2	Applicant Arguments/Remarks Made in an Amendment	Remarks.pdf	188315 284f6243e3cdb7d3995c327513b041908d653f27b	no	5
Warnings:					
Information:					
			Total Files Size (in bytes):	265096	

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.

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

RESPONSE TO NON-FINAL OFFICE ACTION MAILED 03/10/2008

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is in response to the Office action mailed March 10, 2008, in the above-captioned application.

REMARKS

No amendments have been made. Claims 1-30 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. If the present application is allowed, then the applicant will take appropriate action, including a terminal disclaimer if appropriate.

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). Applicant respectfully traverses and requests reconsideration of the Examiner's rejection.

As discussed in the response to the previous office action, independent claims 1, 16, 24, and 29 specify determining that the processor for which LOL is detected is assigned a role of boot processor; and switching the role of boot processor to a spare processor without shutting down the system's operating system.

Regarding claim 1, in applicant's response to the previous office action, the applicant discussed the differences between switching processors in Safford, and switching the role of which processor is the boot processor. In the office action dated 03/10/2008, page 4, paragraph A., the examiner states that the features being discussed are not recited in the claims. Boot processors, and switching the role of boot processors are claimed, and the point of the applicant's discussion was that switching processors in

Safford is not the same as switching the role of which processor is the boot processor. Regarding recovery from lock step, the applicant has not argued that Safford does not recover from a loss of lock step, rather the applicant has argued that recovery from lock step does not equate to switching the role of boot processor. To reiterate, the following is a summary of the points being made by the applicant:

1. Independent claims 1, 16, 24, and 29 specify a boot processor. Safford does not discuss a boot processor.

2. Safford does not teach or suggest switching the role of boot processor.

3. Re-booting a single processor is not the same as switching the role of boot processor.

In the office action dated 03/10/2008, pages 5 and 6, paragraph B., the examiner cites Safford column 4, lines 13-15 and column 5, claims 3 and 7 for support for Safford switching a boot processor. As applicant discussed in the response to the previous office action, re-booting a single processor is not the same as switching the role of which processor is the boot processor.

In the office action dated 03/10/2008, pages 6 and 7, paragraph C., the examiner discusses recovery from loss of lock step, making a hot spare, etc., none of which addresses the claim limitation of switching the role of which processor is the boot processor.

Rejections Under 35 U.S.C. § 103

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.

Claims 1-23 and 29-30 have been rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent number 6,625,749 (Quach) in light of U.S. Patent number 7,251,749 (Fox). Applicant respectfully traverses and requests reconsideration of the Examiner's rejection.

Regarding claims 1, 29 and 30, the examiner cites Quach, figure 1, and column 4, lines 38-41 for determining that a processor for which LOL is detected is assigned the role of boot processor. Figure 1 and the cited text do not illustrate or discuss anything about boot processors. The examiner stipulates that Quach does not explicitly address switching the role of boot processor, and states that Fox teaches a system response to processor failure and dynamically activates a spare processor. Nothing in that discussion provides support for a combination of Quach and Fox teaching or suggesting boot processors or switching the role of boot processors. The examiner then argues that it would be obvious to modify a combination of Quach and Fox, but the argument for modification does not address switching the role of boot processors. No *prima facie* case for obviousness has been presented.

Regarding claims 13-15, the examiner cites Quach, column 13, lines 24-26, for teaching changing a system identifier of a spare processor to that of the boot processor. The renaming being discussed has to do with registers, not processors, and in particular does not teach or suggest anything about a boot processor or switching the role of boot processor. The examiner then further cites Quach, column 13, lines 16-26, and column 14, lines 50-57. Again the discussion in column 13 is about registers. In column 14, claim 6 discusses initializing individual processors, which has nothing to do with a boot processor or switching the role of boot processor.

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: June 5, 2008

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
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
- [ ] Other
- [ ] Petition to extend time to respond
- [ ] Supplemental Declaration

Fee$

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	30	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						$
TOTAL ADDITIONAL FEE FOR THIS AMENDMENT						$ 0

Charge $ 0 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 : June 5, 2008

Telephone : (303) 298-9888

Rev 10/07 (E-TransAmd)

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

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ 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 06/05/2008	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 30	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:
/BONNIE PHOENIX/

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.

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 03/10/2008

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	

NOTIFICATION DATE	DELIVERY MODE
03/10/2008	ELECTRONIC

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.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

Office Action Summary	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 --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 December 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance 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.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *24-28* is/are allowed.

6)☒ Claim(s) *1-23,29 and 30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 October 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ 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:

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)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to the communications filed 12/12/07 in application 10/972,588.

2. Claims 1-30 are again presented for examination.

3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

4. Claims 1-30 are still rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-32 of U.S. application 10/973,077; and claims 1-26 of U.S. application 10/972,796. Although the conflicting claims are not identical, they are not patentably distinct from each other because the claimed subject matter contains obvious modifications to previous claims 1-32 of U.S. application 10/973,077; and claims 1-26 of U.S. application 10/972,796.

5. Claims 1-4, 7, 9-23, and 29 are again rejected under 35 U.S.C. 102(e) as being anticipated over Safford et al. (U.S. patent 7,085,959 hereafter referred to as Safford).

This rejection is being applied for the same reasons set forth in the previous Office Action mailed 09/12/2007.

As per claims 1-4, 7, 9-23, and 29 see the previous office action for the detailed teaching of Safford.

Applicant asserts that Safford does not disclose or suggest following limitations:

A. Safford does not discuss transfer of system information (which processor 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 claim 3), but 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.

B. Safford does not discuss boot processors, the system determining whether a processor is the boot processor, or switching the role of boot processor from one processor to another.

C. Safford does not discuss switching a hot spare to the role of boot processor and operating system to idle a spare processor .

Examiner respectfully transverses Applicant's argument as follows:

A. First, in response to Applicant's argument that the references fail to show certain features of Applicant's invention, it is noted that the feature upon which Applicant relies (i.e., ***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 claim 3), but 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***) is not recited in the rejected claim. Although the claim is interpreted in light of the specification, limitations from the specification is not read into the claims. *In re Van Guens, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).*

Second, Examiner would like to bring Applicant attention to Safford's method and apparatus for recovery from loss of lock step [abstract, col. 2, lines 13-32] comprising ***information stored in the memory*** [col. 1, lines 60 through col. 2, lines 10] and ***master/checker pair used in supporting the recovery from loss of lockstep in a multi-processors system*** [abstract, col. 2, lines 55-57]. In addition, Safford explicitly demonstrated a method and apparatus for recovery from loss of lock step [abstract, col. 2, lines 13-32] comprising ***capability to signal the computer system (i.e., operating system) to idle the lockstep of processors [col. 4, lines 30-63].*** Stafford clearly illustrated a computer system having a means for detecting and recovering of a loss of lock step including idling the lockstep of processor.

B. First, it is NOT true that Safford does not discuss boot processors, assigning the role of boot processor, data storage for assigning the role of boot processor, or transferring the role of boot processor. Safford explicitly disclosed a method and apparatus for recovery from loss of lock step [abstract, col. 2, lines 13-32] comprising ***information stored in the memory*** [col. 1, lines 60 through col. 2, lines 10] and ***master/checker pair used in supporting the recovery from loss of lockstep in a***

multi-processors system [abstract, col. 2, lines 55-57]. Safford demonstrated the re-boot processor and its role of re-boot processor to spare processor [col. 4, lines 13-35 and col. 5, claims 3 and 7] in supporting the computer data processing system for recovering from loss of lockstep in a boot processor (i.e., re-boot processor).

Second, Safford further illustrated capability of moving an architecture state of the loss of lock step processor to a spare of processor, rebooting processor unit becomes a new spare processor unit, and detecting/correcting the loss of lock step in processor [col. 5, claims 1-2, 7, and 14]. Therefore, it is clearly to an ordinary skill in the art that Stafford does teach applicant's limitation.

C. It is NOT true that Safford does not *switching hot spare to the role of boot processor and operating system to idle a spare processor*. Safford explicitly disclosed a method for recovery from loss of lock step in a computer system, comprising: receiving a loss of lock step signal from a processor unit; receiving a notice from the loss of lock step processor unit to take the loss of lock step processor unit off line; and moving an architected state of the loss of lock step processor unit to

a spare processor unit, wherein the spare processor unit becomes an active (i.e., ***hot spare***) processor unit in the computer system. Safford further demonstrated a capability of moving the architected state comprises: copying the architected state from the loss of lock step processor unit to a node controller; and copying the architected state from the node controller to the idle processor unit [col. 5, claims 1 and 5-6]. Therefore, it is clearly to an ordinary skill in the art that Stafford does teach applicant's limitation.

6. Claims 1-23, and 29-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Quach (U.S. patent 6,625,749) over Fox et al. (U.S. patent 7,251,749 hereafter referred to as Fox).

As per claims 1-2:

Quach explicitly teaches the invention. Quach teaches:

- detecting loss of lockstep (LOL) for a processor in a multi-processor system [abstract, col. 3, lines 35-36 and col. 5, lines 20-28].
- determining that the processor for which said LOL is detected is assigned a role of boot processor [fig. 1, col. 4, lines 38-48].

Quach does not explicitly address:

- switching the role of boot processor to a spare processor without shutting down the system's operating system.

However, Quach does disclose capability of:

- the processors execute in a redundant mode [abstract, figure 1];
- the error recovery routine is invoked when the processor detects a error while operating in the redundant execution mode [abstract, col. 4, lines 30-38].

In addition, Fox explicitly teaches:

- system response to processor failure in a multi-processor data processing system [abstract];
- dynamically activate spare processor when processor failure is detected in a multi-processor data processing system [col.1, lines 8-10].

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of Applicant's invention to realizing Fox's dynamically activate spare processor when processor failure is detected in a multi-processor data processing system as being the use of switching the role of boot processor to a spare processor without shutting down the

system's operating system as claimed by Applicant. This is because Quach's mutli-data processors /fault tolerant system explicitly performed data failure detection and recovery (i.e., *loss of lockstep* and/or *fail-over*) within plurality of processors via *state information, parity information*, and *boot information* processes.

This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to improve the multi-data processors, failover, and/or fault-tolerant system operation availability and network/system performance therein with a mechanism to enhance the data connectivity, data debugging, data reliability, and data throughput which eventually will increase its performance, such as data throughput between internal and external devices.

As per claims 3-4:

In addition, Fox teaches:

- the spare processor is active spare [col. 3, line 30-31].
- Causing the system's operating system to idle the active spare [col. 3 , lines 31-36].

As per claim 5:

Quach further teaches:

- the spare processor is a processor module that includes a master and slave processor that operate in lockstep [col. 3, lines 36-37].

As per claims 6 and 8:

Quach further teaches:

- 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 [col. 3, lines 35-65].

In addition, Fox teaches:

- At least one processor of the multiprocessor system [MP] is initially provided as a reserve [hot-spare] processor [abstract].

As per claim 7:

Fox further teaches:

- determining the processor that is assigned the role of spare processor [col. 4, lines 46-48].

As per claims 9-10:

Quach further teaches:

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	**09/01/2023 07:01:32 PM ET**	**185945.012400**

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	Judi Cordo
PATENT CENTER #	62727008	AUTHORIZED BY	Andrew Sommer
CUSTOMER #	56436	FILING DATE	-
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	Scott L. Michaelis

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
DA / 506990	**E202391J 06239742**	**Judi Cordo**

PRE-AUTHORIZED ACCOUNT	PRE-AUTHORIZED CATEGORY
506990	**37 CFR 1.16 (National application filing, search, and examination fees); 37 CFR 1.17 (Patent application and reexamination processing fees); 37 CFR 1.19 (Document supply fees); 37 CFR 1.21 (Miscellaneous fees and charges)**

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
1812	REQUEST FOR EX PARTE REEXAM-NON-STREAMLINED	12600.00	1	12600.00
			TOTAL AMOUNT:	**$12,600.00**

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of Michaelis *et al.*	§	Attorney Docket No.: 185945.012400
	§	
U.S. Patent No. 7,627,781	§	Customer No.: 67395
	§	
Issue Date: December 1, 2009	§	
	§	
Filing Date: October 25, 2004	§	
	§	
For: SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR	§	
	§	
	§	

Mail Stop "*Ex Parte* Reexam"
Attn: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF SERVICE

It is certified that, pursuant to 37 C.F.R. § 1.510(b)(5), copies of the following documents have been served in their entireties on the patent owner at the correspondence address provided for in 37 C.F.R. § 1.33(c):

1. Request for *Ex Parte* Reexamination of U.S. Patent No. 7,627,781 Transmittal Form, PTO/SB/57

2. Request for *Ex Parte* Reexamination of U.S. Patent No. 7,627,781 Pursuant to 37 C.F.R. § 1.510 and accompanying appendices and exhibits:

Ex. 1001	U.S. Patent No. 7,627,781 to Michaelis et al. ("the '781 Patent" or "the Challenged Patent")

Ex. 1002	U.S. File History of the '781 Patent ("U.S. File History")
Ex. 1003	Declaration of Andrew Wolfe, Ph.D.
Ex. 1004	U.S. Patent No. 7,085,959 to Safford ("Safford")
Ex. 1005	U.S. Patent No. 5,450,576 to Kennedy et al. ("Kennedy")
Ex. 1006	U.S. Patent No. 7,251,746 to Fox et al. ("Fox")
Ex. 1007	U.S. Patent No. 7,328,371 to Kalyanasundharam et al. ("Kalyanasundharam")
Ex. 1008	U.S. Patent Pub. No. 2004/0123201 to Nguyen ("Nguyen")
Ex. 1009	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")
Ex. 1010	Intel, *Extensible Firmware Interface Specification*, Version 1.10 (Dec. 1, 2002) ("Intel EFI Specification")
Ex. 1011	U.S. Patent No. 7,191,328 to Hobson ("Hobson")
Ex. 1012	Bozidar Levi, *UNIX Administration, A Comprehensive Sourcebook for Effective Systems and Network Management* (2002) ("Levi")
Ex. 1013	U.S. Patent No. 5,491,788 to Cepulis et al. ("Cepulis")
Ex. 1014	*Intel IA-64 Architecture Software Developer's Manual*, Revision 1.1 (July 2000) ("Intel Itanium Manual")
Ex. 1015	*Intel Multiprocessor Specification*, Version 1.4 (May 1997) ("Intel Multiprocessor Specification")
Ex. AA	Appendix A: Claim chart comparing claims to Kalyanasundharam in view of Kennedy, as set forth in Ground 1.
Ex. BB	Appendix B: Claim chart comparing claims to Safford in view of Kennedy, as set forth in Ground 2.

Ex. CC	Appendix C: Claim chart comparing claims to Kennedy in view of Nguyen and Fox, as set forth in Ground 3.
Ex. DD	Appendix D: Claim chart comparing claims to Safford in light of Cepulis, as set forth in Ground 4.
Ex. A	Curriculum Vitae of Andrew Wolfe, Ph. D.

3. Information Disclosure Statement, PTO/SB/08, listing references cited in the Request for Ex Parte Reexamination of U.S. Patent No. 7,624,302 pursuant to 35 U.S.C. § 302 and 37 C.F.R. § 1.510.

The copy has been served on September 1, 2023, by causing the aforementioned documents to be deposited in the United States Postal Service as priority mail postage pre-paid in a package addressed to:

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

Dated: September 1, 2023

/Jon Bills/
Reg. No. 71,895

GREENBERG TRAURIG, LLP
1750 Tysons Boulevard
McLean, VA 22102
Tel: (703) 749-1370

Counsel for Requester Unified Patents, LLC

EXHIBIT
1007

US007328371B1

(12) **United States Patent**
Kalyanasundharam et al.

(10) **Patent No.: US 7,328,371 B1**
(45) **Date of Patent: Feb. 5, 2008**

(54) **CORE REDUNDANCY IN A CHIP MULTIPROCESSOR FOR HIGHLY RELIABLE SYSTEMS**

(75) Inventors: **Vydhyanathan Kalyanasundharam**, San Jose, CA (US); **William A. Hughes**, San Jose, CA (US); **Philip E. Madrid**, Round Rock, TX (US); **Scott A. White**, Austin, TX (US); **Ajay Naini**, Saratoga, CA (US)

(73) Assignee: **Advanced Micro Devices, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 557 days.

(21) Appl. No.: **10/966,466**

(22) Filed: **Oct. 15, 2004**

(51) **Int. Cl.**
G06F 11/00 (2006.01)

(52) **U.S. Cl.** **714/11**; 712/22; 712/34

(58) **Field of Classification Search** 714/11; 712/22, 34

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,799,022	A	8/1998	Williams	
5,903,717	A *	5/1999	Wardrop	714/12
6,393,582	B1 *	5/2002	Klecka et al.	714/11
6,571,363	B1	5/2003	Steiss	
6,862,693	B2 *	3/2005	Chaudhry et al.	714/11
6,928,583	B2 *	8/2005	Griffin et al.	714/11
7,003,691	B2 *	2/2006	Safford et al.	714/11
7,085,959	B2 *	8/2006	Safford	714/11
2002/0144177	A1 *	10/2002	Kondo et al.	714/13
2005/0273653	A1 *	12/2005	Zubkow	714/11
2005/0278567	A1 *	12/2005	Wolfe et al.	714/11
2006/0248384	A1 *	11/2006	Safford	714/11

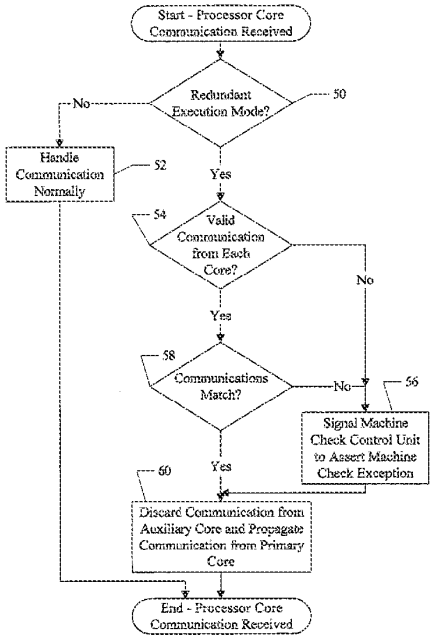
* cited by examiner

Primary Examiner—Christopher McCarthy
(74) *Attorney, Agent, or Firm*—Lawrence J. Merkel; Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.

(57) **ABSTRACT**

In one embodiment, a node comprises a plurality of processor cores and a node controller coupled to the processor cores. The node controller is configured to route communications from the processor cores to other devices in a computer system. The node controller comprises a circuit coupled to receive the communications from the processor cores. In a redundant execution mode in which at least a first processor core is redundantly executing code that a second processor core is also executing, the circuit is configured to compare communications from the first processor core to communications from the second processor core to verify correct execution of the code. In some embodiments, the processor cores and the node controller may be integrated onto a single integrated circuit chip as a CMP. A similar method is also contemplated.

19 Claims, 7 Drawing Sheets



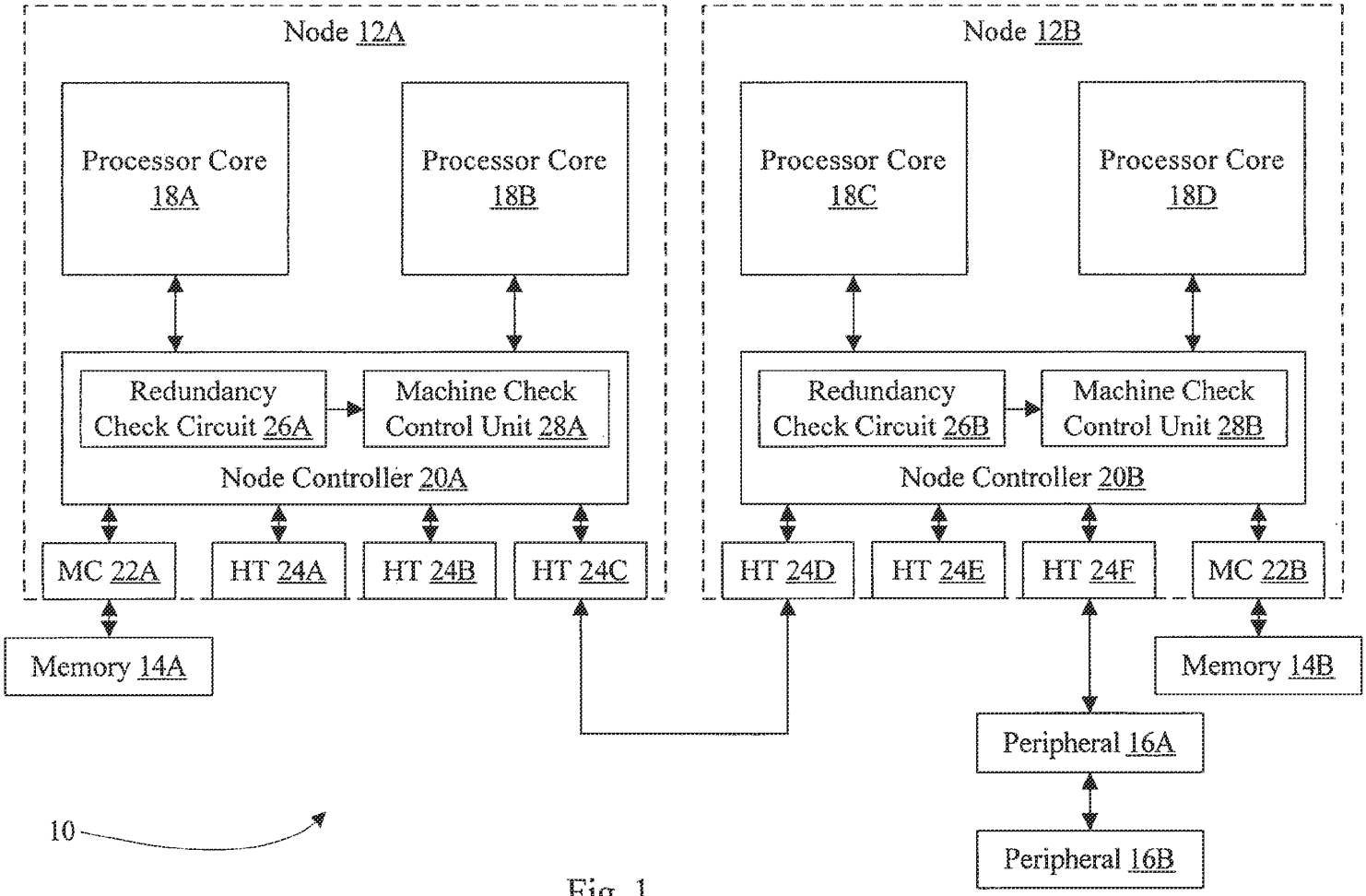

Node 12A
Node 12B
Processor Core 18A
Processor Core 18B
Processor Core 18C
Processor Core 18D
Redundancy Check Circuit 26A
Machine Check Control Unit 28A
Node Controller 20A
Redundancy Check Circuit 26B
Machine Check Control Unit 28B
Node Controller 20B
MC 22A
HT 24A
HT 24B
HT 24C
HT 24D
HT 24E
HT 24F
MC 22B
Memory 14A
Memory 14B
Peripheral 16A
Peripheral 16B
10

Fig. 1

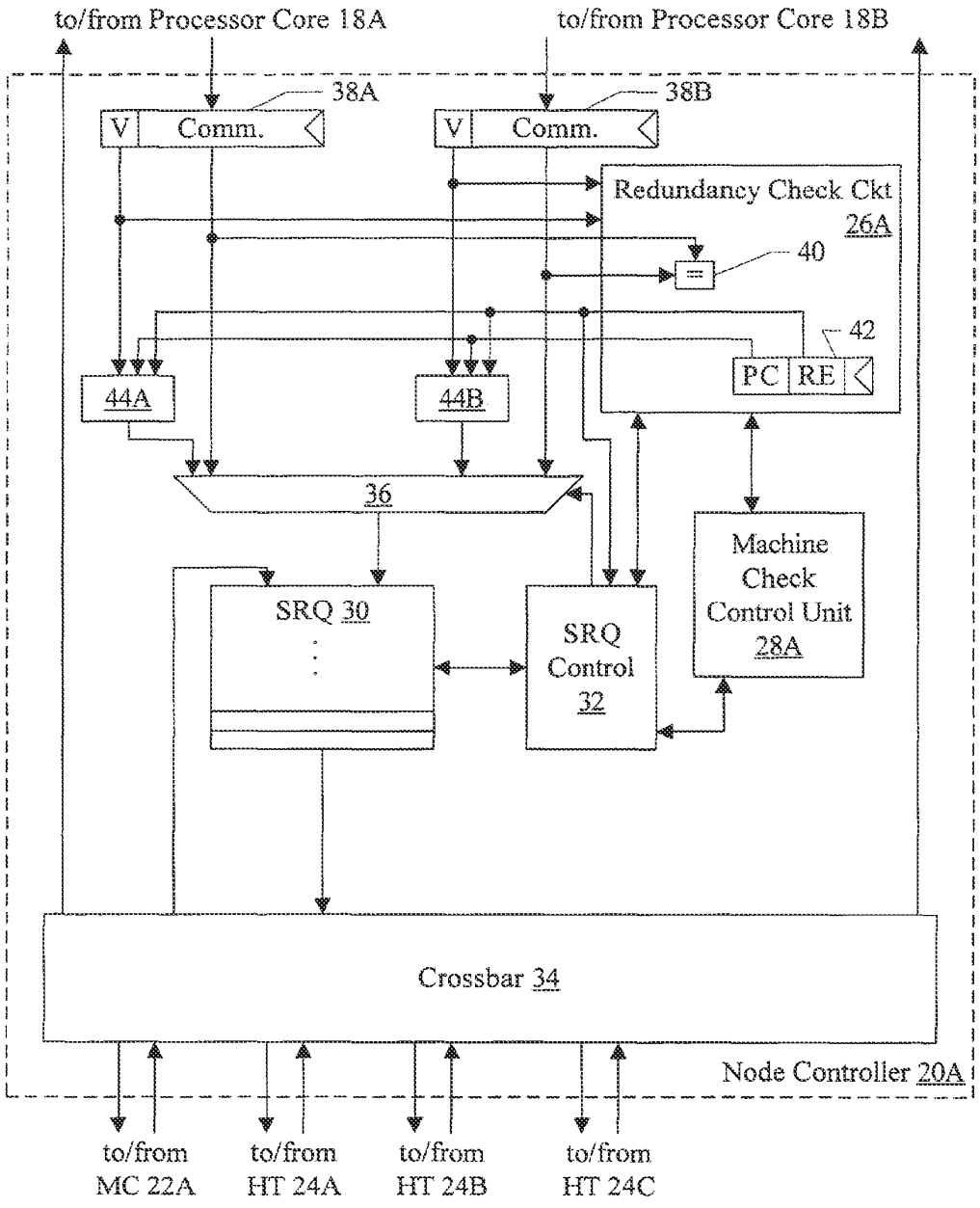

to/from Processor Core 18A
to/from Processor Core 18B
38A
38B
V
Comm.
V
Comm.
Redundancy Check Ckt
26A
40
=
42
PC
RE
44A
44B
36
Machine
Check
Control Unit
28A
SRQ 30
SRQ
Control
32
Crossbar 34
Node Controller 20A
to/from
MC 22A
to/from
HT 24A
to/from
HT 24B
to/from
HT 24C

