et al.	
	1H	6,473,869	10/29/2002	Bissett et al.	
	1I	6,560,682	05/06/2003	Miller et al.	
✓	1J	6,604,177	08/05/2003	Kondo et al.	
AML	1K	6,615,366	09/02/2003	Grochowski et al.	

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	PUBLICATION DATE	NAME OF PATENTEE OR APPLICANT	Pages/Columns/Lines Where Relevant Passages/Figures Appear	Check if Translation attached
	1L					
	1M					
	1N					
	1O					
	1P					

OTHER REFERENCES (including Author, Title, Date, Pertinent Pages, etc.)

AML	1Q	10/972,796 ~~Attorney Docket No. 200404974-1~~, filed concurrently herewith, entitled "SYSTEM AND METHOD FOR MAINTAINING IN A MULTI-PROCESSOR SYSTEM A SPARE PROCESSOR THAT IS IN LOCKSTEP FOR USE IN RECOVERING FROM LOSS OF LOCKSTEP FOR ANOTHER PROCESSOR"
AML	1R	10/973,807 ~~Attorney Docket No. 200404944-1~~, filed concurrently herewith, entitled "SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR"
AML	1S	10/972,835 ~~Attorney Docket No. 200404946-1~~, filed concurrently herewith, entitled "SYSTEM AND METHOD FOR USING INFORMATION RELATING TO A DETECTED LOSS OF LOCKSTEP FOR DETERMINING A RESPONSIVE ACTION"

AP P 11/12/07

EXAMINER	DATE CONSIDERED
David Vaughn	8/29/07

Rev 05/04 (PTO1449)

FORM PTO-1449	ATTY. DOCKET NO.	APPLICATION NO.	CONFIRMATION NO.
LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT'S INFORMATION DISCLOSURE STATEMENT	200404973-1	10/972,588	
	APPLICANT: Scott L. Michaelis et al.		
(Use several sheets if necessary)	FILING DATE	GROUP: 2114	

REFERENCE DESIGNATION **U.S. PATENT DOCUMENTS**

EXAMINER INITIAL		DOCUMENT NUMBER	PUBLICATION DATE	NAME	Pages, Columns, Lines Where Relevant Passages or Figures Appear
AML	2A	6,625,749	09/23/2003	Quach	
	2B	6,687,851	02/03/2004	Somers et al.	
	2C	6,754,787	06/22/2004	Miller et al.	
	2D	2002/0144177	10/03/2002	Kondo et al.	
	2E	2003/0070050	04/10/2003	Miller et al.	
	2F	2004/0006722	01/08/2004	Safford	
✓	2G	2004/0019771	01/29/2004	Quach	
AML	2H	2004/0153857	08/05/2004	Yamazaki et al.	
	2I				
	2J				
	2K				

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	PUBLICATION DATE	NAME OF PATENTEE OR APPLICANT	Pages/Columns/Lines Where Relevant Passages/Figures Appear	Check if Translation attached
	2L					
	2M					
	2N					
	2O					
	2P					

OTHER REFERENCES (including Author, Title, Date, Pertinent Pages, etc.)

PRP 11/12/01

AML	2Q	10/973,076 ~~Attorney Docket No. 200404941-1~~, filed concurrently herewith, entitled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION"
AML	2R	10/973,084 ~~Attorney Docket No. 200404943-1~~, filed concurrently herewith, entitled "SYSTEM AND METHOD FOR CONFIGURING LOCKSTEP MODE OF A PROCESSOR MODULE"
AML	2S	10/973,083 ~~Attorney Docket No. 200404971-1~~, filed concurrently herewith, entitled "SYSTEM AND METHOD FOR REESTABLISHING LOCKSTEP FOR A PROCESSOR MODULE FOR WHICH LOSS OF LOCKSTEP IS DETECTED"

EXAMINER	DATE CONSIDERED
David Huald	8/29/07

Rev 05/04 (PTO1449)