Fig. 2

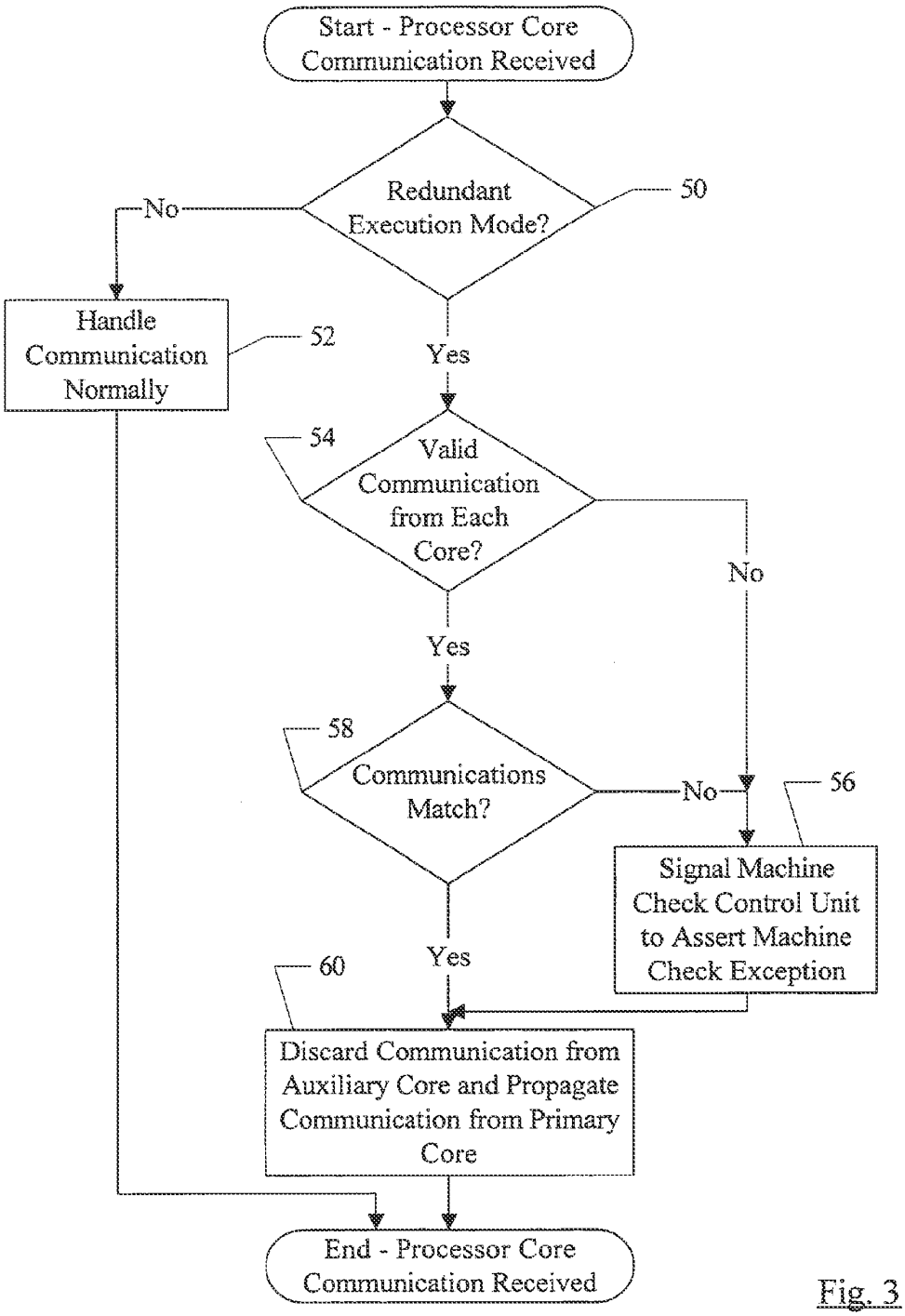

Start - Processor Core Communication Received
Redundant Execution Mode?
50
No
Yes
Handle Communication Normally
52
54
Valid Communication from Each Core?
No
Yes
58
Communications Match?
No
56
Signal Machine Check Control Unit to Assert Machine Check Exception
Yes
60
Discard Communication from Auxiliary Core and Propagate Communication from Primary Core
End - Processor Core Communication Received

Fig. 3

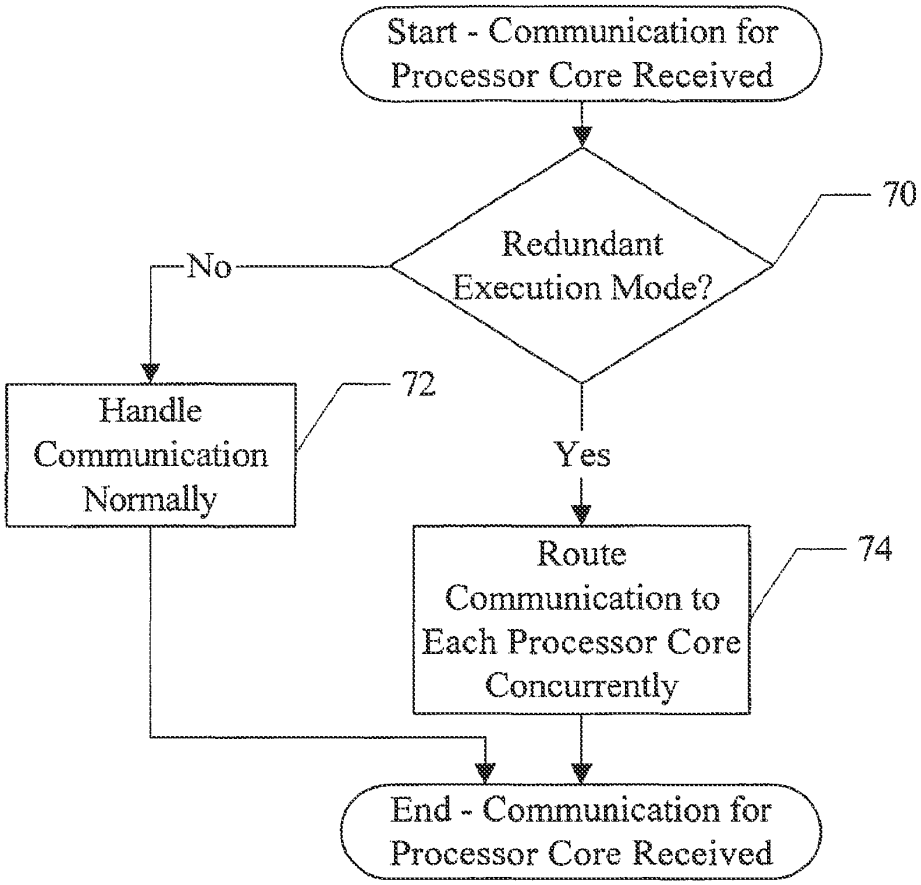

Start - Communication for Processor Core Received
Redundant Execution Mode?
70
No
Yes
Handle Communication Normally
72
Route Communication to Each Processor Core Concurrently
74
End - Communication for Processor Core Received

Fig. 4

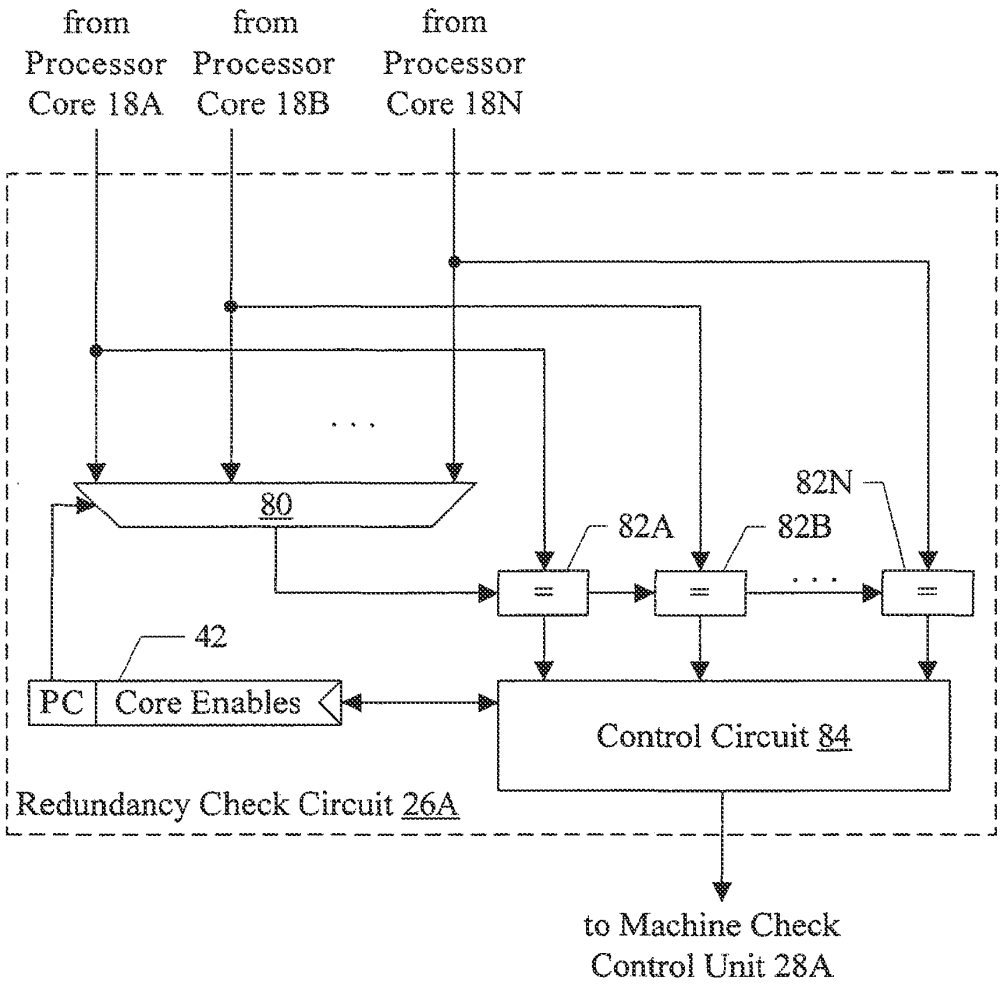

from Processor Core 18A
from Processor Core 18B
from Processor Core 18N
80
82A
82B
82N
=
=
=
42
PC
Core Enables
Control Circuit 84
Redundancy Check Circuit 26A
to Machine Check Control Unit 28A

Fig. 5

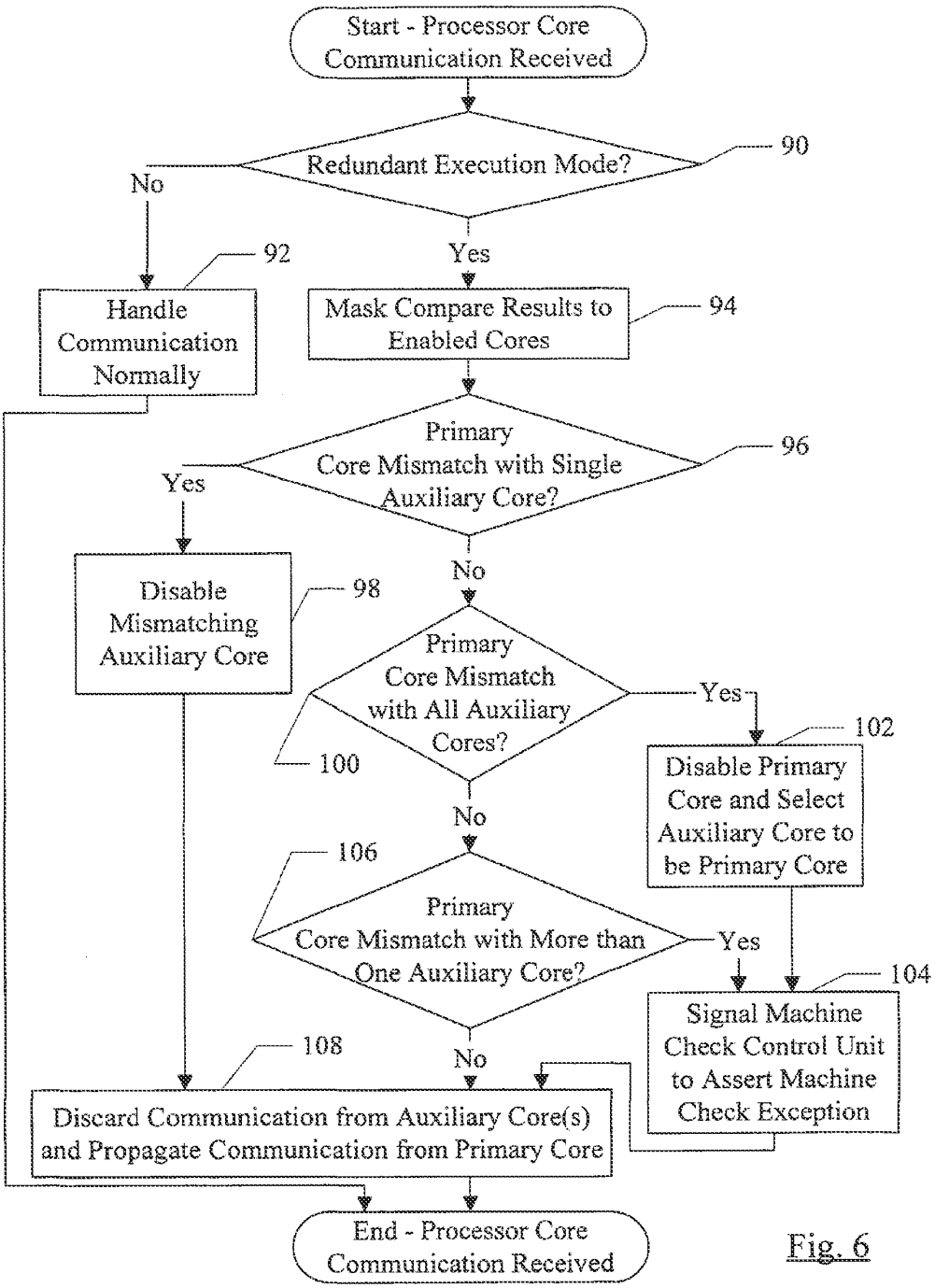

Start - Processor Core Communication Received
Redundant Execution Mode?
90
No
Yes
92
Handle Communication Normally
Mask Compare Results to Enabled Cores
94
Primary Core Mismatch with Single Auxiliary Core?
96
Yes
No
Disable Mismatching Auxiliary Core
98
Primary Core Mismatch with All Auxiliary Cores?
Yes
100
102
Disable Primary Core and Select Auxiliary Core to be Primary Core
No
106
Primary Core Mismatch with More than One Auxiliary Core?
Yes
104
Signal Machine Check Control Unit to Assert Machine Check Exception
108
No
Discard Communication from Auxiliary Core(s) and Propagate Communication from Primary Core
End - Processor Core Communication Received

Fig. 6

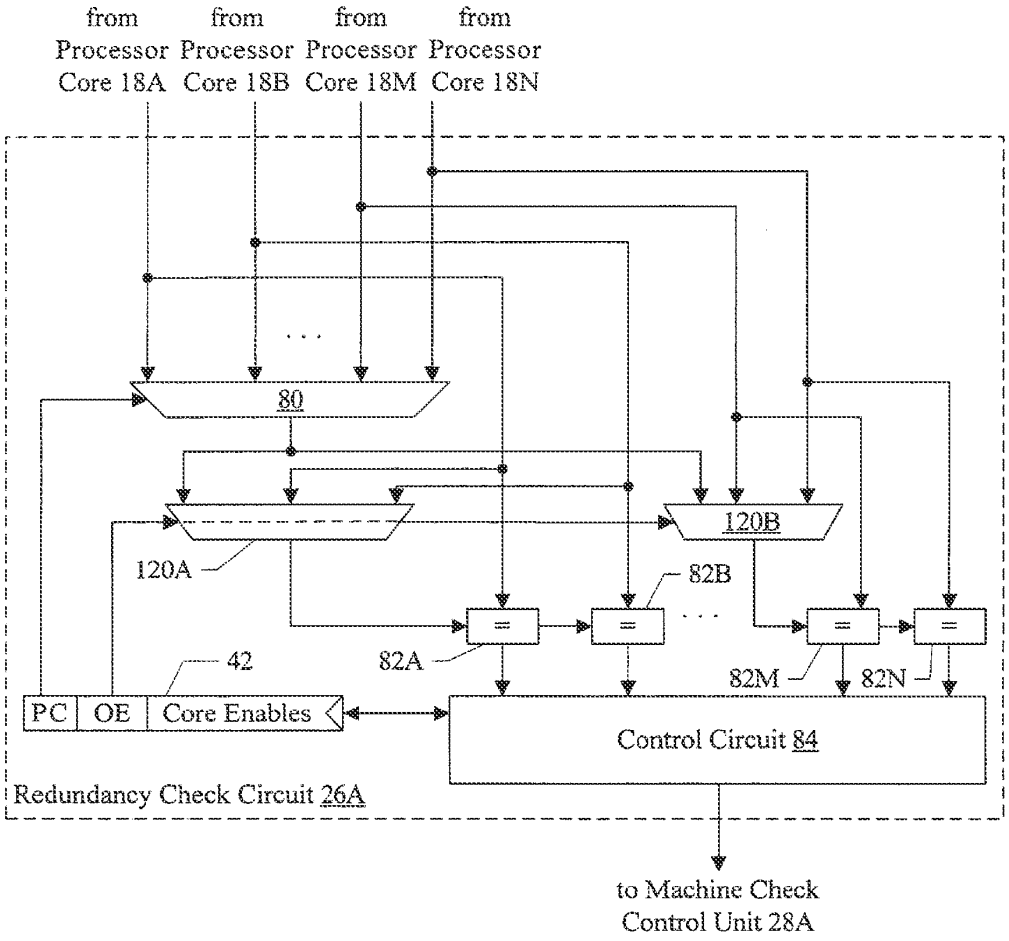

from Processor Core 18A
from Processor Core 18B
from Processor Core 18M
from Processor Core 18N
80
120A
120B
82A
82B
82M
82N
42
PC
OE
Core Enables
Control Circuit 84
Redundancy Check Circuit 26A
to Machine Check Control Unit 28A

Fig. 7

CORE REDUNDANCY IN A CHIP MULTIPROCESSOR FOR HIGHLY RELIABLE SYSTEMS

BACKGROUND

1. Field of the Invention

This invention is related to the field of multiprocessors and, more particularly, to chip multiprocessors (CMPs) and implementing redundant execution for high reliability in CMPs.

2. Description of the Related Art

Fault-tolerant computer systems are used in a variety of applications in the computing world. For example, many 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.

One of the aspects of fault-tolerant computer systems is high reliability through the early detection of errors in operation. For example, some fault-tolerant systems may implement high reliability by including multiple processors and having them redundantly execute the same software code. The results of executing the code can be compared, and any discrepancies may be detected as an error. The processors are discrete, individual integrated circuits, and thus such a solution is costly both in terms of the number of integrated circuits needed and the circuit board space required to connect the processors to the rest of the computer system. Additionally, since the processors are discrete, controlling the processors to execute in lockstep or to otherwise synchronize is complex and error prone, and typically requires additional circuitry to be included in the computer system (external from the processors) to carry out.

SUMMARY

In one embodiment, a node comprises a plurality of processor cores and a node controller coupled to the processor cores. The node controller is configured to route communications from the processor cores to other devices in a computer system. The node controller comprises a circuit coupled to receive the communications from the processor cores. In a redundant execution mode in which at least a first processor core is redundantly executing code that a second processor core is also executing, the circuit is configured to compare communications from the first processor core to communications from the second processor core to verify correct execution of the code. In some embodiments, the processor cores and the node controller may be integrated onto a single integrated circuit chip as a CMP. A similar method is also contemplated.

BRIEF DESCRIPTION OF THE DRAWINGS

The following detailed description makes reference to the accompanying drawings, which are now briefly described.

FIG. **1** is a block diagram of one embodiment of a computer system.

FIG. **2** is a block diagram of one embodiment of a node controller shown in FIG. **1**.

FIG. **3** is a flowchart illustrating operation of one embodiment of a node controller shown in FIGS. **1** and **2** in response to receiving a communication from a processor core.

FIG. **4** is a flowchart illustrating operation of one embodiment of a node controller shown in FIGS. **1** and **2** in response to receiving a communication to be routed to the processor cores.

FIG. **5** is a block diagram of one embodiment of the redundancy check circuit for use with "N" processor cores.

FIG. **6** is a flowchart illustrating operation of one embodiment of the node controller including the redundancy check circuit shown in FIG. **5** in response to receiving a communication from a processor core.

FIG. **7** is a block diagram of another embodiment of the redundancy check circuit for use with "N" processor cores.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

DETAILED DESCRIPTION OF EMBODIMENTS

Turning now to FIG. **1**, a block diagram of one embodiment of a computer system **10** is shown. In the illustrated embodiment, the computer system **10** includes nodes **12A-12B**, memories **14A-14B**, and peripheral devices **16A-16B**. The nodes **12A-12B** are coupled, and the node **12B** is coupled to the peripheral devices **16A-16B**. Each of the nodes **12A-12B** are coupled to respective memories **14A-14B**. The node **12A** comprises processor cores **18A-18B** coupled to a node controller **20A** which is further coupled to a memory controller **22A** and a plurality of HyperTransport™ (HT) interface circuits **24A-24C**. The node **12B** similarly comprises processor cores **18C-18D** coupled to a node controller **20B** which is further coupled to a memory controller **22B** and a plurality of HT interface circuits **24D-24F**. The HT circuits **24C-24D** are coupled (via an HT interface, in this embodiment) and the HT circuit **24F** is coupled to the peripheral device **16A**, which is coupled to the peripheral device **16B** in a daisy-chain configuration (using HT interfaces, in this embodiment). The memory controllers **22A-22B** are coupled to the respective memories **14A-14B**. In one embodiment, each of the nodes **12A-12B** may be a single integrated circuit chip comprising the circuitry shown therein in FIG. **1**. That is, each node **12A-12B** may be a chip multiprocessor (CMP). Other embodiments may implement the nodes **12A-12B** as two or more separate integrated circuits, as desired. Any level of integration or discrete components may be used.

In the illustrated embodiment, each of the node controllers **20A-20B** includes a redundancy check circuit and a machine check control unit. For example, the node controller **20A** includes a redundancy check circuit **26A** and a machine check control unit **28A**, and the node controller **20B** includes a redundancy check circuit **26B** and a machine check control unit **28B**. The operation of the redundancy check circuit **26A** and the machine check control unit **28A** will be described in more detail herein, and the operation of the redundancy check circuit **26B** and the machine check control unit **28B** may be similar.

The node controller **20A** may generally be configured to receive communications from the processor cores **18A-18B**, the memory controller **22A**, and the HT circuits **24A-24C** and to route those communications to the processor cores

18A-18B, the HT circuits 24A-24C, and the memory controller 22A dependent upon the communication type, the address in the communication, etc. In one embodiment, the node controller 20A includes a system request queue (SRQ) into which received communications are written by the node controller 20A. The node controller 20A may schedule communications from the SRQ for routing to the destination or destinations among the processor cores 18A-18B, the HT circuits 24A-24C, and the memory controller 22A. The node controller 20B may be similar with regard to the processor cores 18C-18D, the HT circuits 24D-24F, and the memory controller 22B. Operation of the node 12A and its components will be discussed in more detail below. Operation of the node 12B and its components may be similar.

Generally, the processor cores 18A-18B may use the interface(s) to the node controller 20A to communicate with other components of the computer system 10 (e.g. peripheral devices 16A-16B, processor cores 18C-18D, the memory controllers 22A-22B, etc.). The interface may be designed in any desired fashion. Cache coherent communication may be defined for the interface, in some embodiments. In one embodiment, communication on the interfaces between the node controller 20A and the processor cores 18A-18B may be in the form of packets similar to those used on the HT interfaces. In other embodiments, any desired communication may be used (e.g. transactions on a bus interface, packets of a different form, etc.). Generally, the communications from the processor cores 18A-18B may include requests such as read operations (to read a memory location or a register external to the processor core) and write operations (to write a memory location or external register), responses to probes (for cache coherent embodiments), interrupt acknowledgements, system management messages, etc. In the illustrated embodiment, each processor core 18A-18B may have a separate communication path to the node controller 20A. The communication path may be implemented in a variety of fashions (e.g. a bus, point-to-point interconnect in each direction, etc.).

In one embodiment, the node 12A may support a redundant execution mode in which the processor cores 18A-18B redundantly execute the same software code on the same data. The processor cores 18A-18B may be copies of the same processor design (with the same size caches, the same register files, etc.). Accordingly, if no errors in execution have occurred, communications generated by the processor core 18A should match corresponding communications generated by the processor core 18B in redundant execution mode. For example, the same address should be included in corresponding communications, the same data should be included, etc. Errors that may occur in execution may include soft errors such as soft memory errors (e.g. those detectable using an error checking code (ECC) scheme) or soft communication errors such as noise or interference in the transmission of communications. Errors may also include hard errors such as errors caused by hardware failures.

The redundancy check circuit 26A receives communications generated by the processor cores 18A-18B and, in redundant execution mode, compares the corresponding communications to verify correct execution of the software code. If a mismatch is detected in the comparison, the redundancy check circuit 26A may signal an execution error. An execution error may be any indication that an error has occurred in the execution of the software code, detected via the redundant execution. In the embodiment of FIG. 1, it may not be clear which processor core 18A-18B experienced the error, but the error is detected. Diagnostic software may be used to determine which processor core 18A-18B experienced the error and what corrective measures are to be taken. In various embodiments, the execution error may be signalled in a variety of fashions. For example, in the present embodiment, the redundancy check circuit 26A may signal a machine check exception to the machine control unit 28A. The machine check control unit 28A may cause a machine check exception in at least one of the processor cores 18A-18B (e.g. the processor core 18A-18B designated as the primary processor core described below). Other embodiments may use other exceptions, external interrupts, logging of execution errors to be checked by software, etc. to signal an execution error. Still other embodiments may use any combination of the above to signal an execution error.

In one embodiment, the processor cores 18A-18B may execute in lockstep (that is, may execute the same instructions on the same clock cycle). In CMP embodiments of the node 12A, the processor cores 18A-18B may execute at the same clock frequency and may be driven from the same clock source (derived from a clock input to the node 12A, not shown in FIG. 1). Thus, once the processor cores 18A-18B initiate execution of the software in synchronization, they may remain synchronized as long as they receive communications (e.g. responses to requests) from the node controller 20A concurrently. That is, hardware external to the node 12A may not be required to run the processor cores 18A-18B in lockstep. In embodiments in which the processor cores 18A-18B execute in lockstep, the redundancy check circuit 26A may check that corresponding communications are concurrently received from the processor cores 18A-18B (e.g. received on the same clock cycle). For example, if a valid communication is received from one of the processor cores 18A-18B and a valid communication is not concurrently received from the other processor core 18A-18B, an execution error may be detected.

The node controller 20A may be configured to propagate communications from one of the processor cores 18A-18B (e.g. from the primary processor core) and to discard communications from the other processor core 18A-18B (e.g. from the auxiliary processor core). The node controller 20A may route the propagated communications as appropriate (e.g. to the memory controller 22A or one or more of the HT circuits 24A-24C). Some communications may result in responses to be returned to the processor cores 18A-18B. For example, read requests may result in responses that return the read data. The node controller 20A may concurrently provide the response to both processor cores 18A-18B. Similarly, for external communications received by the node controller 20A that are to be routed to the processor cores 18A-18B (such as probes, system management commands, etc.), the node controller 20A may concurrently provide the communications to the processor cores 18A-18B. For a probe, the probe response from one of the processor cores 18A-18B may be forwarded to the probe source and the other probe response from the other processor core 18A-18B may be discarded after being used to verify the forwarded probe response.

Generally, one of the processor cores 18A-18B may be designated as a primary processor core and the other processor core 18A-18B is referred to as the auxiliary processor core. The primary processor core may be the processor core whose execution results will be used as the result of executing the redundantly-executed software code. The auxiliary processor core is used to verify correct execution by the primary processor core. In other embodiments, more than two processor cores may be provided in a node. In such embodiments, one processor core may be the primary core

and two or more of the remaining processor cores may be auxiliary processor cores. In one implementation, all the remaining processor cores may be used as auxiliary processor cores.

The identification of the primary processor core may be programmable (e.g. in a register in the redundancy check circuit **26**A, in one embodiment). The programmable embodiment is used as an example for the remainder of the disclosure. Alternatively, the primary processor core may be fixedly defined (e.g. via fuses in the node **12**A, pins to the node **12**A, etc.).

In a CMP embodiment, the node **12**A may provide a low cost approach to redundant execution. The node **12**A may be a single integrated circuit chip, occupying one location on a circuit board included in the computer system **10**. Additionally, no external hardware may be required, in some embodiments, as mentioned above. The embodiment of the computer system **10** may be a multiprocessor computer system (one "processor" per node **12**) with redundancy for each "processor" implemented internally. Other embodiments of the computer system **10** may have a single node **12**A and may be uniprocessor system with redundancy. Additionally, in some embodiments, the computer system **10** may execute with redundant execution mode disabled, providing an execution environment without redundancy and each processor core **18**A-**18**B may be used independently.

As mentioned above, the software code that is to be executed redundantly may be executed in lockstep, in some embodiments. By providing responses, probes, and other communications to the processor cores **18**A-**18**B concurrently, the processor cores **18**A-**18**B may naturally remain in lockstep. Synchronization of the launching of execution (to get the processor cores **18**A-**18**B executing in lockstep) may be performed in a variety of fashions. For example, a software semaphores may be used to control the beginning of execution of the software code, and the response that provides the semaphore data that releases execution may be provided concurrently to the processor cores **18**A-**18**B. In other embodiments, the processor cores **18**A-**18**B may be put in lockstep from the first instruction fetch after reset, or the first fetch of the software code to be executed redundantly, again by providing the response for the first fetch to each processor core **18**A-**18**B concurrently.

The memories **14**A-**14**B may comprise any suitable memory devices. For example, a memory **14**A-**14**B may comprise one or more RAMBUS DRAMs (RDRAMs), synchronous DRAMs (SDRAMs), double data rate (DDR) SDRAM, static RAM, etc. The address space of the computer system **10** may be divided among memories **14**A-**14**B. Each node **12**A-**12**B may include a memory map (e.g. in the node controller **20**A) used to determine which addresses are mapped to which memories **14**A-**14**B, and hence to which node **12**A-**12**B a memory request for a particular address should be routed. The memory controllers **22**A-**22**B may comprise control circuitry for interfacing to the memories **14**A-**14**B. Additionally, the memory controllers **22**A-**22**B may include request queues for queuing memory requests, etc.

The HT circuits **24**A-**24**F may comprise a variety of buffers and control circuitry for receiving packets from an HT link and for transmitting packets upon an HT link. The HT interface comprises unidirectional links for transmitting packets. Each HT circuit **24**A-**24**F may be coupled to two such links (one for transmitting and one for receiving). A given HT interface may be operated in a cache coherent fashion (e.g. between nodes **12**A-**12**B) or in a non-coherent fashion (e.g. to/from peripheral devices **16**A-**16**B). In the illustrated embodiment, the HT circuits **24**C and **24**D are coupled via coherent HT links for communicating between the nodes **12**A-**12**B. The HT circuits **24**A-**24**B and **24**E are not in use, and the HT circuit **24**F is coupled via non-coherent links to the peripheral devices **16**A-**16**B.

The peripheral devices **16**A-**16**B may be any type of peripheral devices. For example, the peripheral devices **16**A-**16**B may include devices for communicating with another computer system to which the devices may be coupled (e.g. network interface cards, circuitry similar to a network interface card that is integrated onto a main circuit board of a computer system, or modems). Furthermore, the peripheral devices **16**A-**16**B may include video accelerators, audio cards, hard or floppy disk drives or drive controllers, SCSI (Small Computer Systems Interface) adapters and telephony cards, sound cards, and a variety of data acquisition cards such as GPIB or field bus interface cards. It is noted that the term "peripheral device" is intended to encompass input/output (I/O) devices.

Generally, a processor core **18**A-**18**D may comprise circuitry that is designed to execute instructions defined in a given instruction set architecture. That is, the processor core circuitry may be configured to fetch, decode, execute, and store results of the instructions defined in the instruction set architecture. The processor cores **18**A-**18**D may comprise any desired configurations, including superpipelined, superscalar, or combinations thereof. Other configurations may include scalar, pipelined, non-pipelined, etc. Various embodiments may employ out of order speculative execution or in order execution. The processor core may include microcoding for one or more instructions or other functions, in combination with any of the above constructions. Various embodiments may implement a variety of other design features such as caches, translation lookaside buffers (TLBs), etc.

It is noted that, while the present embodiment uses the HT interface for communication between nodes and between a node and peripheral devices, other embodiments may use any desired interface or interfaces for either communication. For example, other packet based interfaces may be used, bus interfaces may be used, various standard peripheral interfaces may be used (e.g. peripheral component interconnect (PCI), PCI Express™, etc.), etc.

It is noted that, while the computer system **10** illustrated in FIG. **1** comprises two nodes **12**A-**12**B, other embodiments may implement one node or more than two nodes. Similarly, each node **12**A-**12**B may include two or more processor cores, in various embodiments. Various embodiments of the computer system **10** may include different numbers of HT interfaces per node **12**A-**12**B, differing numbers of peripheral devices coupled to one or more of the nodes, etc.

The present discussion refers to a communication from the primary processor core and a corresponding communication from an auxiliary processor core (or corresponding communications from multiple auxiliary processor cores). For a given communication, the corresponding communication may be the communication that, in the absence of any errors, should match the given communication. If the processor cores are running in lock step, the given communication and its corresponding communication(s) may be received in the node controller concurrently (e.g. on the same clock cycle).

Turning now to FIG. **2**, a block diagram illustrating one embodiment of a portion of the node controller **20**A is shown. In the illustrated embodiment, the node controller **20**A includes a system request queue (SRQ) **30**, an SRQ

control unit **32**, a crossbar **34**, a multiplexor (mux) **36**, a set of input flops **38**A-**38**B, the redundancy check circuit **26**A, and the machine check control unit **28**A. The flops **38**A-**38**B are coupled to receive communications from the processors **18**A-**18**B, respectively, and are coupled to the mux **36** and the redundancy check circuit **26**A. The mux **36** is coupled to receive selection controls from the SRQ control unit **32** and is coupled to the SRQ **30**. The SRQ control unit **32** is coupled to the SRQ **30**, the redundancy check circuit **26**A, and the machine check control unit **28**A. The SRQ **30** is coupled to the crossbar **34**, which is further coupled to receive and provide communications to the HT circuits **24**A-**24**C and the memory controller **22**A, and to provide communications to the processor cores **18**A-**18**B. In the illustrated embodiment, the redundancy check circuit **26**A may include a comparator **40** and a configuration register **42**. A pair of valid bit update circuits **44**A-**44**B are also shown in FIG. **2**, coupled to receive valid bits from the flops **38**A-**38**B (respectively), coupled to receive a PC bit and RE bit from the configuration register **42**, and coupled to the mux **36**.

The SRQ **30** is configured to store external communications (e.g. requests, responses, etc.) received by the node **12**A (e.g. through the HT circuits **24**A-**24**C) as well as internal communications generated by the memory controller **22**A (e.g. probes for read/write requests transmitted to the memory controller **22**A, read responses providing data for a read request, etc.) and the processor cores **18**A-**18**B (e.g. requests, responses, etc.). Generally, the SRQ **30** comprises multiple entries, each entry configured to store a different communication. The SRQ control unit **32** may be configured to allocate entries for received communications, and may be configured to schedule communications for transmission through the crossbar **34**. Additionally, the SRQ control unit **32** may be configured to update state in the SRQ **30** for a given communication in response to completion of various internal activities. For example, if a probe is in the SRQ **30** and the corresponding responses from the processors **18**A-**18**B are received for the probe, the SRQ control unit **32** may update the state in the entry storing the probe to record the receipt of the response and to record the coherency state reported in the response.

The crossbar **34** may comprise circuitry configured to route communications between the various sources and destinations. The sources may include the SRQ **30** (a request scheduled by the SRQ control unit **32**), the HT circuits **24**A-**24**C, and the memory controller **22**A. The destinations may include the SRQ **30**, the HT circuits **24**A-**24**C, the memory controller **22**A, and the processor cores **18**A-**18**B. Generally, received communications are routed to the SRQ **30** for later scheduling by the SRQ control unit **32**. Communications scheduled from the SRQ **30** and transmitted to the crossbar **34** may include information identifying the destination or destinations for the communication, which the crossbar **34** may use to route the communications.

In the redundant execution mode, the SRQ control unit **32** may be configured to schedule communications to be routed to the processor cores **18**A-**18**B concurrently for routing to both the processor cores **18**A-**18**B. If redundant execution mode is not active, the SRQ control unit **32** may schedule communications for routing to each of the processor cores **18**A-**18**B independently for each core, if desired. If the illustrated embodiment, the control register **42** may store a redundant mode enable (RE) bit which may indicate whether or not redundant execution mode is active (e.g. active when set, not active when clear, although the opposite meanings may also be used). The SRQ control unit **32** may receive the RE bit for use in scheduling decisions.

In the illustrated embodiment, the SRQ **30** may include a write port shared by the processor cores **18**A-**18**B and a write port (or ports) used by the crossbar **34**. The processor cores **18**A-**18**B may be configured to transmit a communication (e.g. request or response) to the node controller at most once every other clock cycle. The communications may be captured in the flops **38**A-**38**B and provided to the mux **36**. The SRQ control unit **32** may control the mux **36** to write received communications from the flops **38**A-**38**B into the SRQ **30** (or to update an existing entry of the SRQ **30**, in the case of a response). If only one communication is received in a given clock cycle, the SRQ control unit **32** may select the corresponding flop **38**A-**38**B through the mux **36**. If two communications are received in a given clock cycle, the SRQ control unit **32** may select one flop **38**A-**38**B in one clock cycle and, in the following clock cycle, select the other flop **38**A-**38**B. With the maximum transmission of one communication every other clock cycle, there is time to handle both communications before a new communication may be received. In other embodiments, a direct path from the processor cores **18**A-**18**B may be provided to the mux **36** in addition to the flops **38**A-**38**B, if desired. In still other embodiments, a write port for each processor core **18**A-**18**B may be provided on the SRQ **30** and the mux **36** may be eliminated.

More generally, a node **12**A-**12**B may include N processor cores, where N is an integer greater than one. The processor cores may be configured to transmit a communication at most once every N clock cycles. Thus, N simultaneous communications may be captured and written to the SRQ **30** using a shared write port, similar to the above discussion.

In the redundant execution mode, the communications from the primary processor core may be propagated and the communications from the auxiliary processor core(s) may be discarded. In one embodiment, each communication may include a valid bit (V in the flops **38**A-**38**B) that indicates whether or not a valid communication has been received. The valid bit update circuits **44**A-**44**B may be used to selectively clear the valid bit for communications from the auxiliary processor core. In this embodiment, the PC bit from the configuration register **42** may identify the primary processor core. Accordingly, if redundant execution mode is enabled and the processor core **18**A is not identified by the PC bit as the primary processor core, the valid bit update circuit **44**A may ensure that the valid bit from the flop **38**A is cleared when passed to mux **36**. Otherwise, the valid bit from the flop **38**A is passed unmodified. Similarly, if redundant execution mode is enabled and the processor core **18**B is not identified by the PC bit as the primary processor core, the valid bit update circuit **44**B may ensure that the valid bit from the flop **38**B is cleared when passed to mux **36**. Otherwise, the valid bit from the flop **38**B is passed unmodified. In other embodiments, the valid bits may be selectively cleared at the input to the flops **38**A-**38**B, or communications may be propagated/discarded in other fashions. In embodiments that include more than two processor cores, the PC field may be multiple bits identifying the primary processor core and additional valid bit update circuits similar to the circuits **44**A-**44**B may be included for communications from each processor core.

In the illustrated embodiment, the comparator circuit **40** may receive the communications stored in the flops **38**A-**38**B for comparison. If valid communications are received and a mismatch is detected by the comparator circuit **40**, the redundancy check circuit **26**A may signal an execution error.

The valid bits may also be received by the redundancy check circuit **26**A, to verify that the communications are concurrently received.

The machine check control unit **28**A may implement machine check functionality, at least for those machine check errors that are detected outside of the processor cores **18**A-**18**B. Generally, machine check errors may be errors detected in the operation of the hardware itself. For example, error correction code (ECC) errors for memory accesses, communication transmission errors, cyclic redundancy check (CRC) errors, parity errors, and the like may cause machine check errors. If a machine check error is detected (and, in one embodiment, if the error is enabled in the machine check control unit **28**A), a machine check exception may be delivered to at least one of the processor cores **18**A-**18**B. The machine check exception may cause the receiving processor core **18**A-**18**B to suspend current execution and to fetch instructions from a predefined machine check exception vector for processing the machine check exception. As mentioned above, in the present embodiment, one of the machine check errors is an execution error detected by the redundancy check circuit **26**A in the redundant execution mode. If an execution error is signalled by the redundancy check circuit **26**A to the machine check control unit **28**A, the machine check control unit **28**A may cause a machine check exception to be delivered to one of the processor cores **18**A-**18**B. In one embodiment, the machine check exception may be delivered to the primary processor core. In other embodiments, the machine check exception may be delivered to all processor cores, or a subset of the processor cores, if desired.

The SRQ **30** may have any construction (e.g. registers, random access memory (RAM), etc.). In one embodiment, the SRQ **30** may comprise a shift queue storing various state information for each request, an address RAM storing an address provided in the communication, and a data RAM configured to store data associated with the communication (e.g. a cache block located by the address). Each shift queue entry may comprise a pointer to the address RAM and a pointer to the data RAM, locating the corresponding address and data for a communication (if any).

Turning now to FIG. **3**, a flowchart is shown illustrating operation of one embodiment of the node controller **20**A (including the redundancy check circuit **26**A) in response to receiving a communication from a processor core **18**A-**18**B. While the blocks are shown in a particular order in FIG. **3** for ease of understanding, any order may be used. Furthermore, blocks may be performed in parallel by combinatorial logic circuitry in the node controller **20**A. Blocks, portions of the flowchart, or the flowchart as a whole may also be pipelined over multiple clock cycles, if desired. The communication received from the processor core **18**A-**18**B may be a request, a response to a previous communication transmitted to the processor core **18**A-**18**B such as a probe or system management message, etc.

If redundant execution mode is not active (decision block **50**, "no" leg), the communication may be handled normally (block **52**). That is, no comparison with other communications need be performed. The communication may be handled independent of communications from other processor cores, in general, etc.

If redundant execution mode is active (decision block **50**, "yes" leg), the redundancy check circuit **26**A may check if a valid communication is received from each processor core (decision block **54**). Since the flowchart of FIG. **3** is performed in response to receiving a communication from at least one processor core, then a valid communication should have been received from each processor core. If a valid communication was not received from each processor core (decision block **54**, "no" leg), the redundancy check circuit **26**A may signal the machine check control unit **28**A to assert a machine check exception (block **56**). In one embodiment, the machine check exception may be delivered to the primary processor core. If a valid communication is received from each processor core (decision block **54**, "yes" leg), the redundancy check circuit **26**A may determine if the communications match (decision block **58**). For example, the comparator circuit **40** may be used to compare the communications. If the communications do not match (decision block **58**, "no" leg), the redundancy check circuit **26**A may signal the machine check control unit **28**A to assert the machine check exception (block **56**). If the communications match (decision block **58**, "yes" leg), the node controller **20**A may discard the communication from the auxiliary processor core and propagate the communication from the primary processor core (block **60**). Propagating the communication may generally include transmitting the communication to its destination. For example, in the embodiment of FIG. **2**, propagating the communication may include writing the communication to the SRQ **30**, scheduling the communication, and transmitting the communication through the crossbar **34**.

It is noted that checking for the concurrent reception of valid communications (decision block **54**) and checking for a mismatch in the communications (decision block **58**) may be combined. For example, the comparator circuit **40** may receive the valid bits along with the other communication bits and may compare the valid bits in parallel with the other communication bits, detecting a mismatch if at least one bit differs. In other embodiments, the redundancy check circuit **26**A may compare the state of the signals used by the processor core **18**A to communicate with the node controller **20**A to the state of the signals used by the processor core **18**B to communicate with the node controller **20**A on a clock cycle by clock cycle basis. Such embodiments may be used, for example, if the processor cores **18**A-**18**B are executing in lock step. In such embodiments, a communication as used in the present discussion for comparison may be the state of the signals on a given clock cycle.

In one embodiment, the communication from the primary processor core may be propagated even if the machine check exception is signalled, to ensure that the primary processor core may complete the underlying instruction that caused the communication and thus ensuring that the primary processor core may take the exception. For example, if the primary processor core is expecting a response to the communication, the primary processor core may delay the exception until the response is received. Additionally, if the machine check exception is delivered to the primary processor core and not the auxiliary processor core, the communication may be propagated to ensure that the auxiliary processor core may complete the underlying instruction (e.g. by receiving the response). In other embodiments, the communication from the primary processor core may not be propagated (e.g. if the exception cancels the underlying instruction and permits the exception to be taken).

Turning now to FIG. **4**, a flowchart is shown illustrating operation of one embodiment of the node controller **20**A (including the redundancy check circuit **26**A) in response to receiving a communication for a processor core **18**A-**18**B. That is, the communication may be a communication to be routed to the processor core **18**A-**18**B. The communication may be received from the memory controller **22**A or the HT circuits **24**A-**24**C, for example. Such communications may

include a response to a communication previously transmitted by the processor core (e.g. a read response with data, a done message for a write request, etc.) or may be another communication, such as a probe for maintaining cache coherency or a system management message. While the blocks are shown in a particular order in FIG. **4** for ease of understanding, any order may be used. Furthermore, blocks may be performed in parallel by combinatorial logic circuitry in the node controller **20**A. Blocks, portions of the flowchart, or the flowchart as a whole may also be pipelined over multiple clock cycles, if desired.

If redundant execution mode is not active (decision block **70**, "no" leg), the node controller **20**A may handle the communication normally (block **72**). That is, the communication may be scheduled for routing to the processor core **18**A-**18**B to which it is addressed. If the communication is to be routed to each processor core **18**A-**18**B (e.g. a probe), the communication may be independently routed to each processor core. If redundant execution mode is active (decision block **70**, "yes" leg), the node controller **20**A may route the communication concurrently to each processor core **18**A-**18**B (block **74**). For example, in the embodiment of FIG. **2**, the communication may be scheduled when it can be routed to each processor core **18**A-**18**B, and may be routed through the crossbar **34** to both processor cores **18**A-**18**B concurrently.

As mentioned previously, in other embodiments a node **12**A may include more than two processor cores **18**A-**18**B. Thus, in redundant execution mode, there may be more than one auxiliary core. In some embodiments, all processor cores in a node may be used in redundant execution mode unless an error has been detected (one processor core as the primary processor core and remaining processor cores as the auxiliary processor cores). If an error is detected, in some cases, a processor core may be disabled. In other embodiments, a fixed or programmable number of the processor cores (less than the total number in the node) may participate in redundant execution.

In such embodiments, additional information about the error may be presumed in some cases. For example, if a communication from the primary processor core miscompares with the corresponding communication from each of the auxiliary processor cores, it may be reasonable to conclude that the primary processor core experienced the error. 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. Similarly, if a communication from the primary processor core miscompares with the corresponding communication from only one of the auxiliary processor cores (i.e. the primary processor core's communication matches with the other auxiliary processor cores), it may be reasonable to conclude that the auxiliary processor core experienced the error and the erroring auxiliary processor may be disabled. If a miscompare of a communication from the primary processor core and the corresponding communication from at least two, but not all, of the auxiliary processor cores is detected, diagnostic software may be invoked to determine the error.

FIG. **5** is a block diagram of one embodiment of a portion of the redundancy check circuit **26**A that may be used in a node that includes N processor cores, where N is an integer greater than 2. The last processor core is labeled **18**N merely to provide a distinct reference numeral in FIG. **5**.

The portion of the redundancy check circuit **26**A shown in FIG. **5** may include the logic used to compare the communications when there is more than one auxiliary core. The redundancy check circuit **26**A receives the communication from each processor core **18**A, **18**B, to **18**N (e.g. from corresponding input flops similar to flops **36**A-**36**B, in one embodiment). The communications are provided to a mux **80** and to respective comparator circuits **82**A-**82**N. The output of the mux **80** is also coupled to the comparator circuits **82**A-**82**N. The outputs of the comparator circuits **82**A-**82**N are coupled to a control circuit **84**, which is coupled to the machine check control unit **28**A and to the configuration register **42**. The configuration register **42** includes a primary core (PC) field encoded to identify which of the processor cores **18**A-**18**N is designated as the primary core. The configuration register **42** may further include a set of core enables which indicate which cores are participating in redundant execution mode. The combination of the core enables may also serve as the redundant execution mode enable, or a separate enable bit may be provided.

In the illustrated embodiment, the primary processor core is designated by programming the configuration register **42**. Thus, the communication from the designated processor core is selected through the mux **80** (under control of the value in the PC field) for comparison with the communication from all processor cores. In other embodiments, if the primary processor core is fixedly assigned, the mux **80** may be eliminated in favor of hardwiring the primary processor core's communication lines to the comparator circuits **82**A-**82**N (and a comparator circuit **82**A-**82**N for the primary processor core itself may also be eliminated).

The control circuit **84** may receive the comparison results from the comparators **82**A-**82**N, and may optionally disable a processor core or cores from comparison in the redundant execution mode and/or signal a machine check exception dependent on the comparison results. Additional details are provided below with regard to the FIG. **6**.

Turning now to FIG. **6**, a flowchart is shown illustrating operation of one embodiment of the node controller **20**A (including the redundancy check circuit **26**A) in response to receiving a communication from a processor core **18**A-**18**N. While the blocks are shown in a particular order in FIG. **6** for ease of understanding, any order may be used. Furthermore, blocks may be performed in parallel by combinatorial logic circuitry in the node controller **20**A. Blocks, portions of the flowchart, or the flowchart as a whole may also be pipelined over multiple clock cycles, if desired. The communication received from the processor core **18**A-**18**N may be a request, a response to a previous communication transmitted to the processor core **18**A-**18**N such as a probe or system management message, etc. Handling of communications to be routed to the processor cores **18**A-**18**N may be similar to the flowchart of FIG. **4**.