FORM PTO-1449 LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT'S INFORMATION DISCLOSURE STATEMENT (Use several sheets if necessary)	ATTY. DOCKET NO. 200404973-1	APPLICATION NO. 10/972,588	CONFIRMATION NO.
	APPLICANT Scott L. Michaelis et al.		
	FILING DATE	GROUP 2114	

REFERENCE DESIGNATION **U.S. PATENT DOCUMENTS**

EXAMINER INITIAL		DOCUMENT NUMBER	PUBLICATION DATE	NAME	Pages, Columns, Lines Where Relevant Passages or Figures Appear
	3A				
	3B				
	3C				
	3D				
	3E				
	3F				
	3G				
	3H				
	3I				
	3J				
	3K				

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	PUBLICATION DATE	NAME OF PATENTEE OR APPLICANT	Pages/Columns/Lines Where Relevant Passages/Figures Appear	Check if Translation attached
	3L					
	3M					
	3N					
	3O					
	3P					

OTHER REFERENCES (including Author, Title, Date, Pertinent Pages, etc.)

RAM	3Q	Attorney Docket No. 200404970-1, filed concurrently herewith, entitled "SYSTEM AND METHOD FOR SYSTEM FIRMWARE CAUSING AN OPERATING SYSTEM TO IDLE A PROCESSOR" 10/972,888
RAM	3R	Attorney Docket No. 200404972-1, filed concurrently herewith, entitled "SYSTEM AND METHOD FOR REINTRODUCING A PROCESSOR MODULE TO AN OPERATING SYSTEM AFTER LOCKSTEP RECOVERY" 10/972,075
	3S	

RAM 11/12/08

EXAMINER	DATE CONSIDERED
Brad Hale	8/29/07

Rev 05/04 (PTO1449)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

22879 7590 09/23/2008

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LE, DIEU MINH T	
ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 09/23/2008

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/972,588	10/25/2004	Scott L. Michaelis	200404973-1	1079

TITLE OF INVENTION: 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

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1440	$300	$0	$1740	12/23/2008

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

22879 7590 09/23/2008

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/972,588	10/25/2004	Scott L. Michaelis	200404973-1	1079

TITLE OF INVENTION: 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

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1440	$300	$0	$1740	12/23/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
LE, DIEU MINH T	2114	714-012000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
- ☐ Change of correspondence address (or Change of Correspondence Address Form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 .....
2 .....
3 .....

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
- ☐ Issue Fee
- ☐ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies .....

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number ..... (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ..... Date .....

Typed or printed name ..... Registration No. .....

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/972,588	10/25/2004	Scott L. Michaelis	200404973-1	1079

22879 7590 09/23/2008

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LE, DIEU MINH T	
ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 09/23/2008

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 529 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 529 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

Notice of Allowability	Application No.	Applicant(s)	
	10/972,588	MICHAELIS ET AL.	
	Examiner	Art Unit	
	Dieu-Minh Le	2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *the interview on 9/5/08 and communications filed 6/5/08 and 9/10/08*.

2. ☒ The allowed claim(s) is/are *1-12 and 16-30 now as [1-27]*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☒ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/Dieu-Minh Le/
Primary Examiner, Art Unit 2114

Interview Summary	Application No.	Applicant(s)	
	10/972,588	MICHAELIS ET AL.	
	Examiner	Art Unit	
	Dieu-Minh Le	2114	

All participants (applicant, applicant's representative, PTO personnel):

(1) *Dieu-Minh Le*. (3)_____.

(2) *Augustus Winfield [Reg. No. 34,046]*. (4)_____.

Date of Interview: *05 September 2008*.

Type: a)☒ Telephonic b)☐ Video Conference
c)☐ Personal [copy given to: 1)☐ applicant 2)☐ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes e)☒ No.
If Yes, brief description: _____.

Claim(s) discussed: *1-30*.

Identification of prior art discussed: *Safford, Fox and Quach*.