If redundant execution mode is not active (decision block **90**, "no" leg), the communication may be handled normally (block **92**). That is, no comparison with other communications need be performed. The communication may be handled independent of communications from other processor cores, in general, etc.

If redundant execution mode is active (decision block **90**, "yes" leg), the control circuit **84** may mask the comparison results (from the comparator circuits **82**A-**82**N) according to which processor cores are currently enabled, as indicated in the configuration register **42** (block **94**). If the masked comparison results indicate that the communication from the primary processor core mismatches with a communication with a single auxiliary processor core (and matches with communications from other auxiliary processor cores—decision block **96**, "yes" leg), the control circuit **84** may disable the mismatching auxiliary processor core (block **98**).

In one embodiment, the control circuit **84** may merely disable further comparison with the mismatching auxiliary processor core by clearing the enable bit for the mismatching auxiliary processor core in the configuration register **42**. The mismatching auxiliary processor core may continue executing. In other embodiments, the control circuit **84** may indirectly or directly disable the mismatching auxiliary processor core from executing (e.g. by causing a machine check exception and having software disable the mismatching auxiliary processor core, by signaling hardware designed to disable the mismatching auxiliary processor core from executing, by holding the mismatching auxiliary processor core in reset, by putting the mismatching auxiliary processor core in a sleep mode, etc.).

If the masked comparison results indicate that the communication from the primary processor core mismatches with all auxiliary processor cores (that are enabled) (decision block **100**, "yes" leg), 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**). As with the discussion above for disabling the auxiliary processor core, the control circuit **84** may simply disable the primary processor core from comparison by clearing the corresponding enable bit in the configuration register **42**, or may disable the primary processor core from executing at all. The control circuit **84** may also update the PC field of the configuration register **42** to reflect the newly selected primary processor core. The control circuit **84** may use any selection mechanism. For example, the processor cores may be numbered (e.g. 0 to N−1), and the control circuit **84** may increment the number of the presently selected primary processor core to select a new primary processor core. If one or more processor cores are disabled in the configuration register **42**, the control circuit **84** may increment until an enabled processor core is selected. Additionally, the control circuit **84** may signal the machine check control unit **28**A to assert a machine check exception (e.g. to the newly selected primary processor core—block **104**) to permit diagnostic software to perform any desired additional error handling. Alternatively, a machine check exception may not be signalled for this case in other embodiments.

If the masked comparison results indicate that the communication from the primary processor core miscompares with more than one auxiliary processor core, but not all auxiliary processor cores (decision block **106**, "yes" leg), the control circuit **84** may signal the machine check control unit **28**A to assert a machine check exception (e.g. to the primary processor core) to permit diagnostic software to attempt to locate the error (block **104**).

Generally, in each of the mismatch cases and the case of no mismatch, the node controller **20**A may discard the communications from the auxiliary processor cores and propagate the communication from the primary processor core (block **108**).

In another embodiment, a node **12**A may include N processor cores **18**A-**18**N, where N is an even integer (although **18**N is used merely as a unique reference character). The processor cores may be configured as N/2 pairs of redundant processor cores. Each pair may redundantly execute different software code from other pairs. In some embodiments, the processor cores may be preassigned into pairs for redundant execution purposes. For example, "odd" and "even" processor cores may be paired (numbering the processor cores from 0 to N−1 as mentioned above). Thus, processor cores **18**A-**18**B may be paired, as may other odd and even pairings up to processor cores **18**M and **18**N. Alternatively, processor core pairings may be programmable.

In a CMP embodiment of the node **12**A having N/2 redundant processor cores, N/2 separate code sequences may be redundantly executed concurrently within a single integrated circuit. That is, the CMP embodiment may provide a redundant, multiprocessing solution in a single chip implementation.

FIG. **7** is an example block diagram of a redundancy check circuit **26**A that may support both a first redundancy mode having a single primary processor core with multiple auxiliary processor cores and a paired redundancy mode having paired sets of redundant processor cores. In the embodiment of FIG. **7**, the mux **80** is provided to select communications from the various processor cores including the processor cores **18**A, **18**B, **18**M, and **18**N. The mux **80** is controlled by the PC field of the configuration register **42** as previously described. However, rather than being directly coupled to the comparators **82**A-**82**N, the output of mux **80** is provided as an input to a set of muxes such as muxes **120**A-**120**B. There may be a mux similar to muxes **120**A-**120**B for each pair of processor cores in the node **12**A. Each mux **120**A-**120**B may be further coupled to receive the communications from one pair of processor cores (e.g. mux **120**A is coupled to receive the communications from the processor cores **18**A-**18**B and the mux **120**B is coupled to receive the communications from the processor cores **18**M-**18**N). The output of each mux **120**A-**120**B is coupled as an input to a pair of comparators **82**A-**82**N (e.g. the output of the mux **120**A is coupled to the comparators **82**A-**82**B and the output of the mux **120**B is coupled to the comparators **82**M-**82**N). Each comparator **82**A-**82**N has another input coupled to receive a communication from a different one of the processor cores **18**A-**18**N and an output coupled to the control circuit **84**.

If paired redundancy mode is selected (via an OE field in the configuration register **26**A), the muxes **120**A-**120**B may select a communication from one of the odd or even processor cores to which it is coupled. If paired redundancy mode is not selected, the muxes **120**A-**120**B may select the output of the mux **80**. If paired redundancy mode is not selected, the operation of the redundancy check circuit **26**A may be similar to that described above with regard to FIGS. **5** and **6**. Operation of each pair of processor cores in the paired redundancy mode may be similar to that described above with regard to FIG. **3**.

For example, the OE field may comprise two bits, one for the odd processor core and one for the even processor core. Setting the bit for the odd processor core may cause selection of the odd processor core communications through the muxes **120**A-**120**B. Setting the bit for the even processor core may cause selection of the even processor core communications through the muxes **120**A-**120**B. If both bits are clear, paired redundancy mode is not selected. In other embodiments, the opposite meanings may be assigned to the set and clear states of the bits, or other encodings may be used for the OE field.

The odd or even processor core selected via the OE field may be viewed as the primary processor core of the pair. In other embodiments, the OE field may be used to select the paired redundancy mode and either the odd processor core or the even processor core may be preassigned as the primary processor core. In such embodiments, each mux **120**A-**120**B may be coupled to receive the output of the mux **80** and either the odd processor core communication or the even processor core communication, whichever is preas-

signed as the primary processor core. In still other embodiments, the redundancy check circuit may not support the first redundancy mode (having one primary processor core and multiple auxiliary processor cores) and the mux **80** may be eliminated. In other such embodiments, the muxes **120**A-**120**B may also be eliminated and pairs of processor cores may be hardwired to a single comparator circuit for that pair, similar to the embodiment of FIG. **2**. Additionally, the core enables in the configuration register **42** may be used to selectively enable various pairs for redundant execution. The comparison of communications in FIG. **7** may be based on receiving valid communications, or may be on a clock cycle by clock cycle basis (where a communication may be the state of the signals on a given clock cycle for comparison purposes).

Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.

What is claimed is:

1. A node comprising:

a plurality of processor cores; and

a node controller coupled to the plurality of processor cores and configured to route communications from the plurality of processor cores to other devices in a computer system, wherein the node controller comprises a circuit coupled to receive the communications from the plurality of processor cores and, in a redundant execution mode in which at least a first processor core of the plurality of processor cores is redundantly executing code that a second processor core of the plurality of processor cores is also executing, the circuit is configured to compare communications from the first processor core to communications from the second processor core to verify correct execution of the code, and wherein the node controller is configured to discard communications from the first processor core responsive to comparing the communications to corresponding communications from the first processor core and verifying that the communications and corresponding communications match.

2. The node as recited in claim **1** wherein the circuit is configured to compare a first communication from the first processor core to a second communication from the second processor core, wherein the first communication and the second communication are received concurrently by the circuit.

3. The node as recited in claim **2** wherein, if a valid communication is received from a first one of the first processor core and the second processor core and no valid communication is received from the other one of the first processor core and the second processor core, the circuit is configured to signal an execution error.

4. The node as recited in claim **2** wherein a separate communication path is provided between each of the plurality of processor cores and the node controller.

5. The node as recited in claim **1** wherein, if the circuit detects a mismatch between a first communication from the first processor core and a corresponding second communication from the second processor core, the circuit is configured to signal an execution error.

6. The node as recited in claim **5** wherein signalling the execution error comprises signalling a machine check exception.

7. The node as recited in claim **1** wherein the node controller is configured to propagate communications from the second processor core.

8. The node as recited in claim **7** wherein the node controller, in response to receiving a response to a communication propagated from the second processor core, is configured to route the response to both the first processor core and the second processor core.

9. The node as recited in claim **1** wherein a plurality of the plurality of processor cores are configured to redundantly execute the code in the redundant execution mode, wherein the plurality of the plurality of processor cores include the first processor core, the second processor core, and at least one other processor core.

10. The node as recited in claim **9** wherein one of the plurality of the plurality of processor cores is a primary processor core and remaining ones of the plurality of the plurality of processor cores are auxiliary processor cores, and wherein the circuit is configured to select communications received from the primary processor core for comparison to each of the plurality of the plurality of processor cores.

11. The node as recited in claim **10** wherein, if the circuit detects a mismatch between a communication from the primary processor core and a corresponding communication from only one auxiliary processor core, the circuit is configured to disable comparison of communications from that auxiliary processor core.

12. The node as recited in claim **10** wherein, if the circuit detects a mismatch between a communication from the primary processor core and a corresponding communication from each of the auxiliary processor cores, the circuit is configured to disable the primary processor core and select one of the auxiliary processor cores as the primary processor core.

13. The node as recited in claim **10** wherein, if the circuit detects a mismatch between a communication from the primary processor core and a corresponding communication from at least two of the auxiliary processor cores but not all of the auxiliary processor cores, the circuit is configured to signal an execution error.

14. The node as recited in claim **1** wherein the plurality of processor cores comprise a plurality of pairs of processor cores, and wherein each pair of the plurality of pairs of processor cores is configured to redundantly execute code independently of other pairs of the plurality of pairs of processor cores.

15. The node as recited in claim **1** wherein the plurality of processor cores and the node controller are integrated onto a single integrated circuit.

16. A method comprising:

running at least a first processor core of a plurality of processor cores and a second processor core of the plurality of processor cores in a redundant execution mode in which the first processor core redundantly executes the same code that the second processor core is executing;

verifying correct execution of the code in a node controller coupled to the plurality of processor cores, the verifying including comparing communications received in the node controller from the first processor core and the second processor core; and

discarding communications received from the first processor core and propagating corresponding communications received from the second processor core responsive to verifying that the communications from

the first processor core match the corresponding communications from the second processor core.

17. The method as recited in claim 16 wherein the comparing comprises comparing concurrently received communications.

18. The method as recited in claim 17 further comprising detecting a mismatch in the concurrently received communications and signalling an execution error.

19. The method as recited in claim 16 wherein the running at least the first processor core and the second processor core in the redundant execution mode comprises running each of the plurality of processor cores in the redundant execution mode, wherein one of the plurality of processor cores is a primary processor core and remaining ones of the plurality of processor cores are auxiliary processor cores, the method further comprising:

responsive to detecting a mismatch between a communication from the primary processor core and a corresponding communication from only one auxiliary processor core, disabling comparison of communications from that auxiliary processor core;

responsive to detecting a mismatch between a communication from the primary processor core and a corresponding communication from each of the auxiliary processor cores, disabling the primary processor core and selecting one of the auxiliary processor cores as the primary processor core; and

responsive to detecting a mismatch between a communication from the primary processor core and a corresponding communication from at least two of the auxiliary processor cores but not all of the auxiliary processor cores, signalling an execution error.

* * * * *

EXHIBIT
A

Andrew Wolfe Ph.D.

2005 De La Cruz Blvd STE 142
Santa Clara, CA 95050
(408) 394-1096 (mobile)
Email: awolfe@awolfe.org

Education:

Ph.D. in Electrical and Computer Engineering, Carnegie Mellon University, 1992
Visiting Graduate Student, Center for Reliable Computing, Stanford University, 1988-1989
M.S. in Electrical and Computer Engineering, Carnegie Mellon University, 1987
B.S.E.E. in Electrical Engineering and Computer Science, The Johns Hopkins University, 1985

Recent Employment:

Lecturer, [September 2013-present]
Santa Clara University

Teaching graduate and undergraduate courses on embedded computing, mechatronics, real-time systems, computer architecture, design, and community service.

Consultant, [October 2002-present]
Wolfe Consulting

Consultant on processor technology, computer systems, consumer electronics, software, design tools, and intellectual property issues. Testifying and consulting expert for IP and other technology-related litigation matters.

Sample clients include:

Google	Apple	Samsung
Sony	Huawei	HTC
AT&T	Verizon Mobile	T-Mobile
IBM	Motorola/Lenovo	Nintendo
Activision	AMD	Western Digital
Sonos	Qualcomm	LG
Netflix	Roku	Tivo
Synaptics	Sawstop	3M
Medigus	TCL	ZTE
Egnyte	Acer	Waymo

Chief Technical Officer, [1999-2002]; Sr. VP of Business Development, [2001-2002]; VP, Systems Integration, S3 Fellow , [1998 – 1999]; Director of Technology, S3 Fellow , [1997 - 1998]
SONIC|blue, Inc, Santa Clara, CA (formerly S3 Inc.)

Strategic Business Development:
Developed and implemented strategy to reposition S3 from PC graphics into the leading networked consumer electronics company.

- Acquired Diamond Multimedia and coordinated integration of communications, Rio digital music, and workstation graphics divisions into S3.
- Identified and negotiated acquisitions to grow digital media businesses including Empeg, ReplayTV, and Sensory Science.
- Identified and negotiated strategic investments including Comsilica, Intellon, KBGear Interactive, Entridia, DataPlay and others.

- Developed strategy for integrated graphics/core-logic products and established a joint venture with Via Technologies to design and market these products.
- Negotiated divestiture of graphics chip business to Via and the workstation graphics division to ATI.

Product Planning and Development:
- Drove roadmap development within SONICblue product divisions.
- Managed Business Development for all product lines.
- Led New Product Development and Corporate Vision processes.
- Acting co-General Manager of Rio digital music business in 2nd half of 2001. Responsible for all areas of product development, business development, and cost management.
- Managed development of the Savage/MX and Savage/IX mobile 3D graphics accelerators and Savage/NB system logic products.

Public Relations, Public Policy and Investor Relations:
- Present company products and strategy at industry events such as CES, Comdex, and Microprocessor Forum.
- Discuss new products and initiatives with the press.
- Promote issues of interest to SONICblue to industry groups and in Washington.
- Brief analysts, and investors on company progress. Participate in quarterly conference calls.

IP Management and Licensing:
- Negotiated and managed partnership agreements including a critical cross-licensing agreement with Intel.
- Renegotiated technology-licensing agreements with IBM for workstation graphics products.
- Evaluated outside technology opportunities, managed video research and development, and managed corporate IP strategy with legal staff including patent filings, cross licensing, and litigation.

Consulting Professor , [1999-2002]
Stanford University, Stanford, CA

Teaching computer architecture and microprocessor design.

Assistant Professor [1991 - 1997]
Princeton University, Princeton, NJ

Teaching and research in the Electrical Engineering department. Research in embedded computing systems, multimedia, video signal processors, compiler optimization, and high performance computer architecture. Principal investigator or project manager for ~$6M in funded research.

Visiting Assistant Professor , [1992]
Carnegie Mellon University, Pittsburgh, PA

Research and preparation of teaching materials on advanced microprocessor designs including new superscalar and superpipelined processor architectures.

Founder and Vice President and Consultant, [1989 - 1995]
The Graphics Technology Company, Inc., Austin, TX

Founded company to develop touch-sensitive components and systems for the first generation of PDA devices and interactive public systems. Obtained financing from Gunze Corp., Osaka, Japan. Company is now part of 3M.

Senior Electrical Engineer, [1989]
ESL - TRW, Advanced Technology Division, Sunnyvale, CA

Designed the architecture for an Intel i860-based multiple-processor digital signal processing system for advanced military applications. Designed several FPGA interface chips for VME-bus systems.

Design Consultant, [1986 -1987]
Carroll Touch Division, AMP Inc., Round Rock, TX

Developed several new technologies for touch-screen systems. Designed the first ASIC produced for AMP, a mixed-signal interface chip for controlling touch-screen sensors. Developed the system electronics, system firmware, and customer utility software for numerous products including those based on the new ASIC.

Senior Design Engineer, [1983 -1985]
Touch Technology Inc., Annapolis, MD

Advisory Boards:

Director, Turtle Beach Corporation (NASDAQ:HEAR) (formerly Parametric Sound Corporation), KBGear Interactive, Inc., Comsilica, Inc., Rioport.com, various S3 subsidiaries.

Technical Advisory Boards, Ageia, Inc., Intellon, Inc., Comsilica, Inc., Entridia, Inc., Siroyan, Ltd., BOPS, Inc, Quester Venture Funds

Carnegie Mellon University Silicon Valley Advisory Board; Johns Hopkins University Tech Transfer Advisory Board

Awards:

IEEE Fellow - for contributions in hardware code compression of embedded software, power consumption analysis, and optimization, 2022
IEEE Computer Society Distinguished Contributor - 2021
Micro Test-of-Time Award (in recognition of one of the ten most influential papers of the first 25 years of the symposium), 2014
Business 2.0 "20 Young Executives You Need to Know", 2002
Walter C. Johnson Prize for Teaching Excellence, 1997.
Princeton University Engineering Council Excellence in Teaching Award, Spring 1996
AT&T/Lucent Foundation Research Award, 1996.
Walter C. Johnson Prize for Teaching Excellence, 1995
IEEE Certificate of Appreciation, 1995, 2001.
AT&T Foundation Research Award, 1993.
Semiconductor Research Corporation Fellow, 1986 - 1991.
Burroughs Corporation Fellowship in Engineering, 1985 - 1986.

Professional Activities:

Program Chair: Micro-24, 1991, Hot Chips 13, 2001.

General Chair: Micro-26, 1993, Micro-33, 2000.

Associate Editor: IEEE Computer Architecture Letters; ACM Transactions in Embedded Computing Systems

Speaker at CES, WinHec, Comdex, Intel Dev. Forum, Digital Media Summit, Microprocessor Forum, etc.

Keynote speaker at Micro-34, ICME 2002

IEEE B. Ramakrishna Rau Award committee – 2012-2016

IEEE Computer Society Awards Committee – 2015

CES Awards Judge – 2016

Entrepreneurship Mentor – Draper University

Over 50 refereed publications.

Publications since January 2006:

Wolfe, A., "Retrospective on Code Compression and a Fresh Approach to Embedded Systems", IEEE MICRO, July/Aug. 2016, Invited paper.

Patents:

U.S. Pat. 5,041,701 – *Edge Linearization Device for a Contact Input System*, Aug. 20, 1991.
U.S. Pat. 5,438,168 – *Touch Panel*, Aug. 1, 1995.
U.S. Pat. 5,736,688 – *Curvilinear Linearization Device for Touch Systems*, Apr. 7, 1998.
U.S. Pat. 6,037,930 – *Multimodal touch sensitive peripheral device*, March 14, 2000.
U.S. Pat. 6,408,421 – *High-speed asynchronous decoder circuit for variable-length coded data*, June 18, 2002.
U.S. Pat. 6,865,668 – *Variable-length, high-speed, asynchronous decoder circuit*, March 8, 2005
U.S. Pat. 7,079,133 – *Superscalar 3D Graphics Engine*, July 18, 2006
EP 1 661 131 B1 – *PORTABLE ENTERTAINMENT APPARATUS*, Jan. 21, 2009
U.S. Pat. 7,555,006 – *Method and system for adaptive transcoding and transrating in a video network*, June 30, 2009
U.S. Pat. 7,996,595 – *Interrupt Arbitration for Multiprocessors*, Aug. 9, 2011
EP 2 241 979 B1 – *Interrupt Arbitration for Multiprocessors*, Oct. 10, 2011
U.S. Pat. 8,131,970 – *Compiler Based Cache Allocation*, March 6, 2012
U.S. Pat. 8,180,963 – *Hierarchical read-combining local memories*, May 15, 2012
U.S. Pat. 8,193,941 – *Snoring Treatment*, June 5, 2012
U.S. Pat. 8,203,541 – *OLED display and sensor*, June 19, 2012
U.S. Pat. 8,243,045 – *Touch-sensitive display device and method*, August 14, 2012
U.S. Pat. 8,244,982 – *Allocating processor cores with cache memory associativity*, August 14, 2012
U.S. Pat. 8,260,996 – *Interrupt Optimization for Multiprocessors*, Sept. 4, 2012
101185761 (KR) – *Noise Cancellation for Phone Conversation,* Sept. 19, 2012
101200740 (KR) – *OLED display and sensor*, November 7, 2012
101200741 (KR) – T*ouch-sensitive display device and method*, November 7, 2012
U.S. Pat. 8,321,614 – *Dynamic scheduling interrupt controller for multiprocessors*, Nov. 27, 2012
U.S. Pat. 8,352,679 – *Selectively securing data and/or erasing secure data caches responsive to security compromising conditions*, Jan. 8, 2013
U.S. Pat. 8,355,541 – *Texture Sensing*, Jan. 15, 2013
U.S. Pat. 8,370,307 – *Cloud Data Backup Storage Manager*, Feb. 5, 2013
U.S. Pat. 8,398,451 – *Tactile Input Interaction*, March. 19, 2013
JP 5241032 B2 – *Routing Across Multicore Network Using Real World or Modeled Data*, April 13, 2013
ZL201010124820.3 – *Interrupt Optimization for Multiprocessors,* April 17, 2013
U.S. Pat. 8,428,438 – *Apparatus for Viewing Television with Pause Capability*, April 23, 2013
JP 5266197 B2 – *Data Centers Task Mapping,* May 10, 2013
U.S. Pat. 8,508,498 – *Direction and Force Sensing Input Device*, August 13, 2013
U.S. Pat. 8,547,457 – *Camera Flash Mitigation*, October 1, 2013
U.S. Pat. 8,549,339 – *Processor core communication in multi-core processor*, October 1, 2013
101319048 (KR) – *Camera Flash Mitigation*, October 10, 2013
U.S. Pat. 8,628,478 – *Microphone for remote health sensing*, January 14, 2014
101362017 (KR) – *Thread Shift: Allocating Threads to Cores*, Feb. 5, 2014
101361928 (KR) – *Cache Prefill on Thread Migration*, Feb. 5, 2014
101361945 (KR) – *Mapping Of Computer Threads onto Heterogeneous Resources*, Feb. 5, 2014
JP 5487307 B2 – *Mapping Of Computer Threads onto Heterogeneous Resources*, Feb. 28, 2014
JP 5484580 B2 – *Task Scheduling Based on Financial Impact,* Feb. 28, 2014
JP 5487306 B2 – *Cache Prefill on Thread Migration*, Feb. 28, 2014
101372623 (KR) – *Power Management for Processor*, March. 4, 2014
101373925 (KR) – *Allocating Processor Cores with Cache Memory Associativity*, March 6, 2014
U.S. Pat. 8,676,668 – *Method for the determination of a time, location, and quantity of goods to be made available based on mapped population activity*, March 18, 2014
U.S. Pat. 8,687,533 – *Energy Reservation in Power Limited Networks*, April 1, 2014
101388735 (KR) – *Routing Across Multicore Networks Using Real World or Modeled Data*, April 17, 2014
U.S. Pat. 8,725,697 – *Cloud Data Backup Storage*, May 13, 2014
U.S. Pat. 8,726,043 – *Securing Backing Storage Data Passed Through a Network*, May 13, 2014
ZL201010124826.0 – *Dynamic scheduling interrupt controller for multiprocessors*, May 14, 2014
JP 5547820 B2 – *Processor core communication in multi-core processor*, May 23, 2014
U.S. Pat. 8,738,949 – *Power Management for Processor*, May 27, 2014

U.S. Pat. 8,751,854 – *Processor Core Clock Rate Selection*, June 10, 2014
JP 5559891 B2 – *Thermal Management in Multi-Core Processor*, June 13, 2014
101414033 (KR) – *Dynamic Computation Allocation*, June 25, 2014
JP 5571184 B2 – *Dynamic Computation Allocation*, July 4, 2014
101426341 (KR) – *Processor core communication in multi-core processor*, May 23, 2014
U.S. Pat. 8,799,671 – *Techniques for Detecting Encrypted Data*, Aug 5, 2014
101433485 (KR) – *Task Scheduling Based on Financial Impact,* Aug. 18, 2014
U.S. Pat. 8,824,666 – *Noise Cancellation for Phone Conversation,* Sept. 2, 2014
U.S. Pat. 8,836,516 – *Snoring Treatment*, Sept. 16, 2014
U.S. Pat. 8,838,370 – *Traffic flow model to provide traffic flow information*, Sept. 16, 2014
U.S. Pat. 8,838,797 – *Dynamic Computation Allocation*, Sept. 16, 2014
U.S. Pat. 8,854,379 – *Routing Across Multicore Networks Using Real World or Modeled Data*, Oct. 7, 2014
JP 5615361 B2 – *Thread Shift: Allocating Threads to Cores*, Oct. 15, 2014
U.S. Pat. 8,866,621 – *Sudden infant death prevention clothing*, Oct. 21, 2014
U.S. Pat. 8,881,157 – *Allocating threads to cores based on threads falling behind threads*, Nov. 4, 2014
ZL201080024755.5 – *Camera Flash Mitigation*, Nov 5, 2014
U.S. Pat. 8,882,677 – *Microphone for remote health sensing*, Nov. 11, 2014
U.S. Pat. 8,924,743 – *Securing Data Cache through Encryption*, December 30, 2014
U.S. Pat. 8,994,857 – *Camera Flash Mitigation*, March 31, 2015
JP 5699140 B2 – *Camera Flash Mitigation*, April 8, 2015
ZL201080035189.8 – *Thread Shift: Allocating Threads to Cores*, June 10, 2015
ZL201180005030.6 – *Processor core communication in multi-core processor*, June 10, 2015
U.S. Pat. 9,143,814 – *Method and system for adaptive transcoding and transrating in a video network*, Sept 22, 2015
ZL201080035177.5 – *Mapping Of Computer Threads onto Heterogeneous Resources*, Oct. 14, 2015
U.S. Pat. 9,178,694 – *Securing Backing Storage Data Passed Through a Network*, November 3, 2015
U.S. Pat. 9,189,282 – *Thread-to-core mapping based on thread deadline, thread demand, and hardware characteristics data collected by a performance counter*, November 17, 2015
U.S. Pat. 9,189,448 – *Routing image data across on-chip networks*, November 17, 2015
U.S. Pat. 9,208,093 – *Allocation of memory space to individual processor cores*, December 8, 2015
U.S. Pat. 9,239,994 – *Data Centers Task Mapping,* January 19, 2016
ZL201080036611.1 – *Allocating Processor Cores with Cache Memory Associativity*, January 20, 2016
EP2228779 B1 – *Traffic flow model to provide traffic flow information*, Jan. 27, 2016
U.S. Pat. 9,262,628 – *Operating System Sandbox,* February 16, 2016
GB2485682 – *Mapping Of Computer Threads onto Heterogeneous Resources*, Sept. 28, 2016
U.S. Pat. 9,330,137 – *Cloud Data Backup Storage Manager*, May. 3, 2016
ZL201080035185.X – *Cache Prefill on Thread Migration*, Aug. 24, 2016
U.S. Pat. 9,519,305 – *Processor Core Clock Rate Selection*, December 13, 2016
U.S. Pat. 9,569,270 – *Mapping thread phases onto heterogeneous cores based on execution characteristics and cache line eviction count*, February 14, 2017
GB2485683 – *Thread Shift: Allocating Threads to Cores*, Oct. 18, 2017
U.S. Pat. 9,852,435 – *Telemetrics based location and tracking*, December 26, 2017.
U.S. Pat. 9,915,994 – *Power management for processor*, March 13, 2018
U.S. Pat. 9,927,254 – *Traffic flow model to provide traffic flow information*, March 27, 2018
EP2254048 B1 – *Thread Mapping in Multi-Core Processors*, August 29, 2018
U.S. Pat. 10,860,432 – *Cloud Data Backup Storage Manager*, December 8, 2020

Expert testimony by deposition or at trial – April 15, 2018 - -present

Case	Venue	Case Number
Joe Andrew Salazar v HTC Corporation	E.D. Texas	2:16-cv-010986-JRG-RSP
Hitachi Maxell, Ltd. v. ZTE Corporation and ZTE (USA), Inc.	E.D. Texas	5:16-cv-00179 5:16-cv-00178
INTER PARTES REVIEW OF U.S. PATENT NO. 7,663,506 (Mediatek v AMD)	PTAB	IPR2017-00101 IPR2017-00102
Papst Licensing GmbH & Co. KG v. Samsung Electronics Co.,Ltd. and Samsung Electronics America, Inc.	E.D. Texas	6:15-CV-1102
HTC Corporation v. Telefonaktiebolaget LM Ericsson	E.D. Texas	6: 18-cv-00243-JRG
Seven Networks, LLC v ZTE (USA) Inc and ZTE Corporation	N. D. Texas - Dallas	3:17-CV-1495
AGIS Software Development, LLC v. HTC Corporation	E. D. Texas	2:17-cv-514
Barbaro Technologies, LLC v. Niantic, Inc	N.D. CA	3:18-cv-02955-RS
Immersion Corp. v. Samsung Electronics America, Inc. et al	E.D. Texas	2:17-cv-00572
INTER PARTES REVIEW OF U.S. PATENT NO. 7,171,526 (Northstar Innovations – Micron)	PTAB	IPR2018-01004 IPR2018-01005
CISCO SYSTEMS, INC., vs. UNILOC USA, INC., UNILOC 2017 LLC and UNILOC LICENSING USA LLC	N.D. CA	3:18-cv-04991-SI
INTER PARTES REVIEW OF U.S. PATENT NO. 8,020,014 (Intel/VLSI)	PTAB	IPR2018-01661 IPR2018-01312
INTER PARTES REVIEW OF U.S. PATENT NO Patent 9,294,799 (Comcast/Rovi)	PTAB	IPR2019-00299
Inter Partes Review of U.S. Patent No 7720929 (Unified Patents v Datascape)	PTAB	IPR2019-01115
Solas OLED Ltd,, v. Samsung Display Co,, Ltd,, Samsung Electronics Co, Ltd,, and Samsung Electronics America, Inc,,	PTAB	IPR2019-01668
INTER PARTES REVIEW OF U.S. PATENT NO Patent 8,973,069 (Comcast/Rovi)	PTAB	IPR2019-01434
U.S. Patent No. 8,448,215 IPR (Comcast v Rovi)	PTAB	IPR2019-01353
U.S. PATENT NO. 8,847,898 IPR filing (Samsung v Neodron)	PTAB	IPR2020-00234
U.S. PATENT NO. 8,610,009 IPR filing (Samsung v Neodron)	PTAB	IPR2020-00225
U.S. PATENT NO. 10,365,747 IPR filing (Samsung v Neodron)	PTAB	IPR2020-00308
AGIS Software Development LLC v. Google LLC, AGIS Software Development LLC v. WAZE Mobile Limited,and AGIS Software Development LLC v. Samsung Elecs. Co., Ltd. et al	E.D. Texas	2:19-cv-00361-JRG 2:19-cv-00359-JRG 2:19-cv-00362-JRG
Inter Partes Review of U.S. Patent No 8,112,670 (Sony)	PTAB	IPR2020-00726
U.S. Patent 8,819,505 IPR (Intel v PACT)	PTAB	IPR2020-00525
Inter Partes Review of U.S. Patent No 8,078,540 (Sony)	PTAB	IPR2020-00922
Inter Partes Review of U.S. Patent No 9,037,807 (Intel v PACT)	PTAB	IPR2020-00540

HONG KONG UCLOUDLINK NETWORK TECHNOLOGY LIMITED AND UCLOUDLINK (AMERICA), LTD., vs. SIMO HOLDINGS INC. AND SKYROAM, INC.,	N.D. CA	3:18-cv-05031-EMC
Joe Andrew Salazar v AT&T Mobility, LLC et al.	E.D. Texas	2:20-cv-00004-JRG
Innovative Memory Solutions Inc. v Micron, Inc	Delaware	14-1480 RGA
Inter Partes Review of U.S. Patent No. 6,411,941	PTAB	IPR2021-01338 IPR2021-01406
Maxell, Ltd. et al v. VIZIO, Inc	C.D. CA	2-21-cv-06758
Inter Partes Review of U.S. Patent No. 7,619,912	PTAB	IPR2022-00615
Inter Partes Review of U.S. Patent Nos. 11,016,918 and 11,232,054	PTAB	IPR2022-00996 IPR2022-00999
In the Matter of Certain Smart Televisions, - Maxell, Ltd. et al v. VIZIO, Inc	ITC	337-TA-1338
INTER PARTES REVIEW OF U.S. PATENT NOS. 8,787,060 and 9,318,160	PTAB	IPR2022-01427 IPR2022-01428

EXHIBIT
1005

US005450576A

United States Patent [19]

Kennedy

[11] Patent Number: 5,450,576

[45] Date of Patent: Sep. 12, 1995

[54] **DISTRIBUTED MULTI-PROCESSOR BOOT SYSTEM FOR BOOTING EACH PROCESSOR IN SEQUENCE INCLUDING WATCHDOG TIMER FOR RESETTING EACH CPU IF IT FAILS TO BOOT**

[75] Inventor: **Barry Kennedy**, Santa Ana, Calif.

[73] Assignee: **Ast Research, Inc.**, Irvine, Calif.

[21] Appl. No.: **52,818**

[22] Filed: **Apr. 26, 1993**

Related U.S. Application Data

[63] Continuation of Ser. No. 721,692, Jun. 26, 1991, abandoned.

[51] **Int. Cl.**[6] **G06F 11/20**

[52] **U.S. Cl.** **395/650**; 371/72; 364/DIG. 1; 364/230.6; 364/267.9; 364/268.3; 395/200.01; 395/182.11; 395/183.12

[58] **Field of Search** 395/575, 800, 700; 371/9.1, 11.1, 11.3, 12, 16.3

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,786,433	1/1974	Notley et al.	395/650
4,117,974	10/1978	Ciaramella	371/51.1
4,127,768	11/1978	Negi et al.	395/425
4,578,773	3/1986	Desai et al.	395/725
4,593,349	6/1986	Chase et al.	395/750
4,679,166	7/1987	Berger et al.	395/650
4,803,682	2/1989	Hara et al.	371/12
4,882,669	11/1989	Miura et al.	371/12
4,890,285	12/1989	Dichiara	371/16.3
4,943,911	7/1990	Kopp et al.	395/375
4,975,831	12/1990	Nilsson et al.	395/800
5,038,320	8/1991	Heath et al.	371/11.1
5,070,450	12/1991	Holman, Jr. et al.	395/750
5,109,521	4/1992	Culley	395/800
5,155,833	10/1992	Cullison et al.	395/700
5,168,555	12/1992	Byers et al.	395/725

FOREIGN PATENT DOCUMENTS

486304A2	11/1991	European Pat. Off. .
63-083856	4/1988	Japan .
63-104167	5/1988	Japan .

OTHER PUBLICATIONS

Taetow, W. "DSP mit universeller Struktur", *Elektronik*, pp. 96–102, vol. 37, No. 19., Sep. 1988 with translation.

Matthes, Wolfgang, "Multimikrorechnersysteme 9", *Radio Fernsehen Elektronik*, pp. 773–776, vol. 33, No. 12, Dec. 1984 with translation.

"Start-Up Master Processor Selection Method for Multi-Processor Systems", IBM Technical Disclosure Bulletin, vol. 33, No. 4, Sep. 1990, pp. 375–376.

Bursky, Dave, "More, Not Fewer, Chips Give PC's Flexible Options," *Electronic Design*, vol. 38, No. 10, pp. 95–98, May 24, 1990.

Thorson, Mark, "S3 Introduces Flexible PC Chip Set Family," *Microprocessor Report*, vol. 4, N. 9, pp. 1–2, 9–11, May 18, 1990.

Thorson, Mark, "ACI Bus Supports 486 Multiprocessing," *Microprocessor Report, vol. 4, No. 11, pp. 9–11, Jun. 20, 1990.*

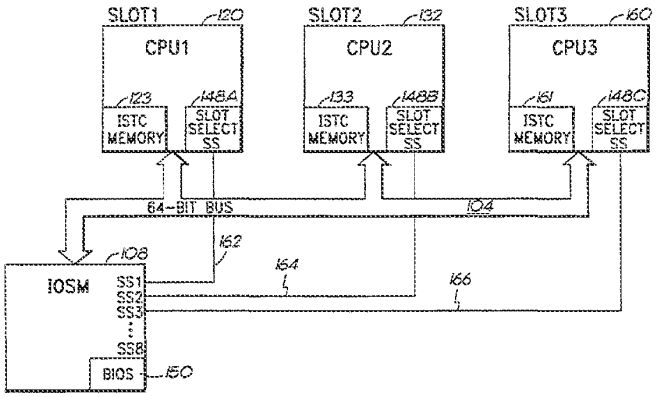
S3, Advanced Cache Controller Preliminary Information Sheet (Apr. 1990).

Primary Examiner—Parshotam S. Lall
Assistant Examiner—Richard L. Ellis
Attorney, Agent, or Firm—Knobbe, Martens, Olson & Bear

[57] **ABSTRACT**

A system for controlling initialization and self test operations in a multiprocessor system facilitates the use of central processing units based around differing microprocessor types. More specifically, the present invention involves storing configuration information, initialization self-test code, and boot code specific to each processor, memory module, or I/O circuit board in non-executable form in a non-volatile memory, on the respective circuit board, and storing the executable portion of the boot code needed by the initial boot processor in a centrally accessible non-volatile memory. Moreover, the present invention provides a system to automatically transfer default system initialization operations from a default processor to a first alternative processor if the default processor fails, and to automatically transfer initialization operations to a second alternative processor if the first alternative processor fails, and so forth depending upon how many alternative processors are installed in the multiprocessor system.

8 Claims, 4 Drawing Sheets



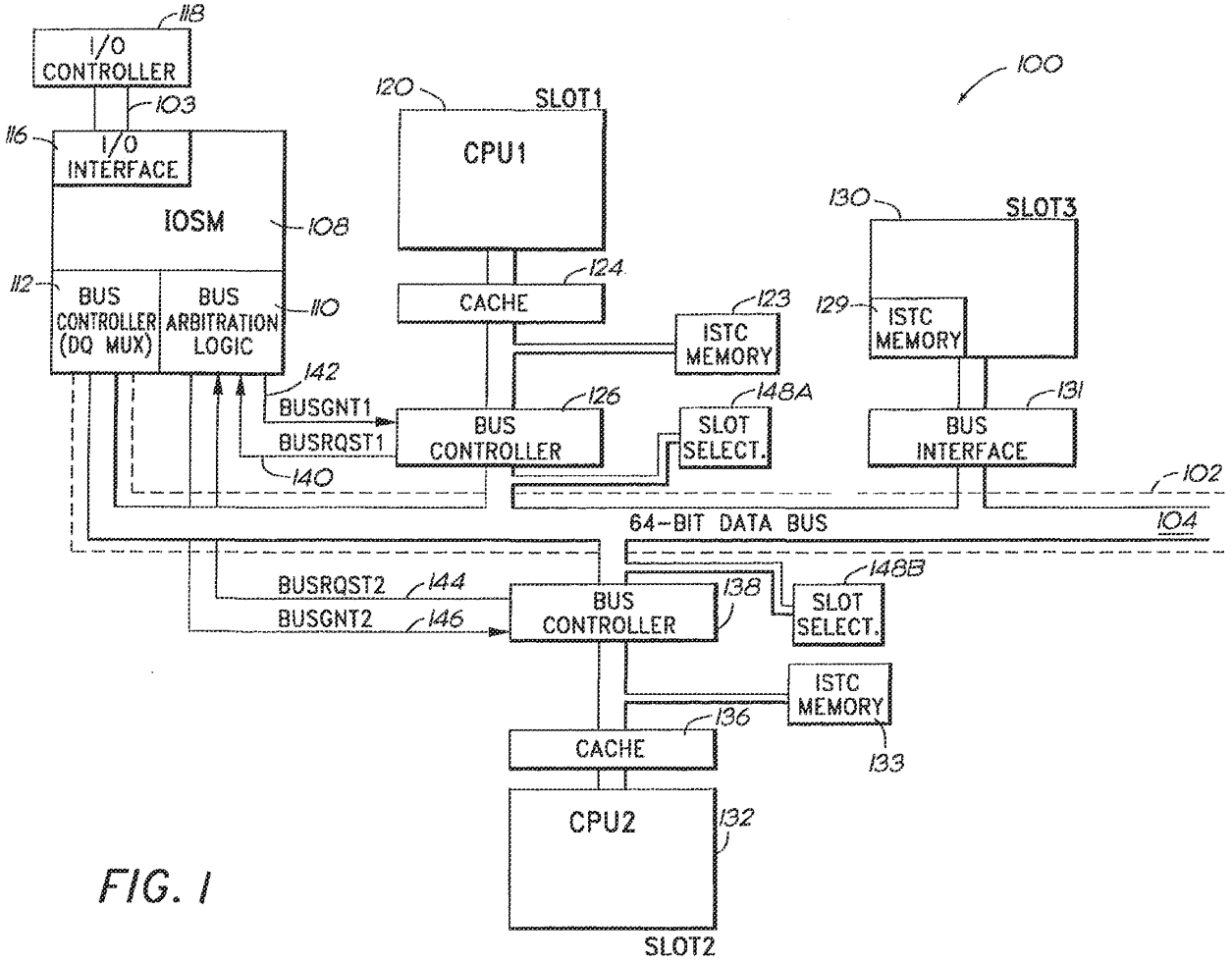

I/O CONTROLLER
118
103
I/O INTERFACE
116
IOSM
108
BUS CONTROLLER (DQ MUX)
BUS ARBITRATION LOGIC
112
110
142
BUSGNT1
BUSRQST1
140
120
SLOT1
CPU1
124
CACHE
123
ISTC MEMORY
126
BUS CONTROLLER
148A
SLOT SELECT.
130
SLOT3
129
ISTC MEMORY
131
BUS INTERFACE
100
102
64-BIT DATA BUS
104
BUSRQST2
144
BUSGNT2
146
BUS CONTROLLER
138
148B
SLOT SELECT.
ISTC MEMORY
133
136
CACHE
CPU2
132
SLOT2

FIG. 1

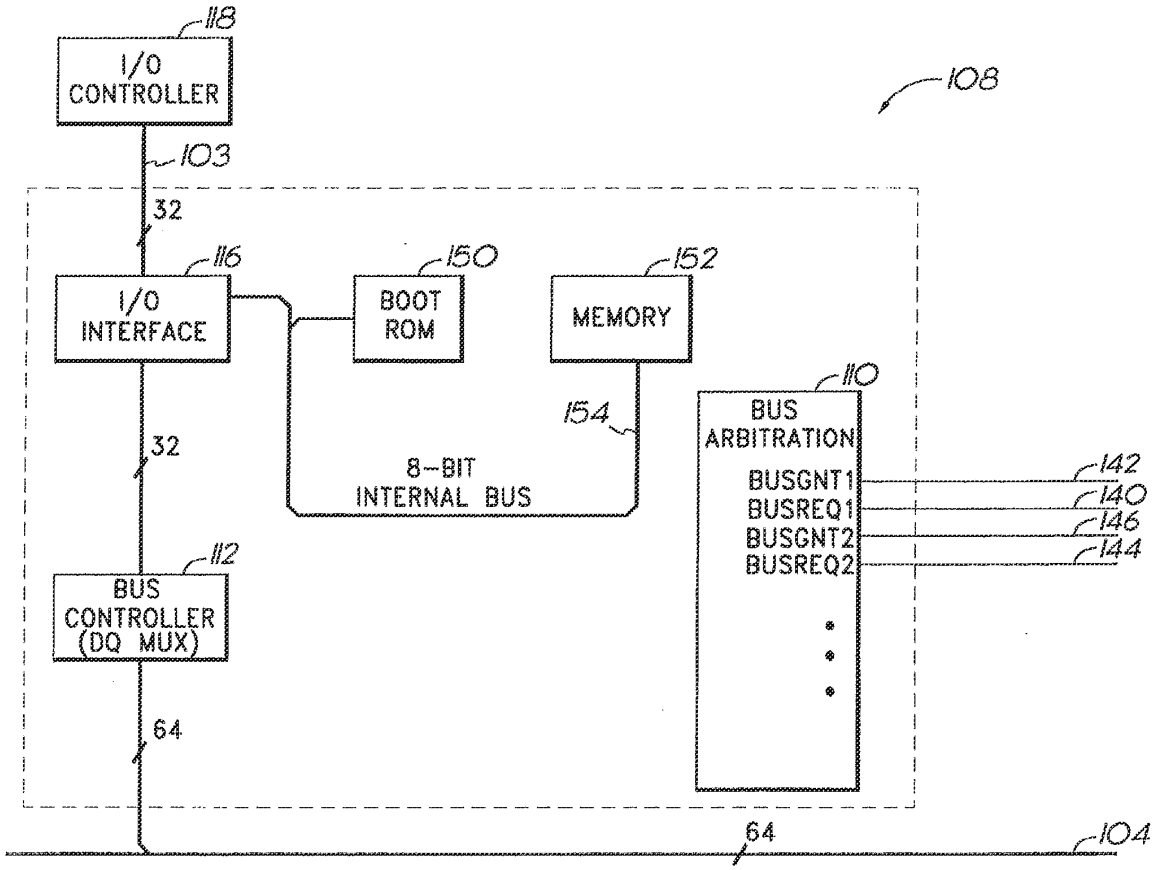

118
I/O CONTROLLER
108
103
32
116
I/O INTERFACE
150
BOOT ROM
152
MEMORY
110
BUS ARBITRATION
154
32
8-BIT INTERNAL BUS
142
BUSGNT1
140
BUSREQ1
146
BUSGNT2
144
BUSREQ2
112
BUS CONTROLLER (DQ MUX)
64
64
104

FIG. 2

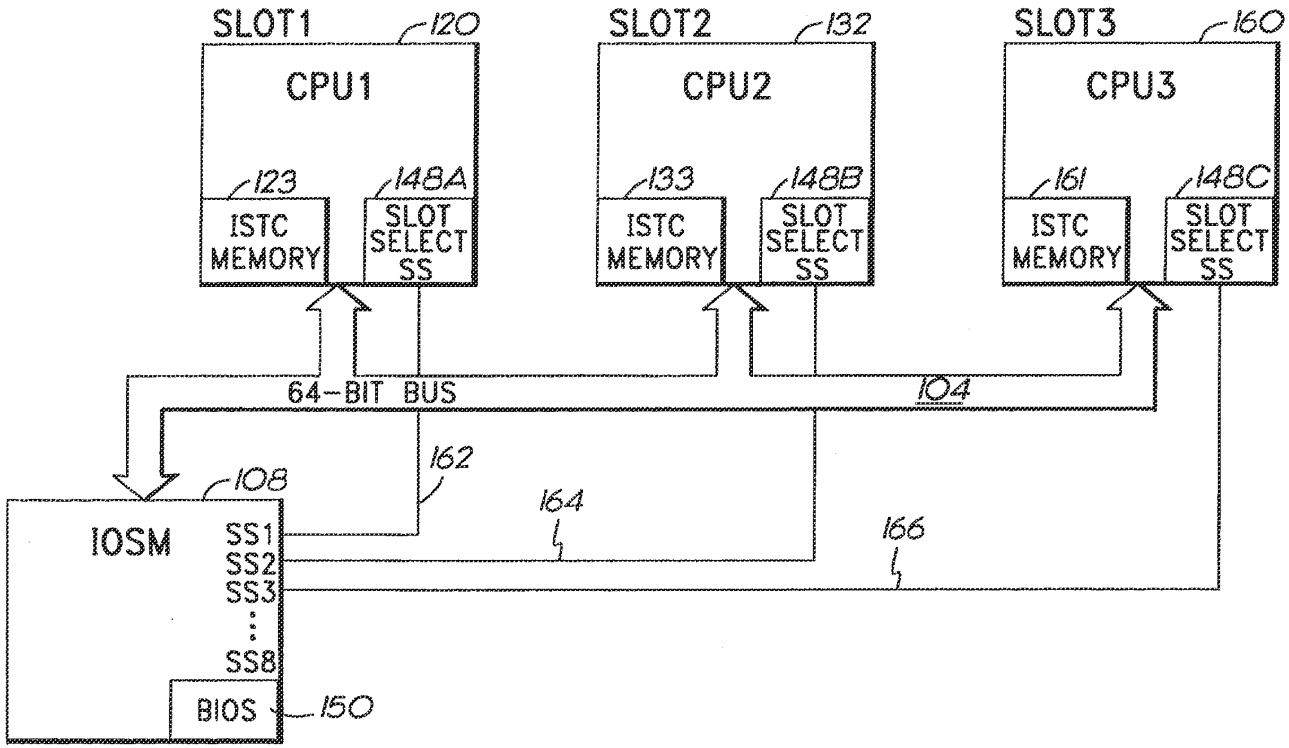

SLOT1
120
CPU1
123
ISTC MEMORY
148A
SLOT SELECT SS
SLOT2
132
CPU2
133
ISTC MEMORY
148B
SLOT SELECT SS
SLOT3
160
CPU3
161
ISTC MEMORY
148C
SLOT SELECT SS
64-BIT BUS
104
162
164
166
108
IOSM
SS1
SS2
SS3
SS8
BIOS
150

FIG. 3

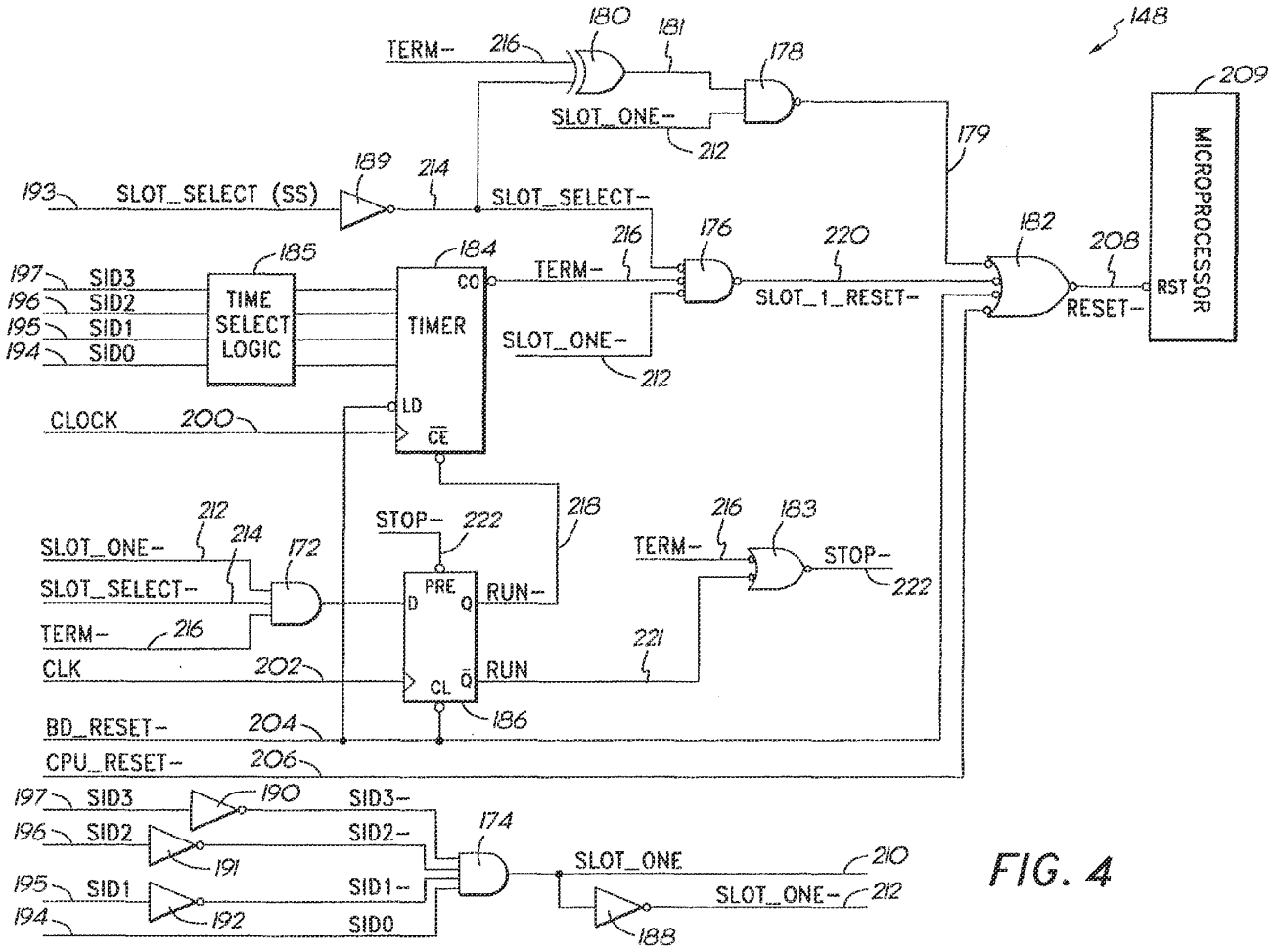

148
TERM-
216
180
181
178
SLOT_ONE-
212
209
MICROPROCESSOR
RST
179
193
SLOT_SELECT (SS)
189
214
SLOT_SELECT-
182
208
RESET-
176
220
SLOT_1_RESET-
197 SID3
196 SID2
195 SID1
194 SID0
185
TIME SELECT LOGIC
184
TIMER
CO
LD
CE
CLOCK
200
SLOT_ONE-
SLOT_SELECT-
TERM-
CLK
202
212
214
172
STOP-
222
218
PRE
D
Q
RUN-
RUN
CL
Q
186
216
183
STOP-
221
BD_RESET-
204
CPU_RESET-
206
197 SID3
196 SID2
195 SID1
194
190
191
192
SID3-
SID2-
SID1-
SID0
174
SLOT_ONE
210
SLOT_ONE-
212
188

FIG. 4

DISTRIBUTED MULTI-PROCESSOR BOOT SYSTEM FOR BOOTING EACH PROCESSOR IN SEQUENCE INCLUDING WATCHDOG TIMER FOR RESETTING EACH CPU IF IT FAILS TO BOOT

This application is a continuation of application Ser. No. 07/721,692, filed Jun. 26, 1991 now abandoned.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to management of initialization operations in multiprocessor computer systems. Specifically, the invention relates to a method and apparatus to automatically control the determination of which processor in the system will boot the system. If the boot processor fails, control automatically shifts to an alternate processor. Additionally, the present invention involves a system wherein each central processing unit includes a corresponding copy of its own boot code. The boot code is transferred to memory for execution by the corresponding processor to carry out initialization operations.