Agreement with respect to the claims f)☒ was reached. g)☐ was not reached. h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Main issues discussed were the claimed invention over the above prior arts, double patenting rejection, 35 USC 101 for claims 29-30 and the proposed amended claims 1-30. Applicant's representative agreed to file TD and supplemental amendment according to communications filed 9/10/2008*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

/Dieu-Minh Le/
Primary Examiner, Art Unit 2114

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10972588	MICHAELIS ET AL.
	Examiner	Art Unit
	Dieu-Minh Le	2114

SEARCHED			
Class	Subclass	Date	Examiner
714	10, 11, 12, 13	9/4/08	dml
712	31	9/4/08	dml

SEARCH NOTES		
Search Notes	Date	Examiner
EAST search all files updated	9/4/08	dml
Inventor name search	9/4/08	dml
NPL search files [IEEE, Google Scholar]	9/11/08	dml

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
714	10-13	9/4/08	dml
712	31	9/4/08	dml
	See Interference search data file		

	/Dieu-Minh Le/ Primary Examiner.Art Unit 2114

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 1079

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/972,588	10/25/2004	713	2114	200404973-1

APPLICANTS
Scott L. Michaelis, Plano, TX;
Anurupa Rajkumari, Round Rock, TX;
William B. McHardy, Fort Collins, CO;

** CONTINUING DATA ********** na dml

** FOREIGN APPLICATIONS **********

** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** na dml
01/14/2005

Foreign Priority claimed ☐ Yes ☑ No 35 USC 119(a-d) conditions met ☐ Yes ☑ No Verified and Acknowledged /DIEU-MINH T LE/ Examiner's Signature	☐ Met after Allowance dml Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		TX	4	30 27	4

ADDRESS
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400
UNITED STATES

TITLE
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

FILING FEE RECEIVED		
1058	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other ☐ Credit

BIB (Rev. 05/07).

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	804	(714/11).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
L2	565	(714/10).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
L3	371	(714/12).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
L4	696	(714/13).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
L5	381	(714/31).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:19
L6	204	(loss lose los $3 lost$3) same (lockstep lock-step "lock step")	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:21
L7	1586	(boot$4 or initializ$4) adj processor	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:22
L8	52623	(switch$4 chang$4 failover fail-over "fail over") same (spare slave redundan$4 backup back-up "back up" second$4) same processor	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:22

L9	30192	(pair spare slave redundan $4 backup back-up "back up" second$4) adj2 processor	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:22
L10	82	(detect$3 track $3 monitor$3) same 6	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:23
L11	163830	without same ("shut$4 down" stop$4 idle interrupt $4) same (OS "operating system")	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:24
L12	32	10 and 8 and 9	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:29
L13	7	12 and 1	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:29
L14	3	12 and 2 and 11	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:30
L15	7	12 and (3 4 5) and 11	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:31
L16	22	12 and ("714"/ $.ccls.) and 11	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:38
L17	12	(switch$4 chang$4) same role same 7	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:39
L18	7	16 and 17	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/11 12:40

9/ 11/ 08 12:54:58 PM
C:\ Documents and Settings\ MLe\ My Documents\ EAST\ Workspaces\ 10972588i. wsp

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	40917	multi-processor "multi processor" (plural$3 adj processor)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 12:32
L2	1742	(boot$4 or initializ$4) adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 12:32
L3	94	L2 same 1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 12:32
L4	16	(assign$4 giv$3 order$3 decid$3 mak$3) same role same 2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 12:35
L5	2459	lockstep lock-step	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:00
L6	25133	(loss adj L5) LOL	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:00
L7	56	(detect$3 or track$4) adj L6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:00

L8	13	3 same L7	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:00
L9	0	8 not 4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
L10	803	(714/11).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
L11	370	(714/12).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
L12	693	(714/13).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
L13	682	(714/10).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
L14	104	(712/31).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
L15	2237	L10 L11 L12 L13 L14	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01