2. Description of the Related Art

The processing requirements of multi-user computer systems for personal and business needs continuously become more demanding. For instance, more complex application programs for use with local area networks are continuously being developed. Moreover, many multi-user systems provide access to more than one operating environment, such as UNIX and DOS, on the same system.

In general, the computers servicing these needs are single processor systems conforming to conventional architectures using a standard input/output I/O bus such as the Extended Industry Standard Architecture (EISA). New and more powerful systems constantly emerge. However, upgrading an old system generally requires investing in substantial hardware modifications or buying a new system.

One solution to the constantly changing power of microprocessors controlling a system is the CUPID architecture designed by AST Research Inc. In the CUPID architecture, the microprocessor based central processing unit (CPU) is not permanently attached to the backplane bus, but is a removable circuit board running at its own speed, asynchronous with the backplane bus operations. Thus, when more power from the microprocessor is desired, a faster CPU can replace the existing CPU.

However, as processing power demands increase, application software and operating systems performance would benefit from an architecture similar to the CUPID architecture, but which has multiprocessor capabilities to provide parallel processing, and to service high numbers of simultaneous users while still retaining high batch performance.

A number of problems hamper the development of such a multiprocessor architecture, such as, determining which processor boots the system, and, if one processor fails to boot the system, enabling another alternate processor to take over boot operations. Moreover, if after one processor boots the system (by executing instructions out of a read-only-memory (ROM)), but another processor utilizes a different instruction set than the boot processor installed in the system, then this other processor will require access to a different boot ROM.

SUMMARY OF THE INVENTION

Advantageously, a system to provide multiprocessor capabilities would overcome these problems and provide expandability capabilities for adding additional memory, I/O controllers, and CPUs. Such a system would provide high performance batch processing and/or high-availability, multi-user services. The system architecture would also be suitable for a wide range of microprocessor-based designs.

The present invention provides a multiprocessor interconnection architecture with a backplane system bus for use with multiple processors. 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. This provides fault tolerance and high availability for the system. Additionally, 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.

One aspect of the invention involves a multiprocessor distributed initialize and self test system for use with a multiprocessor interconnect having a backplane bus with connector slots capable of receiving multiple central processing units, memory modules, and other input/output modules. 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. The system also has centrally accessible boot code executable by the default central processing unit and the alternate central processing unit. A first slot select circuit associated with the default central processing unit which allows the default central processing unit to execute boot operations, is assigned a pre-determined time-out period, selected by a respective slot identification code. If the time-out period elapses before the default central processing unit successfully boots, the first slot select circuit disables the default central processing unit by placing it in a reset state.

A second slot select circuit associated with the alternate central processing unit holds the alternate processor in a reset state until either the default central processing unit successfully boots the system, or a predetermined period of time, selected by a respective slot identification code, elapses indicating that the default central processing unit failed to boot the system.

In one embodiment the default boot central processing unit comprises an INTEL 80486 compatible central processing unit, and the alternate central processing unit comprises an INTEL 80486 compatible central processing unit.

Another aspect of the present invention provides a multiprocessor distributed initialize and self test system for use with a multiprocessor interconnect, wherein the interconnect includes a backplane bus capable of connecting multiple central processing units, memory modules, and other input/output modules. The system has a first central processing unit installed in the backplane

bus, a second central processing unit installed in the backplane bus, and a centrally accessible memory. The system also has centrally accessible boot code accessible and executable by the default central processing unit. A first non-volatile memory associated with the first central processing unit contains initialization and self test code specific to the first central processing unit. In a preferred embodiment, this is transferred to the centrally accessible memory and assembled for execution by the first central processing unit. A second non-volatile memory associated with the second central processing unit contains boot code specific to the second central processing unit. Preferably, this code may be transferred to the centrally accessible memory and assembled for execution by the second central processing unit.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of an exemplary multiprocessor interconnection system with a backplane system bus according to the present invention.

FIG. **2** is a more detailed block diagram of the input-/output services module (IOSM) shown in FIG. **1**.

FIG. **3** is a block diagram of a configuration of a multiprocessor interconnection system with three central processing units and illustrates the slot select interconnections between the IOSM and the slot select circuits of the present invention.

FIG. **4** is a schematic diagram of the slot select logic of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. **1** is a block diagram of a multiprocessor interconnection system **100**. The system **100** of the present embodiment comprises, in general, a backplane system bus **102** with a 64-bit multiple processor data bus **104**, an input/output (I/O) bus **103**, which advantageously comprises a 32-bit Extended Industry Standard Architecture (EISA) bus in the present embodiment, an input-/output services module (IOSM) **108** with a corresponding bus controller **112**, a conventional I/O controller(s) **118**, a first central processing unit (CPU1) **120** with a corresponding cache **124** and bus controller **126**, a memory module **130** with a corresponding bus interface **131**, and a second central processing unit (CPU2) **132** with a corresponding cache **136** and bus controller **138**. Preferably, the I/O controller(s) **118** comprise(s) conventional EISA or ISA compatible controller(s) in the present embodiment. Advantageously, the I/O bus **103** has 8 I/O connectors (e.g., conventional EISA I/O connectors) for the I/O controller(s) **118**, and the backplane system bus **102** for the 64-bit system has eight system connectors along the 64-bit system bus **104**. An additional connector designated to contain the IOSM **108** is located between the backplane system bus **102** and the I/O bus **103**. The IOSM **108** interfaces the 64-bit system bus **104** with the 32-bit I/O bus **103**.

Advantageously, the bus connectors for the 64-bit system are **240** pin METRAL connectors from DuPont, and the bus connector for the IOSM **108** is a 480 pin METRAL connector. The I/O bus connectors in the present embodiment are standard connectors from Burndy Corp., as well known in the art.

The IOSM

The IOSM **108**, as shown in more detail in FIG. **2**, comprises bus arbitration logic **110**, the bus controller **112** which interfaces with the 64-bit system bus **104**, an I/O interface **116**, which interfaces with the 32-bit I/O bus **103**, a central boot read-only-memory (ROM) **150**, a memory **152**, and an internal 8-bit data bus **154** which interconnects the central boot ROM **150**, the memory **152** and the I/O interface **116**. Preferably, the internal 8-bit bus **154** also connects to a real time clock (not shown), a parallel port (not shown) a serial port (not shown), a floppy disk controller (not shown), a keyboard controller (not shown), and a system timer/counter (not shown), as well understood in the art.

The I/O interface **116** advantageously comprises a conventional EISA bus controller chip set, well known in the art, and interfaces the conventional I/O controllers **118** and the internal 8-bit bus **154** with the 64-bit multiple processor system bus **104** via the bus controller **112**. The bus controller **112** interfaces with the system bus **104** using a 32-bit to 64-bit multiplexer/demultiplexer (a double word/quad word multiplexer ('DQ MUX')). The DQ-MUX of the bus controller **112** breaks up 64-bit words into two 32-bit words, or combines pairs of 32-bit words into 64-bit words, as well known in the art. Advantageously, the bus controller **112** also includes a single level storage buffer (not shown).

In the present embodiment, the 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. Moreover, any alternate boot processors are, or emulate an INTEL 80486 or 80386. However, 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. Advantageously, the memory **152** comprises 8 Kbytes of complementary metal oxide semi-conductor (CMOS), static random access memory (SRAM).

The bus arbitration logic **110** accepts a number of individual bus requests from various devices which can become bus masters and provides a signal to grant the bus to the device requesting the bus as well understood in the art. The bus arbitration logic **110** operates on a conventional scheme consisting of two signals carried by a bus request (BUSRQSTn) signal line and a bus grant (BUSGNTn) signal line, one of each for every device which can become a bus master. The bus arbitration logic **110** communicates with bus controllers for these operations. For example, the bus controller **126** for the CPU1 **120** (FIG. **1**) requests the bus by activating a BUSRQST1 signal line **140**, and the bus arbitration logic **110** responds with an active signal on the a BUSGNT1 signal line **142** to grant the bus to the bus controller **126**. Similarly, the bus controller **138** for the CPU2 **132** requests the bus by activating the BUSRQST2 signal line **144**, and the bus arbitration logic **110** grants the bus to the bus controller **138** by activating the BUSGNT2 signal line **146**. The I/O interface **116** may also obtain control of the bus, on behalf of an I/O controller **118** requesting to be a bus master, with corresponding BUSRQST0 and BUSGNT0 signal lines (not shown).

Devices installed on the system bus **102** advantageously accept a 64-bit transfer even though the actual device may not utilize a full 64-bit data bus. For instance, if the CPU1 **120** is based upon an INTEL 80486 which uses a 32-bit data bus, the bus controller **126** accepts a 64-bit transfer from the system bus **102**, places

this data into the cache **124** which provides a 32-bit interface to the CPU1 **120**.

The CPU Modules

The CPU1 module **120** could be any microprocessor chip set running at any speed. In the present embodiment, at least one CPU is based upon an INTEL 80486 or compatible microprocessor. Accordingly, throughout the remainder of this description, references to the CPU1 **120** assume an INTEL 80486-based CPU with supporting resources and on-board crystal oscillators for independent timing. Other CPUs in the system need not be 080486-based as explained in more detail herein.

CPUs installed in the bus **104** may have independent asynchronous timing with respect to the bus **104**.

The 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**, and further comprises a slot select circuit **148**A which is explained in detail below. In one alternative embodiment of the invention, the non-volatile memory **123** comprises a programmable read-only-memory (PROM), as well known in the art. In the present embodiment, the cache **124** for the CPU1 **120** is a 256-Kbyte, two-way, set-associative, write-back cache with a 32-byte line size (4 bus cycles×8 bytes). The cache **124** interfaces the asynchronous CPU1 **120** with the synchronous 64-bit bus **104** via the bus controller **126** which responds to signals on the BUSRQST1 signal line **140** and the BUSGNT1 signal line **142** as explained. The cache **124** supports write-back and the conventional Modified, Exclusive, Shared, Invalid (MESI) protocol to maintain cache coherency for the multiprocessor system **100**. The cache **124** has a corresponding 64-bit interface (not shown) for the 64-bit bus **104** and a 32-bit interface (not shown) with the 80486 processor. When the cache **124**, or any other cache, generates a write-back cycle, it asserts an active low signal on the write-back start (WBSTRT-) control line (not shown) to indicate the beginning of a write-back cycle from the cache, as well understood in the art.

The CPU2 **132** is similar to the CPU1 **120** except that the CPU2 **132** need not be an 80486 or 80386 based CPU. The CPU2 **132** includes an non-volatile ISTC memory **133** which contains configuration information and ISTC for the microprocessor of the CPU2 **132**. The CPU2 **132** also comprises a slot select circuit **148**B, and has a corresponding bus controller **138** and a cache **136** similar to those associated with CPU1 **120**. Further CPUs may also be added to the system and need not comprise INTEL 80486 or 80386 based CPUs.

The Memory Modules

In the present embodiment, the memory module **130** accepts 64-bit transfers. However, memory modules need not be capable of accepting full 64-bit transfers. Advantageously, the memory **130** comprises 40-bit single-in-line memory modules (SIMMs) which could be constructed from 1-Meg-by-4 or 4-Meg-by-4 dynamic random access memories (DRAMs). Toshiba's THM401020SG-80 is an exemplary 10-chip SIMM appropriate for use in the present system. The memory **130** supports 64 megabytes (Mbytes) (per module) of RAM with 1-Meg-by-4 DRAM based SIMMs, or 256 megabytes (per module) with 4-Meg-by-4 DRAM based SIMMs. The present embodiment allows up to four 256 Mbyte memory modules to be installed in the system. The memory module **130** also includes error correction code (ECC) capability for reliability. However, to eliminate the read-modify-write cycle caused by 32-bit operations on a 64-bit ECC memory, ECC is performed on a 32-bit basis and not on a 64-bit basis.

The memory module **130** also comprises a DRAM controller (not shown) which provides conventional row address select (RAS), column address select (CAS), hidden refresh, address multiplexing, page mode and burst mode support. Accordingly, the DRAM controller comprises DRAM RAS/CAS address multiplexers, RAS/CAS generation logic, refresh counters, page-mode/page-hit logic and DRAM address drivers, all well understood in the art. The DRAM controller derives memory timing from an external clock signal carried on the system bus **104**, and therefore, runs synchronously with the system bus **104**.

In the present embodiment, the memory **130** also includes a corresponding bus interface **131** to the 64-bit bus **104**. Advantageously, the bus interface **131** comprises a 8-level deep by 64-bit wide register. Parity generation and checking for the bus interface **131** is performed on a byte-by-byte basis on the 64-bit bus side of the register.

Additionally, the memory has an associated non-volatile ISTC memory **129** which contains the ISTC for the memory **130**.

FIG. **3** illustrates an embodiment with three CPUs: the CPU1 **120** installed in slot **1**, the CPU2 **132** installed in slot **2**, and a CPU3 **160** installed in slot **3** with a corresponding ISTC memory **161** and a slot select circuit **148**C. FIG. **3** further illustrates the connections from the slot select circuits **148**A, **148**B, and **148**C to the IOSM **108**. As seen in FIG. **3**, the slot select circuit **148**A for the CPU1 **120** connects to the IOSM **108** with a slot select signal line (SS1) **162**. Similarly the slot select circuits **148**B and **148**C connect to the IOSM **108** with slot select signal lines (SS2 and SS3) **164** and **166** respectively. Any other slot select circuits on CPUs installed in the system have a corresponding slot select signal line from the IOSM **108**. The SS1, SS2 and SS3 signal lines **162**, **164**, and **166** respectively, connect to a respective SLOT_SELECT signal line **193** (FIG. **4**) for each slot select circuit **148**A, **148**B and **148**C, respectively. The signals on the SSn signal lines are controlled by a memory or I/O mapped slot select register (not shown) in the IOSM **108**. Thus, the CPUs in the system can access or change the signals on these lines by executing a read or write to the memory or I/O mapped address for the register. These SSn signal lines are used during boot operations as explained herein.

The Slot Select Circuits

FIG. **4** illustrates the logic **148** corresponding to the slot select circuits **148**A and **148**B and slot select circuits for any other CPU installed in the system. The slot select logic **148** comprises AND gates **172**,**174**, **176**, a NAND gate **178**, an EXOR gate **180**, OR gates **182**,**183**, a timer **184**, time select logic **185**, a latch **186**, and inverters **188**, **189**, **190**, **191**, and **192**. The timing circuit responds to signals on a SLOT_SELECT signal line **193**, SLOT_ID (SID0, SID1, SID2, SID3) signal lines **194**–**197**, a CLOCK signal line **200**, a CLK signal line **202**, a BD_RESET- signal line **204**, and a CPU_RESET- signal line **206**. The slot select circuit **148** provides a signal on a RESET- signal line **208** connected to the reset pin of the microprocessor **209** on the CPU board. The slot select circuit **148** further responds to a number of internal signal lines, such as a SLOT_ONE signal line **210**, a SLOT_ONE- signal line **212**, a SLOT_SELECT- signal line **214**, a TERM- signal line **216**, a RUN- signal line **218**, a SLOT_1_RESET- signal line

220, a RUN signal line 221, and a STOP- signal line 222. The functions of these signal lines are further explained herein.

Distributed Initialize Function

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.

Another potential difficulty with the multiprocessor system 100 is that a central system boot ROM 150 should be accessible by all CPUs for initialization operations. However, the boot code for each CPU in the system may differ. Thus, changes to the system boot code are often needed when the processor type changes, when problems are found, or when the memory, cache and I/O features are enhanced. Even if boot code for numerous processors were included in the central boot ROM 150, as new developments emerge, the boot ROM would need updating.

The present invention solves this problem by providing memories (e.g., ISTC memory 123, ISTC memory 133, and ISTC memory 129, etc.) for each CPU, I/O board, and memory module. These ISTC memories contain configuration, and initialization and self-test code (ISTC) (commonly referred to as the power-on self test (POST) and INITIALIZE portion of the boot code in the art) specific to the respective CPU, I/O board or memory module. 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. For instance the memory 123 contains configuration information and ISTC for the CPU1 120 and the memory 133 contains configuration information and ISTC for CPU2 132, and the memory 129 contains configuration information and ISTC for testing and initializing the memory module 130.

Advantageously, the PROMs 123, 129, 133, and any other ISTC memories are memory mapped, and preferably addressable in upper memory space (e.g., above the 2 gigabyte boundary). The precise address location mapped to each ISTC memory on a circuit board depends upon the slot in which the board is installed. For instance, advantageously, the memory 123 on CPU1 120 is addressable beginning at address location D100,0000 hexadecimal (hex), if CPU1 120 is installed in slot 1, and the memory 133 on CPU2 132 is addressable beginning at address location D200,0000 hex, if CPU2 132 is installed in slot 2 on the bus.

However, if each circuit board ISTC memory contained boot code in executable form (e.g., 32-bit words for INTEL 80486 based CPUs) and was directly accessible for execution of the code in the ISTC memory, then each board would need data shift logic to interface the ISTC memory with the 64-bit system bus 104. This logic would add to the complexity, and therefore increase the cost, of the system. Therefore, in the present embodiment, the ISTC memories are 8 bits (1 byte) wide and are not directly accessible for execution of the code (i.e. the ISTC memories are execution inaccessible). In the present embodiment, the board specific code held in the ISTC memories on the circuit boards is accessible on a byte-by-byte, non-executable basis. Accordingly, in the present embodiment, the ISTC memories interface with the least significant byte of the system bus 104.

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. The non-executable code is assembled by transferring the code, byte-by-byte, from the byte-wide memory into 32-bit words in the memory 130.

The assembly of the information in the ISTC memories into system memory 130 preferably involves moving the information, byte-by-byte in the present embodiment, into a register until 4 bytes are assembled in the register (or 8 bytes if the system memory is 8 bytes wide). Then the contents of the register are preferably moved to the system memory 130 in one move operation. Alternatively, the ISTC memories could be assembled directly on a byte-by-byte basis into the system memory, but this would require more time because of the increase in system memory accesses.

As well understood in the art, conventional addressing allows byte addressing of the bytes on the system bus 104. For instance, the INTEL 80386 and 80486 provide address lines starting at address line A2 (the third significant bit in the address) and provide four byte enable lines for selecting which bytes are active during a bus transaction. As explained above, in the present embodiment, the ISTC memories are one byte wide and interface to the least significant byte of the system bus 104. Therefore, the byte enable lines select the least significant byte as active during transfers from the ISTC memories to the memory 130 and any data on the remaining seven bytes of the system bus 104 are ignored.

However, in an alternative embodiment, the ISTC memories may be wider that one byte. Preferably, the width of the ISTC memory interface to the system bus 104 is contained along with the configuration information in the ISTC memory. Thus, the system can determine how to transfer the information from an ISTC memory to the memory 130.

In order to properly access the ISTC memories, the address lines of these ISTC memories connect to system address lines A3 and above (where A0 is the least significant bit). In this manner, incrementing the address on the system address lines by eight, is detected by the ISTC memories as an increment of one on address line A3, and the byte enable line are used to select which bytes are active on the system bus 104.

For instance, if the ISTC memory interface to the system bus 104 is two bytes wide, then the software which transfers the information from an ISTC memory to the system memory 130 for execution preferably transfers two bytes at a time instead of a single byte per transfer. The byte enable lines are used to select the two least significant bytes as active on the system bus 104 with each transfer.

Once the ISTC is assembled in memory, it is executed from memory by the corresponding CPU, or executed by the boot CPU if the ISTC corresponds to a memory module or I/O circuit board.

However, until a processor is controlling the system, the non-executable code from the ISTC memories is inaccessible by a CPU attempting to boot the system. Therefore, a central boot ROM is still needed that is accessible to any CPU which can boot the system. In

the present embodiment, a central system boot ROM **150** accessible by all processors contains INTEL 80×86 executable boot code in addition to conventional DOS BIOS CALL code which may be used by the operating system, as well understood in the art. Accordingly, in the present embodiment, the boot CPU is based around an INTEL 80×86 compatible processor (e.g., an INTEL 80486). Because the only board unique to the entire system is the IOSM **108**, the boot ROM **150** (FIG. 2) provides the central system boot code and contains the conventional DOS BIOS code.

Upon power-up reset, the default boot CPU (assume this is CPU1 **120** for present discussion) executes code from the system boot ROM **150** (this code is for an INTEL 80×86 which is why the boot CPU is an 80×86 based CPU in the present embodiment). 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). If a slot contains a CPU, then the boot CPU will receive information about the type of CPU installed in the slot.

The CPU1 **120** then tests the first megabyte of memory in the system and sets a 'check-in-word' in the CMOS portion of the memory **152**. Once the integrity of the first megabyte of memory is established, the CPU1 **120** copies and assembles its own ISTC byte-by-byte from its ISTC memory **123** into memory in 32-bit word executable format and begins executing its own ISTC. Once the CPU1 **120** is initialized, it checks itself 'in' by writing to the check-in-word in the CMOS portion of the memory **152**.

After the CPU1 **120** is checked 'in', it copies the ISTC from the ISTC memory **129** to the memory **130**, executes the ISTC, and creates a memory configured/-memory good table in the CMOS portion of the memory **152**.

Once the memory module **130** is configured (and any other memory modules installed in the system are configured), the CPU1 **120** transfers and executes the ISTC from appropriate ISTC memories on the I/O controllers that are installed on the system bus **104**. Then, the CPU1 **120** transfers the next CPU's (assume CPU2 **132** for present discussion) ISTC from the next CPUs's associated ISTC memory (memory **133** for the CPU2 **132**) to the memory **130** and allows the CPU2 **132** to exit a reset state and execute its own ISTC. After successful completion of the initialization and self test, the CPU2 **132** checks 'in.'

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.

If one or more CPUs in the system are not INTEL 80×86 based CPUs, then the ISTC memories for these CPUs would hold the entire boot code for these CPUs, not just the configuration information and ISTC.

A time limit is provided for each stage of the boot process so that if the time allotted for a device to initialize and check-in is exceeded, that device is assumed to be non-functional and is disabled. The remaining hardware may then attempt to complete the boot process. When multiple CPUs are installed in the system, if the default boot processor exceeds the time allotted for it to boot (it fails), an alternative boot processor can take over boot operations as explained below.

Alternative Boot Processor Functions

In a multiprocessor environment, one critical determination is which processor will boot the system. Moreover, if one processor fails, provision for disabling the non-functional processor and allowing another alternative processor to boot the system enhances the availability of the system. With multiple CPUs connected to the same bus, the problem is automatic boot control of the system. The reason for the difficulty is that until at least one CPU is operating the system, no microprocessor control is available to supervise the other processors.

Accordingly, in the present invention, each CPU on the bus includes slot select logic **148** (FIG. 4). In general, at power-up or system reset the slot select logic **148** for each CPU, is loaded with a time-out count dependant upon the slot in which the board is installed. Each slot connector has four hard connections which provide signals for a 4-bit identification code (slot ID) on signal lines **194–197** for a board installed in the slot. The signals on the SLOT_ID signal lines **194–197** select a pre-determined time-out period from the time select logic **185** corresponding to the time-out period assigned to the slot in which the slot select logic **148** is installed. The time-out period for the slot select logic **148A** is a lower value than the time-out period for the slot select logic **148B** and so forth. Advantageously, the time select logic **185** may comprise a programmable logic array or a read-only-memory with time-out values programmed corresponding to each slot. In general, the slot select logic **148** on the default boot CPU allows this CPU to boot. The remainder of the slot select logic circuits **148**, installed in other slots, hold the associated CPUs in reset until the predetermined time-out period has expired for the associated CPU, or until the boot CPU successfully boots the system and then activates the CPUs in other slots.

If the default boot CPU does not successfully boot and disable its own slot select logic **148** before it times-out, it enters a reset state. The next CPU (based upon a slot identification) then times-out and exits the reset state and attempts to boot the system as an alternate boot processor. The alternate boot processors are the same type processor as the default boot processor so that they can execute the boot code in the central boot ROM **150** (e.g., in a preferred embodiment, the alternate boot processor(s) is an 80486 compatible). As part of the alternate processor boot operations, it disables CPUs installed in slots with a lower number than its own that have not checked 'in.' If the alternate boot CPU fails to boot, then another CPU (of the same processor type) has a chance to boot the system and so forth.

If after all the CPUs installed in the system have booted, but one or more have failed to check-in, the boot CPU disables any CPUs which failed to check-in.

The slot select logic **148** is described in more detail in reference to the CPU1 **120** installed in slot **1** as the default boot processor and the CPU2 **132** installed in slot **2** as the alternate boot processor. The slot select circuit **148A** for CPU1 **120** is identical to the slot select circuit **148B** for CPU2 **132**, however, the circuits **148A**, **148B** provide slightly different functions depending on in which slot each circuit is installed.

Installed in Slot 1

In the present embodiment, slot **1** contains the default boot CPU. Thus, the CPU1 **120** in slot **1** is first allowed to boot the system. The general function of the slot select logic **148A** installed in slot **1** is to allow the CPU1 **120** to perform system boot operations. However, if the

CPU1 120 fails to boot properly, the slot select logic 148A disables CPU1 120 and control automatically shifts to the next CPU in the system—in the depicted embodiment, the CPU2 132 in slot 2—as described in more detail below.

The SLOT_ONE signal line 210, the SLOT_SELECT signal line 193 and a TERM- signal line 216 carry the main control signals for the slot select logic 148A and the resulting signal from the logic 148A is a reset signal on the RESET-signal line 208 (FIG. 4). The values of the reset signal for varying combinations of signal levels on the SLOT_SELECT signal line 193, the TERM- signal line 216 and the SLOT_ONE signal line 210 for circuit boards installed in slot 1 are shown in Table 1 below. Corresponding values for circuit boards installed in slots 2–8 are shown in Table 2 below.

TABLE 1

	Slot 1			
	SLOT_ONE	SLOT_SELECT	TERM—	RESET—
BOARD RESET	1	1	1	0
FAIL	1	1	0	0
RUN	1	0	0	1
SHUT DOWN	1	0	0	0

As seen in row 1 of Table 1, at board reset, the SLOT_SELECT line 193 and the TERM- signal line 216 from the timer 184 are both at a high level, regardless of the slot in which a circuit board is installed. For a circuit board installed in slot 1, the SLOT_ONE signal line 210 is also high because the AND gate 174 in conjunction with the inverters 190, 191, and 192 decodes the slot ID on signal lines 194–197 to produce an active signal on the SLOT_ONE signal line 210. This signal is inverted by the inverter 188 to produce a low signal on the SLOT_ONE- signal line 212. The high signal on the SLOT_SELECT signal line 193 is inverted by the inverter 190 to produce a low signal on a SLOT_SELECT- signal line 214.

The function of the SLOT_SELECT line 193 changes for the slot select circuit 148A in slot 1 depending on the output of the timer. While the timer is running, the signal on the SLOT_SELECT signal line 193 has no effect on the slot select circuit 148A. When the timer stops running, then a low signal on the SLOT_SELECT signal line 193 indicates that the slot is enabled, while a high signal disables the CPU1 120 in slot 1.

As long as the BD_RESET- signal line 204 is active, all circuit boards in the system are held in reset because the active signal on the BD_RESET- signal line 204 activates the OR gate 182 connected to the RESET- signal line 208 which provides a reset signal to the microprocessor 209 (FIG. 4).

The BD_RESET- signal line 204 becoming active also loads the time-out value from the time select logic 185 into the timer 184, and clears the latch 186 which produces a low signal on the RUN- signal line 218 and a high signal on the RUN signal line 221. The low signal on the RUN- signal line 218 enables the timer 184 to start counting down from the value provided by the time select logic 185 once the BD_RESET- signal line 204 becomes inactive. The timer 184 continues to run until the latch 186 produces a high signal on the RUN- signal line 218. This occurs if the AND gate 172 is activated or the STOP- signal line 222, which is connected to the preset input of the latch 186, becomes active, as further explained herein.

Until the timer, which is counting down, reaches the terminal count of zero, the signal on the TERM- signal line 216 remains high. With the signal on the TERM- signal line 216 at a high level, the AND gate 176 also remains inactive. However, because the SLOT_SELECT- signal line 214 and the SLOT_ONE- signal line carry low signals at this point, if the timer 184 reaches zero before the signal on the SLOT_SELECT-signal line 214 becomes high (the CPU1 120 failed to boot), the signal on the TERM- signal line 216, which becomes active when the timer reaches zero, activates the AND gate 176 to provide a low signal on the SLOT_1_RESET- signal line 220.

The low signal on the SLOT_1_RESET- signal line 220 activates the OR gate 182 to produce a low signal on the RESET- signal line 208 and hold the microprocessor 209 in reset (i.e. to disable the circuit board).

The active signal on the TERM- signal line 216 activates the OR gate 183 to produce an active signal on the STOP-signal line 222. The active signal on the STOP- signal line 222 stops the timer 184 by activating the preset connection to the latch 186. Thus, the latch 186 drives the RUN- signal line 218 high to stop the timer 184 from operating and also drives the RUN signal line 221 low which also activates the OR gate 183 to produce the active signal on the STOP- signal line 222. Advantageously, this prevents the latch 186 from activating the RUN- signal line 218 until another board reset is signalled on the BD_RESET- signal line 204. The signals stabilize as indicated in the FAIL row in Table 1.

Because the signal on the SLOT_ONE- signal line 212 is low, the NAND gate 178 remains inactive with a high signal on a signal line 179. Thus, the signal from the NAND gate 178 does not effect operation of the OR gate 182 if the circuit board is installed in slot 1. Accordingly, as long as the BD_RESET- signal line 204 and the CPU_RESET- signal line 206 also remain inactive, the signal from AND gate 176 on the SLOT_1_RESET- signal line 220 controls the OR gate 182.

If the CPU1 120 boots properly, it changes the level of SS1 signal line 162 from the IOSM 108 by writing to the memory or I/O mapped slot select register in the IOSM 108 as previously explained. The SS1 signal line 162 connects to the SLOT_SELECT (SS) signal line 193 for slot select circuit 148A. The change in the level on the SS1 signal line results in a low level on the SLOT_SELECT signal line 193 and a high level on the SLOT_SELECT- signal line 214. The high level on the SLOT_SELECT- signal line 214 prevents the AND gate 176 from becoming active when the TERM- signal line 216 becomes active upon the timer 184 reaching zero. Thus, when the CPU1 120 boots properly, the SLOT_1_RESET- signal line 220, the OR gate 182, and the RESET- signal line 208 remain inactive even if the TERM- signal line 216 becomes active.

Even after the CPU1 120 in slot 1 has booted, the timer continues to time-out to activate the TERM- signal line 216. The timer stops at zero because the

active signal on the TERM- signal line 216 activates the OR gate 183 to produce an active signal on the STOP- signal line 222. The active signal line the STOP- signal line 222 presets the latch 186 to produce a high signal on the RUN- signal line 218 which stops the timer 184 from operating, and leaves the timer 184 with the TERM- signal line 216 active. Accordingly, when the CPU1 in slot 1 properly boots, the slot select logic becomes stable with AND gate 176 and the OR gate 182 inactive so that signal line 208 is inactive and the processor is not held in reset. The stable signals are indicated by the 'RUN' row in Table 1.

From the state shown in the RUN row of Table 1, the IOSM 108 can reset the CPU1 120 by activating the corresponding SS1 signal line 162 which connects to the SLOT_SELECT signal line 193 of the slot select circuit 148A. An active signal on the SLOT_SELECT signal line 193 resets the CPU1 120 because the corresponding active signal on the SLOT_SELECT- signal line 214 activates the AND gate 174 (the TERM- signal line 216 and the SLOT_ONE- signal line 212 are also active) to activate the SLOT_1_RESET- signal line 220 which activates the OR gate 182 as previously explained. The signals stabilize as indicated in the 'Shut Down' row of Table 1.

With the CPU1 120 installed in slot 1, the AND gate 172 remains inactive and does not effect operation of the slot select logic 148A because the SLOT_ONE- signal line 212 is low which prevents the AND gate 172 from being activated.

Installed in Slot 2-8

The general function of the slot select logic 148 installed in any of the slots 2-8 is to hold the corresponding CPU in reset until CPUs in previous slots have had sufficient (pre-determined) time to boot. In other words, for the CPU2 132 installed in slot 2, the slot select logic 148B holds the CPU 132 in a reset state until sufficient time has elapsed for the CPU1 120 to boot. If the CPU1 120 does not complete boot operations within the time allotted, then the CPU2 132 takes over default boot operations as an alternative boot processor instead of CPU1 120. If the CPU1 120 does boot properly, then the CPU2 132 checks in as an additional processor rather than taking over as a boot processor.

Table 2 below indicates the value of the signals on the RESET- signal line 208 for various combinations of signals on the TERM- signal line 216, the SLOT_SELECT signal line 193 and the SLOT_ONE signal line 210. The function of the slot select logic 148 for slots 2-8 is explained with reference to CPU2 132 in slot 2 of the bus 104.

TABLE 2

	Slots 2-8			
	SLOT_ONE	SLOT_SELECT	TERM—	RESET—
BD_RESET	0	1	1	0
ALT BOOT	0	1	0	1
RUN	0	0	1	1
FAIL	0	0	0	0

Upon a board reset, the active pulse on the BD_RESET-signal line 204 loads the timer 184 with the value from the time select logic 185 corresponding to the slot 2 four-bit slot ID on lines 194-197 and the timer 184 begins to operate (as explained for slot 1). The slot ID value loaded in the timer 184 is greater than the value for slot 1 and is pre-selected to allow enough time for the CPU1 120 in slot 1 to execute initialization operations.

At board reset, the SLOT_SELECT signal line 193 is high, as seen in Table 2, which is inverted by the inverter 189 to a low level on the SLOT_SELECT- signal line 214. The TERM-signal line 216 is high because the timer 184 is active. The high signal on the TERM- signal line 216 and the low level on the SLOT_SELECT- signal line 214 activate the EXOR gate 180 which produces a high level on a signal line 181 connected to the NAND gate 178. The high level signals on the signal line 181 and the SLOT_ONE- signal line 212 combine to activate the NAND gate 178 which produces an active signal on the signal line 179. The active signal on signal line 179 activates the OR gate 182 to produce an active signal on the RESET- signal line 208 to hold the CPU2 132 in reset.

For CPU2 132 in slot 2, the SLOT_ONE signal line is inactive because the AND gate 174 in conjunction with the inverters 190, 191, and 192 does not detect the correct slot ID on signal lines 194-197 to indicate that the CPU2 132 is installed in slot 1. Thus, the SLOT_ONE- signal line 212 is high because the inactive signal on the SLOT_ONE signal line 210 is inverted by the inverter 188 to produce a high signal on the SLOT_ONE- signal line 212.

If the CPU1 120 in slot 1 boots properly before the timer 184 for the CPU2 132 in slot 2 times out, the timer 184 for the CPU2 132 is prevented from counting and the RESET-signal line 208 becomes inactive, allowing the CPU2 132 to boot. The signals stabilize as seen in the 'RUN' row of Table 2.

The timer stops because when the CPU1 120 boots properly, the CPU1 120 changes the level of the SS2 signal line 164 in the IOSM 108 which is connected to the SLOT_SELECT line 193 for the slot select circuit 148B. This results in a high signal on the SLOT_SELECT- signal line 214. The high signal on the SLOT_SELECT- signal line 214, the high signal on the SLOT_ONE- signal line 212, and the high signal on the TERM- signal line 216 combine to activate the AND gate 172 to produce a high signal for the input to the latch 186. A pulse on the CLK signal line 202 causes the high signal to propagate through the latch 186 to the RUN- signal line 218 to stop the timer 184. This results in a low signal on the RUN signal line 221, which causes the OR gate 183 to activate the STOP- signal line 222. The active signal on the STOP-signal line 221 presets the latch 186 to hold the RUN- signal line 218 high and the RUN signal line 221 low. Accordingly, the signal on the TERM- signal line 216 remains high because the timer did not time-out.

The RESET- signal line 208 becomes inactive because the low signal on the SLOT_SELECT signal line 193 is inverted by the inverter 189 to a high level on the SLOT_SELECT- signal line 214. The high signals on the TERM- signal line 216 and the SLOT_SELECT- signal line 214 combine to cause the EXOR gate 180 to become inactive and produce an inactive low signal on

the signal line 181. The inactive signal on the signal line 181 from the EXOR gate 180 inactivates the NAND gate 178 which produces an inactive signal on the signal line 179 to inactivate the NOR gate 182. The AND gate 182 produces an inactive signal on the RESET- signal line 208 which allows the microprocessor of the CPU2 132 to exit the reset state and initialize itself as explained above.

If the CPU1 120 in slot 1 does not boot (proper boot is indicated by a change in the level of the SLOT_SELECT signal line 193 from high to low) before the timer 184 in slot 2 times-out, then the CPU2 132 exits the reset state and attempts to boot the system as an alternate boot CPU.

The TERM- signal line 216 becomes active when the timer 184 times-out. The SLOT_SELECT signal line 193 remains high because the CPU1 132 (which failed to boot) did not change the level of this signal. The active (low) signal on the TERM- signal line 216 and low signal on the SLOT_SELECT-signal line 214 deactivate the EXOR gate 180 which produces an inactive signal on the signal line 181 to deactivate the NAND gate 178. The inactive NAND gate 178 provides an inactive signal on the signal line 179 which allows the 0R gate 182 to become inactive and produce an inactive signal on the RESET- signal line 208. Accordingly, the CPU2 132 exits from the reset state and takes over boot operations in place of the CPU1 120. By checking the level of the SLOT_SELECT signal line 193, the CPU2 132 determines that the CPU1 132 failed to boot and that it (the CPU2 132) is now the boot CPU. If the CPU2 132 boots properly, then the signals stabilize as indicated in the 'ALT BOOT' row of Table 2.

The functional level of the signal on the SLOT_SELECT signal line 193 changes depending upon the output of the timer 184 for slot select logic 148B installed in slot 2 (as well as for slot select logic 148 installed in all other slots 3–8). While the timer is running, a low signal on the SLOT_SELECT signal 193 freezes the timer and indicates that slot 2 is enabled and a high on the SLOT_SELECT signal line 193 disables the slot. When the timer 184 stops running, a low signal on the SLOT_SELECT signal line 193 indicates that the slot 2 is disabled, and a high signal indicates that the slot is enabled. The change in level on the SLOT_SELECT signal line 193 from high to low if the CPU2 132 also failed to boot occurs because the timer 184 for CPU3 160 in slot 3 will time-out and begin to boot the system. When the CPU3 160 begins boot operations, it disables the CPU2 132 by changing the level of the SS2 signal line 164 in the IOSM 108 to a low level.

Thus, regardless of which CPU boots the system, the boot CPU can let other CPUs time out, at which time they will begin to initialize. Otherwise, the boot CPU can specifically enable them before they time-out by use of the SLOT_SELECT signal line 193, or the boot CPU can disable the timer on any or all CPUs and hold the corresponding CPUs in reset by toggling the corresponding SLOT_SELECT line 193.

Accordingly, after the CPU1 120 in slot 1 has failed to boot properly, and the CPU2 132 in slot 2 takes over boot operation, the timer 184 of slot select circuit 148B has stopped operating and the signal on the TERM- signal line 216 is active. The level on the SLOT_SELECT line 193 remains high as explained above which indicates at this point that the CPU2 132 is enabled. However, if the level on the SLOT_SELECT signal line 193 now changes to low (driven by the IOSM 108 with the SS2 signal line 164), indicating that the CPU2 132 failed to boot the system within the time allotted for the CPU2 132 to complete boot operations, then the CPU2 132 enters a reset state because the active signal on the TERM- signal line 216 combines with the now high signal on the SLOT_SELECT- signal line 214 in the EXOR gate 180 to produce an active signal on the signal line 181. The active signal on the signal line 181 combines with the high signal on the SLOT_ONE- signal line 212 to activate the NAND gate 178 which in turn provides an active signal on the signal line 179 to activate the OR gate 182 to hold the microprocessor of the CPU2 132 in reset. Accordingly, the signals become stable as indicated in the 'FAIL' row of Table 2.

Although preferred embodiments of the present invention has been described and illustrated above, those skilled in the art will appreciate that various changes and modifications can be made to the present invention without departing from its spirit. Accordingly, the scope of the present invention is deemed to be limited only by the scope of the following appended claims.

What is claimed is:

1. A system for coordinating the initialization of multiprocessor systems utilizing a multiprocessor backplane bus, wherein said backplane bus has connector slots for connecting multiple central processing units, memory modules, and other input/output modules, said system for coordinating comprising:

a default boot central processing unit coupled to said backplane bus in a first slot;

at least one alternate boot central processing unit coupled to said backplane bus in a second slot, said alternate boot central processing unit substantially compatible with said default boot central processing unit;

a first slot select circuit coupled to said default boot central processing unit, said first slot select circuit configured to release said default boot central processing unit from a reset state upon system start-up and allow said default central processing unit to execute said boot code, and further configured to place said default boot central processing unit in a reset state if a first predetermined time-out period, selected by a first respective slot identification code, elapses before said default central processing unit successfully boots; and

a second slot select circuit coupled to said alternate boot central processing unit, wherein said second slot select circuit is configured to hold said alternate processing unit in a reset state until either said default boot central processing unit successfully boots, or a second predetermined time-out period, selected by a second respective slot identification code, elapses, indicating that said default boot central processing unit failed to boot the system, and wherein said second slot select circuit is further configured to release said alternate boot central processing unit from said reset state after said default central processing unit successfully boots or said second predetermined time-out period elapses.

2. The system for coordinating the initialization of multiprocessor systems of claim 1, wherein said second predetermined time-out period is longer than said first predetermined time-out period.

3. The system for coordinating the initialization of multiprocessor systems of claim 1, wherein said second predetermined time-out period is equal to said first predetermined time-out period.

4. The system for coordinating the initialization of multiprocessor systems of claim 1, wherein said default boot central processing unit comprises an INTEL 80486 based compatible processor and supporting resources, and wherein said alternate boot central processing unit comprises and INTEL 80486 based compatible processor and supporting resources.

5. A system for coordinating the initialization and self-test of multiprocessor systems, said multiprocessor system having a backplane bus with slots for connecting multiple central processing units, memory modules, and other input/output modules, said system comprising:

a first central processing unit coupled to said backplane bus in a first slot, said first central processing unit assigned to boot first upon system start-up;

a second central processing unit coupled to said backplane bus in a second slot, said second central processing unit substantially compatible with said first central processing unit;

a first non-volatile memory having memory locations for storing central processing unit boot instructions in executable form, said first non-volatile memory directly accessible by said first and said second central processing units;

first means coupled to said first central processing unit for releasing said first central processing unit from a reset state during system start-up to allow said first central processing unit to execute said central processing unit boot instructions, and for placing said first central processing unit in a reset state if a first predetermined time-out period, selected by a first respective slot identification code, elapses before said first central processing unit successfully boots; and

second means coupled to said second central processing unit for holding said second central processing unit in the reset state until either said first central processing unit successfully boots, or a second predetermined time-out period, selected by a second respective slot identification code, elapses, and for releasing said second central processing unit from said reset state after said first central processing unit successfully boots or said second predetermined time-out period elapses.

6. The system for coordinating the initialization and self-test of multiprocessor systems of claim 5, further comprising first slot identification code circuitry coupled to said first slot and a second slot identification code circuitry coupled to said second slot, said first and said second slot identification code circuitry configured to provide mutually exclusive slot identification codes to a central processing unit, memory module, or input-/output module coupled to the backplane bus in the respective slot.

7. The system for coordinating the initialization of multiprocessor systems of claim 5, wherein said second predetermined time-out period is equal to said first predetermined time-out period.

8. The system for coordinating the initialization of multiprocessor systems of claim 5, wherein said second predetermined time-out period is longer than said first pre-determined time-out period.

* * * * *

EXHIBIT
AA

Appendix A

This claim chart compares claims 1-6 and 9-15 to U.S. Patent No. 7,328,371 to Kalyanasundharam et al. ("Kalyanasundharam") (Ex. 1007) in view of U.S. Patent No. 5,450,576 to Kennedy ("Kennedy") (Ex. 1005). Kalyanasundharam qualifies as prior art under at least 35 U.S.C. § 102(e) and is a valid prior-art reference for the purpose of establishing an obviousness rejection under 35 U.S.C. § 103(a). Kennedy qualifies as prior art under at least 35 U.S.C. §§ 102(a), 102(b), and 102(e) and is a valid prior-art reference for the purpose of establishing an obviousness rejection under 35 U.S.C. § 103(a).

The table below reflects Ground 1 presented in the corresponding Request for Reexamination. Further, corroborated evidence reflecting the perspective knowledge of a person having ordinary skill in the art at the time of the invention of the '781 Patent (a "POSITA") is provided through the declaration of Wolfe (Ex. 1003).

References
Kalyanasundharam (Ex. 1007)
Kennedy (Ex. 1005)

Claim Language	Relevant Prior Art Disclosures and Argument
[1.pre] A system comprising:	Kalyanasundharam disclosed a computer system. For example, Kalyanasundharam taught that "the computer system 10 includes nodes 12A-12B, memories 14A-14B, and peripheral devices 16A-16B." *Kalyanasundharam* (Ex. 1007) at 2:26-29.

[1.1] a plurality of processors;	Kalyanasundharam disclosed multiple processors executing in lockstep. For example, Kalyanasundharam taught 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. "For example, some fault-tolerant systems may implement high reliability by including **multiple processors** and having them redundantly execute the same software code." *Id.* at 1:23-25. Kalyanasundharam's description of processor cores is consistent with the '781 Patent's use of the term "processor." *See, e.g., '781 Patent* (Ex. 1001) at 6:13-19. For example, the '781 Patent notes that "[s]ystem 10 includes OS 11, as well as master processor 12A and slave processor 12B (collectively referred to as a lockstep processor pair 12)." *See id* at 6:13-15. The '781 Patent further notes that "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 12A is a first core and slave processor 12B is a second core." *See id.* at 6:15-19. Accordingly, Kalyanasundharam disclosed multiple processors executing in lockstep.