L16	803	(714/11).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
L17	289	spare adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
L18	15	L7 and L2 and L17	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
L19	221	(loss lose los$3 lost$3) same (lockstep lock-step "lock step")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
L20	92	(detect$3 track$3 monitor$3) same L19	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
L21	3	L20 and L16 and L18	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
L22	803	(714/11).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/04 13:46
L23	370	(714/12).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/04 13:46
L24	693	(714/13).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/04 13:46
L25	563	(714/10).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/04 13:46
L26	15199	without same (shutt$3 turn-off power-off) same ((operating adj system) OS)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 14:09

L27	9	4 and 26	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 14:10
L28	0	27 not 8	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 14:12

9/4/08 2:14:58 PM
C:\Documents and Settings\MLe\My Documents\EAST\Workspaces\10972588b.wsp

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10972588	MICHAELIS ET AL.
	Examiner	Art Unit
	Dieu-Minh Le	2114

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	02/28/2008	09/11/2008							
1	1	✓	=							
2	2	✓	=							
3	3	✓	=							
4	4	✓	=							
5	5	✓	=							
6	6	✓	=							
7	7	✓	=							
8	8	✓	=							
9	9	✓	=							
10	10	✓	=							
11	11	✓	=							
12	12	✓	=							
	13	✓	-							
	14	✓	-							
	15	✓	-							
13	16	✓	=							
14	17	✓	=							
15	18	✓	=							
16	19	✓	=							
17	20	✓	=							
18	21	✓	=							
19	22	✓	=							
20	23	✓	=							
21	24	=	=							
22	25	=	=							
23	26	=	=							
24	27	=	=							
25	28	=	=							
26	29	✓	=							
27	30	✓	=							

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10972588	MICHAELIS ET AL.
	Examiner	Art Unit
	Dieu-Minh Le	2114

ORIGINAL		INTERNATIONAL CLASSIFICATION									
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED				
714	12	G	0	6	F	11 / 03 (2006.01.01)					

CROSS REFERENCE(S)					
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
714	11				

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	14	17												
2	2	15	18												
3	3	16	19												
4	4	17	20												
5	5	18	21												
6	6	19	22												
7	7	20	23												
8	8	21	24												
9	9	22	25												
10	10	23	26												
11	11	24	27												
12	12	25	28												
	13	26	29												
	14	27	30												
	15														
13	16														

NONE	Total Claims Allowed:	
(Assistant Examiner) (Date)	27	
/Dieu-Minh Le/ Primary Examiner.Art Unit 2114 09/12/2008	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner) (Date)	1	1

U.S. Patent and Trademark Office Part of Paper No. 20080911

Application Number	Application/Control No.	Applicant(s)/Patent under Reexamination	
	10/972,588	MICHAELIS ET AL.	
		Le	2114

Document Code - DISQ	Internal Document – DO NOT MAIL

TERMINAL DISCLAIMER	☒ APPROVED 2 TD's	☐ DISAPPROVED
Date Filed : 9.10.08	This patent is subject to a Terminal Disclaimer	

Approved/Disapproved by: Anyne

U.S. Patent and Trademark Office

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.: 10/972,588 EXAMINER: Le, Dieu Minh T

FILED: 10/25/2004 ART UNIT: 2114

INVENTOR: Scott L. Michaelis *et al.* CONFIRMATION NO.: 1079

ATTORNEY DOCKET NUMBER 200404973-1

AMENDMENT IN RESPONSE TO PHONE INTERVIEW ON 09/05/2008

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Amendments to the claims are reflected in the listing of claims beginning on page 2 of this paper.

Remarks begin on page 7 of this paper.

1. (currently amended) A method comprising:

detecting loss of lockstep (LOL) for a processor in a multi-processor system;

determining that the processor for which said LOL is detected is assigned a role of boot processor; ~~and~~

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

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.

2. (original) 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.

3. (original) The method of claim 1 wherein the spare processor is an active spare.

4. (original) The method of claim 3 wherein said switching further comprises:

causing the system's operating system to idle the active spare.

5. (original) The method of claim 1 wherein the spare processor is a processor module that includes a master and slave processors that operate in lockstep.

6. (original) 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.

7. (original) The method of claim 1 further comprising:

determining the processor that is assigned the role of spare processor.

8. (original) 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.

9. (original) The method of claim 1 wherein said determining step is performed by system firmware.

10. (original) The method of claim 1 wherein said switching step is performed by system firmware.

11. (original) 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.

12. (original) 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.

13. (canceled)

14. (canceled)

15. (canceled)

16. (currently amended) A method comprising:

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;

determining one of said lockstep pair of processors that is not the cause of said LOL;

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; and

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 lockstep pair of processors for which said LOL is detected as the boot processor; and

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.

17. (original) The method of claim 16 wherein the spare processor is a hot spare that is held in idle until switched the role of boot processor.

18. (original) The method of claim 16 wherein the spare processor is an active spare.

19. (original) The method of claim 18 further comprising:

causing the system's operating system to idle the active spare.

20. (original) The method of claim 16 wherein said switching causes the system's operating system to recognize said spare processor as said boot processor without shutting down said operating system.

21. (original) The method of claim 16 wherein said determining step is performed by system firmware.

22. (original) The method of claim 16 wherein said copying step is performed by system firmware.

23. (original) The method of claim 16 wherein said switching step is performed by system firmware.

24. (original) A system comprising:

a plurality of processor modules that each include a master processor and a slave processor that operate in lockstep;

an operating system;

error detection logic operable to detect loss of lockstep (LOL) for at least one of said processor modules; and

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;

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.

25. (original) The system of claim 24 wherein the another processor module is a processor module assigned a role of spare processor.

26. (original) The system of claim 24 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.

27. (original) The system of claim 24 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.

28. (original) The system of claim 24 further comprising:

non-volatile data storage storing information that assigns the role of system boot processor to one of said plurality of processor modules.

29. (currently amended) A system comprising:

system firmware means for detecting loss of lockstep (LOL) for a processor in a multi-processor system;

system firmware means for determining whether the processor for which said

LOL is detected is assigned the role of boot processor; and

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.

30. (original) The system of claim 29 wherein said processor in said multi-processor system is an Itanium Processor Family (IPF) processor.

REMARKS

Claims 1, 16, and 29 have been amended. Claims 13-15 have been canceled. Claims 1-12 and 16-30 remain in the application. Reexamination and reconsideration is respectfully requested.

Double Patenting Rejections

Claims 1-30 have been rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-32 of commonly owned U.S. application 10/973,077 and claims 1-26 of commonly owned U.S. application 10/972,796. Note that U.S. application 10/972,796 has issued as U.S. Patent Number 7,366,948. Terminal Disclaimers have been submitted separately.

Rejections Under 35 U.S.C. § 102

Claims 1-4, 7, 9-23, and 29 have been rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by U.S. Patent 7,085,959 (Safford). In a phone interview on September 5, 2008, the examiner and the below signed attorney discussed amendments that might make the claims allowable. In response to that interview, claim 1 has been amended to incorporate the limitations of claims 13-15 and claims 13-15 have been canceled. Claim 16 has been amended to incorporate similar limitations as claim 1. In addition, claim 29 has been amended to add system firmware as additional structure for the means plus function claims.

In view of the above, it is believed all of the pending claims are now in condition for allowance.

Respectfully submitted,
KLAAS, LAW, O'MEARA & MALKIN, P.C.

By: /Augustus W. Winfield/

Augustus W. Winfield, Reg. No. 34,046
1999 Broadway, Suite 2225
Denver, Colorado 80202
Telephone: (303) 298-9888

Date: September 10, 2008

PTO/SB/25 (08-08)
Approved for use through 09/30/2008. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TERMINAL DISCLAIMER TO OBVIATE A PROVISIONAL DOUBLE PATENTING REJECTION OVER A PENDING "REFERENCE" APPLICATION	Docket Number (Optional) 200404973-1

In re Application of: Scott L. Michaelis et al.

Application No.: 10/972,988

Filed: October 25, 2004

For: 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

Hewlett-Packard Development Company, L.P. the owner*, of 100 percent interest in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of any patent granted on pending **reference** Application Number 10/973,077 filed on October 25, 2004, as such term is defined in 35 U.S.C. 154 and 173, and as the term of any patent granted on said **reference** application may be shortened by any terminal disclaimer filed prior to the grant of any patent on the pending **reference** application. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and any patent granted on the **reference** application are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of any patent granted on the instant application that would extend to the expiration date of the full statutory term as defined in 35 U.S.C. 154 and 173 of any patent granted on said **reference** application, "as the term of any patent granted on said **reference** application may be shortened by any terminal disclaimer filed prior to the grant of any patent on the pending **reference** application." In the event that: any such patent granted on the pending **reference** application: expires for failure to pay a maintenance fee, is held unenforceable, is found invalid by a court of competent jurisdiction, is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321, has all claims canceled by a reexamination certificate, is reissued, or is in any manner terminated prior to the expiration of its full statutory term as shortened by any terminal disclaimer filed prior to its grant.

Check either box 1 or 2 below, if appropriate.

1. ☐ For submissions on behalf of a business/organization (e.g., corporation, partnership, university, government agency, etc.), the undersigned is empowered to act on behalf of the business/organization.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

2. ☒ The undersigned is an attorney or agent of record. Reg. No. 44,295

/Steven L. Webb/ 9/10/08
Signature Date

Steven L. Webb
Typed or printed name

(970) 898-3451
Telephone Number

☐ Terminal disclaimer fee under 37 CFR 1.20(d) is included.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner). Form PTO/SB/96 may be used for making this statement. See MPEP § 324.

This collection of information is required by 37 CFR 1.321. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/26 (05-08)
Approved for use through 09/30/2008. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT	Docket Number (Optional) 200404973-1

In re Application of: Scott L. Nicholls et al.

Application No.: 10/973,077

Filed: October 25, 2004

For: 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

HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.

The owner*, HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P., of 100 percent interest in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term **prior patent** No. 7,360,948 as the term of said prior patent is defined in 35 U.S.C. 154 and 173, and as the term of said **prior patent** is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the **prior patent** are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term as defined in 35 U.S.C. 154 and 173 of the **prior patent**, "as the term of said **prior patent** is presently shortened by any terminal disclaimer," in the event that said **prior patent** later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

Check either box 1 or 2 below, if appropriate.

1. ☐ For submissions on behalf of a business/organization (e.g., corporation, partnership, university, government agency, etc.), the undersigned is empowered to act on behalf of the business/organization.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

2. ☒ The undersigned is an attorney or agent of record. Reg. No. 46,395

/Steven L. Webb/ — Signature

9/10/08 — Date

Steven L. Webb — Typed or printed name

(970) 898-3452 — Telephone Number

☐ Terminal disclaimer fee under 37 CFR 1.20(d) included.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner). Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

This collection of information is required by 37 CFR 1.321. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Electronic Patent Application Fee Transmittal

Application Number:	10972588
Filing Date:	25-Oct-2004
Title of Invention:	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
First Named Inventor/Applicant Name:	Scott L. Michaelis
Filer:	Augustus W. Winfield/Joy Reinhart
Attorney Docket Number:	200404973-1

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Statutory disclaimer	1814	2	130	260
		Total in USD ($)		**260**

Electronic Acknowledgement Receipt

EFS ID:	3917555
Application Number:	10972588
International Application Number:	
Confirmation Number:	1079
Title of Invention:	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
First Named Inventor/Applicant Name:	Scott L. Michaelis
Customer Number:	22879
Filer:	Augustus W. Winfield/Joy Reinhart
Filer Authorized By:	Augustus W. Winfield
Attorney Docket Number:	200404973-1
Receipt Date:	10-SEP-2008
Filing Date:	25-OCT-2004
Time Stamp:	13:13:46
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Miscellaneous Incoming Letter	Transmittal.pdf	128086 2ac2308eeb0355dfe7e92acaae5492dele1de89c	no	1

Warnings:	
Information:	

2	Amendment/Req. Reconsideration-After Non-Final Reject	Amendment.pdf	51413 05e4b880b7395f6ed983babbfdab9ca0520e8f603d2	no	1
Warnings:					
Information:					
3	Claims	Claims.pdf	554753 2f0883098e2c713527a6908258c4d3b62d8b81958	no	5
Warnings:					
Information:					
4	Applicant Arguments/Remarks Made in an Amendment	Remarks.pdf	148620 a1a8736879f8804a736d6f3b327792ec0b893b85	no	2
Warnings:					
Information:					
5	Terminal Disclaimer Filed	TerminalDisclaimer1.pdf	219620 fa2524262f3bf9605f8462a8e35a7f23298787f2	no	1
Warnings:					
Information:					
6	Terminal Disclaimer Filed	TerminalDisclaimer2.pdf	230698 52c89abb26d8312e83a8ca33419236f2f01a5e6	no	1
Warnings:					
Information:					
7	Fee Worksheet (PTO-06)	fee-info.pdf	30278 6f8d3bca5d1598d0ae0f748f871b08f3b5f3bdab05	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	1363468		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200404973-1

Inventor(s): Scott L. Michaelis et al.

Application No.: 10/972,588

Filing Date: October 25, 2004

Confirmation No.: 1079

Examiner: Le, Dieu Minh T.

Group Art Unit: 2114

Title: 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

Mail Stop Amendment
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER FOR RESPONSE/AMENDMENT

Transmitted herewith is/are the following in the above-identified application:

- [X] Response/Amendment
- [ ] New fee as calculated below
- [X] No additional fee
- [X] Other Terminal Disclaimer (2)
- [ ] Petition to extend time to respond
- [ ] Supplemental Declaration

Fee$ 260

CLAIMS AS AMENDED BY OTHER THAN A SMALL ENTITY

(1) FOR	(2) CLAIMS REMAINING AFTER AMENDMENT	(3) NUMBER EXTRA	(4) HIGHEST NUMBER PREVIOUSLY PAID FOR	(5) PRESENT EXTRA	(6) RATE	(7) ADDITIONAL FEES
TOTAL CLAIMS	27	MINUS	30	= 0	X $50	$ 0
INDEP. CLAIMS	4	MINUS	4	= 0	X $210	$ 0
[ ] FIRST PRESENTATION OF A MULTIPLE DEPENDENT CLAIM					+ $370	$ 0
EXTENSION FEE	[ ] 1st Month: $120	[ ] 2nd Month: $460	[ ] 3rd Month: $1050	[ ] 4th Month: $1640		$ 0
					OTHER FEES	$ 260
					TOTAL ADDITIONAL FEE FOR THIS AMENDMENT	$ 260

Charge $ 260 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

Respectfully submitted,
Scott L. Michaelis et al.

By: /Augustus W. Winfield/
Augustus W. Winfield
Attorney/Agent for Applicant(s)
Reg No. : 34,046
Date : September 10, 2008
Telephone : (303) 298-9888

Rev 10/07 (E-Transmittal)

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
10/972,588	10/25/2004	

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY ☐		OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)		RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	09/10/2008	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 27	Minus	** 30	= 0	X $ =		OR	X $60=	0
	Independent (37 CFR 1.16(h))	* 4	Minus	***4	= 0	X $ =		OR	X $210=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)	OR	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/DALE A. HALL/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Electronic Acknowledgement Receipt

EFS ID:	3408883
Application Number:	10972588
International Application Number:	
Confirmation Number:	1079
Title of Invention:	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
First Named Inventor/Applicant Name:	Scott L. Michaelis
Customer Number:	22879
Filer:	Jay Kevin Malkin/Joy Reinhart
Filer Authorized By:	Jay Kevin Malkin
Attorney Docket Number:	200404973-1
Receipt Date:	05-JUN-2008
Filing Date:	25-OCT-2004
Time Stamp:	14:18:51
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1	Miscellaneous Incoming Letter	Transmittal.pdf	76781 2afde66f0355fa20900c6d53fa6e3612 9434d0d	no	1
Warnings:					
Information:					