[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; and	Kalyanasundharam disclosed, or at least rendered obvious, this limitation by disclosing a PC bit in a configuration register that identifies the role of a processor as either a primary processor or an alternate processor. For example, Kalyanasundharam taught that "the communications from the primary processor core may be propagated and the communications from the auxiliary processor core(s) may be discarded…In this embodiment, the **PC bit** from the **configuration register 42** may **identify the primary processor core.**" *Kalyanasundharam* (Ex. 1007) at 8:34-44. "The **configuration register 42** includes a **primary core (PC) field** encoded to **identify which of the processor cores 18A-18N is designated as the primary core.**" *Id.* at 8:34-44. "In such embodiments, one processor core may be the **primary core** and two or more of the remaining processor cores may be **auxiliary processor cores.**" *Id.* at 4:66 – 5:2. Accordingly, Kalyanasundharam described assigning the roles of primary processor core and auxiliary processor core to certain processors via the PC bit in the configuration register. *See id* at 8:34-44, 4:66 – 5:2. In view of these citations and the knowledge that a boot processor was a common component in computer systems (*e.g.,* Kalyanasundharam's system), a POSITA would have understood Kalyanasundharam as disclosing a boot processor and a spare processor. *See supra* § I.C.2; *see also Wolfe Decl.* (Ex. 1003) ¶ 169. Even if Kalyanasundharam did not explicitly identify which processor was the boot processor, Kalyanasundharam 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, Kalyanasundharam taught data storage that stores role assignments (*e.g.,* primary core and auxiliary core) of the processors, and Kennedy taught that such assignments could be the roles of boot processor and alternate processor. *See Wolfe Decl.* (Ex. 1003) ¶ 169. A POSITA would have been motivated to combine these disclosures from Kalyanasundharam and Kennedy to improve the reliability of Kalyanasundharam'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 Kalyanasundharam's

	system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam 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.*

[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,	Kalyanasundharam as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the replacement of a primary processor with an auxiliary processor in response to a lockstep mismatch. For example, Kalyanasundharam taught 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. "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. Accordingly, Kalyanasundharam described transferring the role of primary processor to an auxiliary processor in response to a mismatch involving the primary processor. *See id.* In view of these citations and the knowledge that a boot processor was a common component in computer systems (*e.g.*, **Kalyanasundharam's system**), a POSITA would have understood Kalyanasundharam as disclosing the failover of a boot processor to a spare processor in response to a loss of lockstep. *See supra* § I.C.2; *see also Wolfe Decl.* (Ex. 1003) ¶ 170. Even if Kalyanasundharam did not explicitly identify the mismatched processor as having been the boot processor, Kalyanasundharam as modified by Kennedy would have performed boot processor failover in response to a lockstep mismatch. *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, Kalyanasundharam taught logic that causes its system to replace the primary processor with an auxiliary processor in response to a mismatch involving the primary processor, and Kennedy taught logic that causes its system to fail over the boot

	processor to an alternate processor. *See Wolfe Decl.* (Ex. 1003) ¶ 170. A POSITA would have been motivated to combine these disclosures from Kalyanasundharam and Kennedy to improve the reliability of Kalyanasundharam'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 Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam 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.*

[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.	Kalyanasundharam as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing processors that have the same functionality as a hot spare. For example, **Kalyanasundharam taught 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. Even if Kalyanasundharam did 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 as the hot spare recited in claim 1. For example, in view of the above citations, Kalyanasundharam taught 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. *See Wolfe Decl.* (Ex. 1003) ¶ 171. For at least these reasons, a POSITA would have understood Kalyanasundharam as having taught that the auxiliary processor that assumes the role of primary processor **would have been disabled prior to the primary processor's** mismatch. *Id.* Accordingly, Kalyanasundharam as modified by Kennedy taught that the processor assigned the role of spare would be held in reserve until a loss of lockstep is detected, thereby rendering this subject matter obvious to a POSITA. *Id.*

[2] The system of claim 1 wherein said data storage is non-volatile data storage.	Kalyanasundharam as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing a PC bit in a configuration register that identifies the role of a processor as either a primary processor or an alternate processor. For example, Kalyanasundharam taught that "the communications from the primary processor core may be propagated and the communications from the auxiliary processor core(s) may be discarded…In this embodiment, **the PC bit from the configuration register 42** may identify the primary processor core." *Kalyanasundharam* (Ex. 1007) at 8:34-44. "The **configuration register 42** includes **a primary core (PC) field** encoded to **identify which of the processor cores 18A-18N is designated as the primary core."** *Id.* at 8:34-44. Even if Kalyanasundharam did not explicitly identify the configuration register as being non-volatile data storage, Kalyanasundharam as modified by Kennedy would have included non-volatile memory that contains the configuration information. For example, "The 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 data storage. *See Wolfe Decl.* (Ex. 1003) ¶ 172. It was well within the skill of a POSITA to modify Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam as modified by Kennedy would have rendered the subject matter of claim 2 obvious to a POSITA. *Id.*

[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.	Kalyanasundharam 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 Kalyanasundharam did not explicitly disclose a user interface that receives input specifying the processor assigned the role of boot processor, Kalyanasundharam 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) ¶ 173. 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 Kalyanasundharam 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 **Kalyanasundharam's system by Kennedy in this manner, and** thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam 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.*

[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.	Kalyanasundharam 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 Kalyanasundharam did not explicitly disclose a user interface that receives input specifying the processor assigned the role of spare processor, Kalyanasundharam 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) ¶ 174. 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 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 Kalyanasundharam 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 Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam 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.*

[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.	Kalyanasundharam as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing nodes that each included a primary processor and an auxiliary processor. For example, Kalyanasundharam taught that "[t]he **nodes 12A-12B** are coupled." *Kalyanasundharam* (Ex. 1007) at 2:30. "The **node 12A comprises processor cores 18A-18B** coupled to a node controller 20A." *Id.* at 2:33-34. "The **node 12B similarly comprises processor cores 18C-18D** coupled to a node controller 20B." *Id.* at 2:36-37. The node controller 20A may be configured to **propagate communications from one of the processor cores 18A-18B (e.g., from the primary processor core)** and to **discard communications from the other processor core 18A-18B (e.g., from the auxiliary processor core).**" *Id.* at 4:36-40. "In one embodiment, **each of the nodes 12A-12B may be a single integrated circuit chip** comprising the circuitry shown therein in FIG. 1. That is, **each node 12A-12B may be a chip multiprocessor (CMP).**" *Id.* at 2:46-49. In view of these citations, Kalyanasundharam taught nodes that each included a primary processor and an auxiliary processor, but only the primary processor propagates communications because the auxiliary processor's communications are discarded. Therefore, under the broadest reasonable interpretation consistent with the '781 Patent, a POSITA would have understood Kalyanasundharam's nodes as each constituting a processor module, Kalyanasundharam's primary processor core as constituting a master processor, and Kalyanasundharam's auxiliary processor core as constituting a slave processor. *See Wolfe Decl.* (Ex. 1003) ¶ 175. Accordingly, Kalyanasundharam as modified by Kennedy taught processor modules that each include a master processor and a slave processor. *Id.*

[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.	Kalyanasundharam as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing that the spare to the boot processor may be an enabled (*e.g.*, active) processor core by control circuit. For example, **Kalyanasundharam taught that "[t]he control circuit 84 may use** any selection mechanism. For example, the processor cores may be numbered (*e.g.*, 0 to N-1), and the control circuit 84 may **increment** the number of the presently selected primary processor core **to select a new primary processor core."** *Kalyanasundharam* (Ex. 1007) at 13:29-**33. "If one or more** processor cores are disabled in the configuration register 42, the control circuit 84 may increment until **an enabled processor core is selected."** *Id.* at 13:33-36. In view of these citations, Kalyanasundharam taught that the spare to the primary processor may be an enabled (*e.g.*, active) processor core selected by control circuit 84. *See Wolfe Decl.* (Ex. 1003) ¶ 176. Although Kalyanasundharam did not explicitly use the term **"boot processor" when describing the primary** processor, Kalyanasundharam as modified by Kennedy would have performed boot processor failover. Based on the **supplementation of Kennedy's disclosures, a POSITA would have understood Kalyanasundharam's primary processor to** have constituted a boot processor for at least the same reasons provided above in connection with claim 1. *See id.* Accordingly, Kalyanasundharam as modified by Kennedy taught that the spare processor is an active spare before being transferred the role of boot processor. *Id.*
[9.pre] A method comprising:	**Kalyanasundharam disclosed a method. "A similar method** is **also contemplated."** *Kalyanasundharam* (Ex. 1007), Abstract.

[9.1] assigning a role of boot processor to one of a plurality of processors in a system;	Kalyanasundharam as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the assignment of a primary processor role to a certain processor core. For example, Kalyanasundharam taught that "the communications from **the primary processor core** may be propagated and the communications from the auxiliary processor core(s) may be discarded…In this embodiment, **the PC bit** from the configuration register 42 may **identify the primary processor core.**" *Kalyanasundharam* (Ex. 1007) at 8:34-44. "The **configuration register 42** includes a **primary core (PC) field** encoded to **identify which of the processor cores 18A-18N** is designated as **the primary core.**" *Id.* at 8:34-44. In view of these citations, Kalyanasundharam described assigning the role of primary processor to one of processor cores 18A-18N. In view of these citations and the knowledge that a boot processor was a common component in computer systems (*e.g.,* Kalyanasundharam's system), a POSITA would have understood Kalyanasundharam as disclosing a boot processor and a spare processor. *See supra* § I.C.2; *see also Wolfe Decl.* (Ex. 1003) ¶ 177. Even if Kalyanasundharam did not explicitly identify which processor was the boot processor, Kalyanasundharam 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." *Kennedy* (Ex. 1005) at 7:9-13. Therefore, Kalyanasundharam taught assigning the role of primary processor to a certain processor core, and Kennedy taught assigning the role of boot processor to one of its processors. *See Wolfe Decl.* (Ex. 1003) ¶ 177. A POSITA would have been motivated to combine these disclosures from Kalyanasundharam and Kennedy to improve the reliability of Kalyanasundharam's system by assigning 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 Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam as modified by Kennedy taught assigning the role of boot processor to one of its processor cores. *Id.*

[9.2] assigning a role spare processor to another of said plurality of processors;	Kalyanasundharam as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the assignment of an auxiliary (*e.g.,* spare) processor role to at least one of its processor cores. For example, Kalyanasundharam **taught that "the communications from the primary processor** core may be propagated and the communications from the **auxiliary processor core(s)** may be discarded…In this embodiment, **the PC bit** from the configuration register 42 may **identify the primary processor core."** *Kalyanasundharam* (Ex. 1007) at 8:34-44. **"The configuration register 42** includes a primary core (PC) field encoded to **identify which of the processor cores 18A-18N** is designated as **the primary core."** *Id.* at 8:34-44. **"In such embodiments, one processor core may** be the primary core and two or more of the remaining processor cores may be **auxiliary processor cores."** *Id.* at 4:66 – 5:2. **"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. In view of these citations, Kalyanasundharam described assigning the role of auxiliary (*e.g.,* spare) processor to at least one of its processor cores. Accordingly, Kalyanasundharam as modified by Kennedy taught assigning the role of spare processor to one of its processor cores. *Id.*

[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; and	Kalyanasundharam as modified by Kennedy disclosed, or at least rendered obvious, this limitation by disclosing the replacement of a primary processor with an auxiliary processor in response to a lockstep mismatch. For example, Kalyanasundharam taught 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. "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. Accordingly, Kalyanasundharam described changing the role of primary processor to an auxiliary processor in response to a mismatch (*e.g.,* loss of lockstep) involving the primary processor. *See id.* In view of these citations and the knowledge that a boot processor was a common component in computer systems (*e.g.,* Kalyanasundharam's system), a POSITA would have understood Kalyanasundharam as disclosing the failover of a boot processor to a spare processor in response to a loss of lockstep. *See supra* § I.C.2; *see also Wolfe Decl.* (Ex. 1003) ¶ 178. Even if Kalyanasundharam did not explicitly identify the mismatched processor as having been the boot processor, Kalyanasundharam as modified by Kennedy would have changed the boot processor assignment in response to a lockstep mismatch. *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, Kalyanasundharam taught changing the role of primary processor to the spare processor in response to a loss of lockstep involving the primary processor, and Kennedy taught changing the role of boot processor to an alternate processor.

	See Wolfe Decl. (Ex. 1003) ¶ 178. A POSITA would have been motivated to combine these disclosures from Kalyanasundharam and Kennedy to improve the reliability of Kalyanasundharam'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 Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam as modified by Kennedy taught changing the role of boot processor to the spare processor during system runtime in response to detecting loss of lockstep for the boot processor. *Id.*

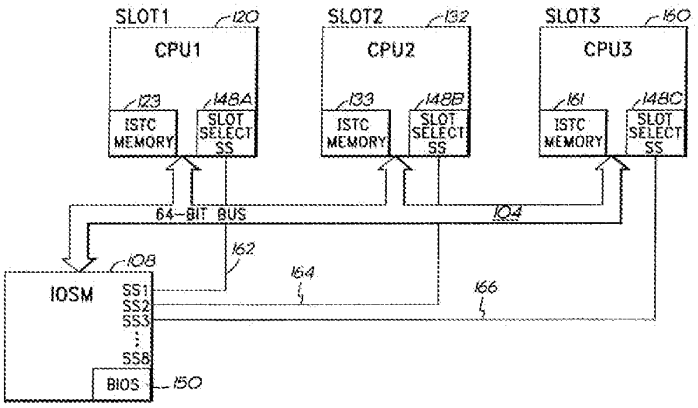
[9.4] responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system.	Kalyanasundharam as modified by Kennedy disclosed, or at least rendered obvious, this limitation 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. In other words, even if Kalyanasundharam did not explicitly disclose an operating system that recognizes the spare processor as the boot processor after the role change, Kalyanasundharam 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. *See Wolfe Decl.* (Ex. 1003) ¶ 179. Or at the very least, Kalyanasundharam as modified by Kennedy would have rendered the subject matter of claim 9 obvious to a POSITA due at least in part to Kennedy's central boot ROM 150 being used by the operating system and storing the boot code executed in conjunction with the operating system to boot the system. *Id.* It was well within the skill of a POSITA to modify Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.*
[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.
[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.

[12] 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; and	Kalyanasundharam 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. Even if Kalyanasundharam did not explicitly disclose determining a processor that is coupled to a common communication bus with the boot processor, Kalyanasundharam 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) ¶ 180. 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 Kalyanasundharam to improve the communication capabilities of Kalyanasundharam's system for fault tolerance and high availability. *Id.* It was well within the skill of a POSITA to modify Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam as modified by Kennedy taught this claim limitation. *Id.*

[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	Kalyanasundharam 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 Kalyanasundharam did not explicitly disclose selecting the processor coupled to the common communication bus as the spare processor, Kalyanasundharam 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. In view of these citations, Kennedy taught selecting one or more alternate (*e.g.,* spare) boot processors by installing those alternate boot processors into specific slots corresponding to that alternate boot role on the backplane bus. A POSITA would have been motivated to combine these disclosures from Kalyanasundharam and Kennedy to improve the reliability of Kalyanasundharam'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) ¶ 181. It was well within the skill of a POSITA to modify Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam as modified by Kennedy taught this claim limitation. *Id.*

[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.	Kalyanasundharam 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. For example, Kalyanasundharam 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 Kalyanasundharam did not explicitly disclose doing so via system firmware, Kalyanasundharam 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) ¶ 182. 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 Kalyanasundharam 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 Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam as modified by Kennedy would have rendered the subject matter of claim 13 obvious to a POSITA. *Id.*

[14] The method of claim 9 further comprising: during system boot-up, system firmware booting the spare processor as an active spare.	Kalyanasundharam 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. For example, Kalyanasundharam taught booting the spare processor as an active spare for at least the same reasons provided above in connection with claim 6. Even if Kalyanasundharam did not explicitly disclose doing so via system firmware, Kalyanasundharam 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) ¶ 183. 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 Kalyanasundharam and Kennedy to enable Kalyanasundharam'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 Kalyanasundharam's system by Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam as modified by Kennedy would have rendered the subject matter of claim 14 obvious to a POSITA. *Id.*
[15] The method of claim 9 wherein system firmware performs said changing operation.	Kalyanasundharam 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. Kalyanasundharam taught

	performing the changing operation for at least the same reasons provided above in connection with claim 9. Even if Kalyanasundharam did not explicitly disclose doing so via system firmware, Kalyanasundharam 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) ¶ 184. 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.** *See id.* A POSITA would have been motivated to combine these disclosures from Kalyanasundharam 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 Kalyanasundharam's system by** Kennedy in this manner, and thus this combination of Kalyanasundharam and Kennedy would have been expected to succeed. *Id.* Accordingly, Kalyanasundharam as modified by Kennedy would have rendered the subject matter of claim 15 obvious to a POSITA.

EXHIBIT
BB

Appendix B

This claim chart compares claims 1-6 and 9-15 to U.S. Patent No. 7,085,959 to Safford ("Safford") (Ex. 1004) and U.S. Patent No. 5,450,576 to Kennedy ("Kennedy") (Ex. 1005). Safford qualifies as prior art under at least 35 U.S.C. §§ 102(a) and 102(e) and is a valid prior-art reference for the purpose of establishing an obviousness rejection under 35 U.S.C. § 103(a). Kennedy qualifies as prior art under at least 35 U.S.C. §§ 102(a), 102(b), and 102(e) and is a valid prior-art reference for the purpose of establishing an obviousness rejection under 35 U.S.C. § 103(a).

The table below reflects Ground 2 presented in the corresponding Request for Reexamination. Further, corroborated evidence reflecting the perspective knowledge of a person having ordinary skill in the art at the time of the invention of the '781 Patent (a "POSITA") is provided through the declaration of Wolfe (Ex. 1003).

References
Safford (Ex. 1004)
Kennedy (Ex. 1005)

Claim Language	Relevant Prior Art Disclosures
[1.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.
[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.

[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; and	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 supra* § I.C.2; *see also Wolfe Decl.* (Ex. 1003) ¶ 187. Even if Safford is somehow interpreted to lack an explicit disclosure of 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.*

[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 supra* § I.C.2; *see also 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.*
[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.

[2] 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. *Id.*

[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.	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) ¶ 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.*

[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.	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.*

[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.	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.
[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.	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.*

[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.

[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 supra* § I.C.2; *see also 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.*
[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.*

[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; and	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 supra* § I.C.2; *see also 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.*

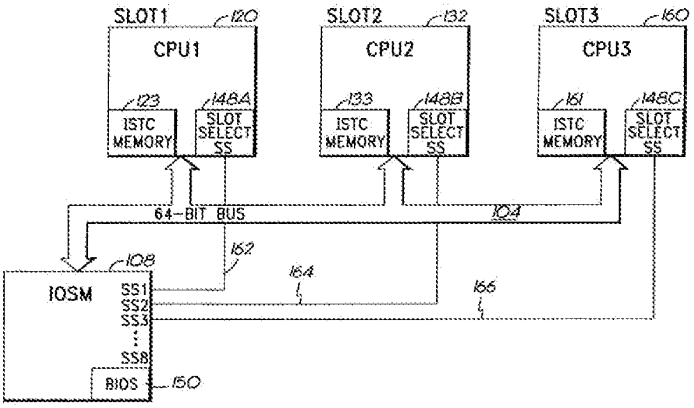
[9.4] responsive to changing the assignment of the role of boot processor to said spare processor, an operating system of the system recognizing said spare processor as the boot processor without requiring shutdown of said operating system.	Safford as modified by Kennedy disclosed, or at least rendered 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. Even if Safford did not explicitly disclose 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 processor**s…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.*
[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.
[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.
[12] 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; and	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. Even if Safford did not explicitly disclose determining a processor that is coupled to a common communication bus with the boot processor, 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.*

[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.*

[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.	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.*

[14] 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.*
[15] 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.

EXHIBIT
CC

Appendix C

This claim chart compares claims 1-6 and 9-15 to U.S. Patent No. 5,450,576 to Kennedy et al. ("Kennedy") (Ex. 1005) in view of U.S. Patent Publication No. 2004/0123201 to Nguyen et al. ("Nguyen") (Ex. 1008) and Fox U.S. Patent No. 7,251,746 to Fox ("Fox") (Ex. 1006). Kennedy qualifies as prior art under at least 35 U.S.C. §§ 102(a), 102(b), and 102(e) and is a valid prior-art reference for the purpose of establishing an obviousness rejection under 35 U.S.C. § 103(a). Nguyen qualifies as prior art under at least 35 U.S.C. §§ 102(a) and 102(e) and is a valid prior-art reference for the purpose of establishing an obviousness rejection under 35 U.S.C. § 103(a). Fox qualifies as prior art under at least 35 U.S.C. §§ 102(e) and is a valid prior-art reference for the purpose of establishing an obviousness rejection under 35 U.S.C. § 103(a).

The table below reflects Ground 3 presented in the corresponding Request for Reexamination. Further, corroborated evidence reflecting the perspective knowledge of a person having ordinary skill in the art at the time of the invention of the '781 Patent (a "POSITA") is provided through the declaration of Wolfe (Ex. 1003).

References
Kennedy (Ex. 1005)
Nguyen (Ex. 1008)
Fox (Ex. 1006)

Claim Language	Relevant Prior Art Disclosures and Argument
[1.pre] A system comprising:	Kennedy disclosed a system. For example, Kennedy taught "multiprocessor computer systems." *Kennedy* (Ex. 1005) at 1:14-15.
[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.

[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; and	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.*

[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.*

[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.*

[2] 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.*

[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.	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.*

[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.	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.*

[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.	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.

[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.	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.*

[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.	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.*

[8] 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) ¶ 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.
[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.
[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.*

[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.*

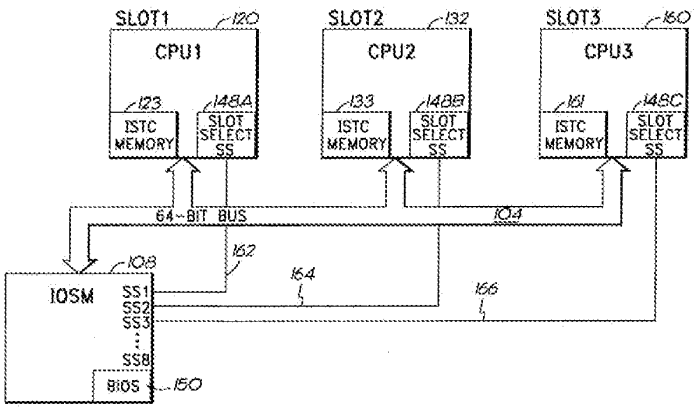
[9.3] responsive to detecting loss of lockstep for said boot processor, changing, during system runtime, the assignment of the role of boot processor to said spare processor; and	Kennedy as modified by Nguyen disclosed, or at least rendered obvious, this limitation by disclosing the replacement of the 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.*

[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.*
[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.
[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.

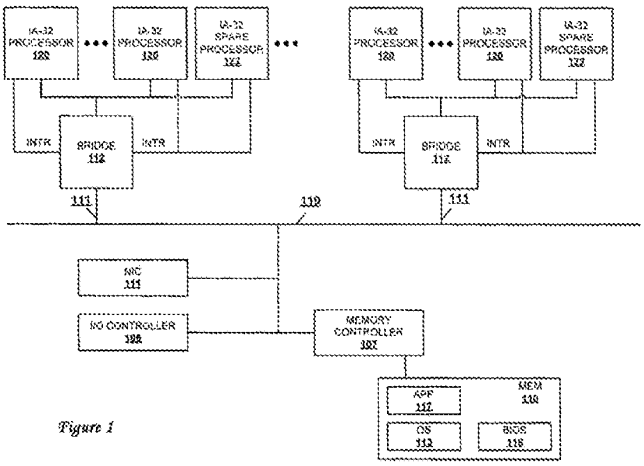
[12] The method of claim 9 further comprising [12.1] determining at least one processor that is coupled to a common communication bus with said one of said plurality processors assigned to the role of boot processor; and	Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this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

[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.	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 com**prises 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.*

[14] 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.*
[15] 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.*
[16][16.pre] A system comprising:	Kennedy disclosed a system. For example, Kennedy taught "multiprocessor computer systems." *Kennedy* (Ex. 1005) at 1:14-15.
[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.*
[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.*
[16.3] a second communication	Kennedy as modified by Nguyen and Fox disclosed, or at least rendered obvious, this limitation by disclosing the processors as being

bus with which at least a second two of said plurality of lockstep processor modules are communicatively coupled;	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.c*
[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.*

[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.
[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.
[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.
[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.
[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.	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.
[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.
[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.
[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.
[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.

[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.
[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.
[28]	Claim 28 recites subject matter that is substantially similar to claims 4 and 20 and is therefore rendered obvious for the same reasons as claims 4 and 20 above.
[29]	Claim 29 recites subject matter that is substantially similar to a limitation of claim 1 and is therefore rendered obvious for the same reasons as that limitation of claim 1 above.
[30]	Claim 30 recites subject matter that is substantially similar to portions of claims 9 and 13-15 and is therefore rendered obvious for the same reasons as claims 9 and 13-15 above.

EXHIBIT
DD

Appendix D

This claim chart compares claims 1 and 6 to U.S. Patent No. 7,085,959 to Safford ("Safford") (Ex. 1004) in light of U.S. Patent 5,491,788 to Cepulis et al. ("Cepulis")(Ex. 1013). Safford qualifies as prior art under at least 35 U.S.C. §§ 102(a) and 102(e) and is a valid prior-art reference for the purpose of establishing an obviousness rejection under 35 U.S.C. § 103(a). Cepulis qualifies as prior art under at least 35 U.S.C. §§ 102(a) and 102(b).

The table below reflects Ground 4 presented in the corresponding Request for Reexamination. Further, corroborated evidence reflecting the perspective knowledge of a person having ordinary skill in the art at the time of the invention of the '781 Patent (a "POSITA") is provided through the declaration of Wolfe (Ex. 1003).

References
Safford (Ex. 1004)
Cepulis (Ex. 1013)

Claim Language	Relevant Prior Art Disclosures
[1.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.
[1.1] a plurality of processors;	Safford disclosed a plurality of processors included in the computer system. For example, Safford disclosed "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.
[1.2] data storage storing information that assigns a role of	Safford disclosed this limitation, whether expressly or inherently, by disclosing memory

boot processor to one of said plurality of processors and assigns a role of spare processor to another of said plurality of processors; and	that stores the architected states of microprocessors, including the designation of a hot standby processor, and by disclosing the logical CPU 0 designation. For example, Safford **disclosed "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) ¶ 230. 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, e.g., Cepulis* (Ex. 1013) at 2:52-64, *see also*

	Wolfe Decl. (Ex. 1003) ¶ 230. For example, during boot-processor selection, If CPU P0 is not functioning properly and cannot perform even the most basic functions, it is designated as inoperative. The hardware system then awakens CPU P1 and repeats the process of testing the CPU. If CPU P1 is operational, then it is designated as CPU L0, and it boots the remainder of the system. On the other hand, if CPU P1 also fails, it is also given an inoperative designation. The computer system then turns on CPU P2, and repeats the process. The process repeats until an operational microprocessor is found to perform the CPU L0 functions.[1] *Cepulis* (Ex. 1013) at 2:52-64. **A POSITA would have also recognized "CPU 0" and "CPU 1" as being physical designations** used by computing system 100 to identify processors 111 and 113. The use of both logical and physical designators by a computing system is generally necessary when the role of boot processor can be assigned and reassigned to a processor regardless of its physical position within a system. *See Wolfe Decl.* (Ex. 1003) ¶ 230; *see also Safford* (Ex. 1007) at 4:52-53 **("The processor pair 125/127 then becomes the logical CPU 0 in the computer system 100").** Thus, a POSITA would have understood **"logical CPU 0" to be the boot**-processor designation used by an operating system to determine which of its processors is the boot processor and/or to distinguish its boot processor from its non-boot processors. *See Wolfe Decl.* (Ex. 1003) ¶ 230. Furthermore, because the **architected states of Safford's processors are** copied from memory, a POSITA would have understood the memory to be storing the boot-processor designation (*e.g.*, logical CPU 0) and the hot-standby designation for the processors so

[1] **Cepulis is provided herein to explain the meaning of the term "boot processor" used in Safford, the primary reference.** *See* M.P.E.P. § 2131.01(B).

	assigned by Safford's system. *See id.* In view of these citations and the knowledge that a boot processor was a common component identified **as "logical CPU 0" in computer systems (***e.g.*, **Safford's system), a POSITA would have** understood Safford as disclosing data storage that stores information assigning the role of boot processor to one of the processors and assigns the role of spare processor to another one of the processors. *See supra* § I.C.2; *see also Wolfe Decl.* (Ex. 1003) ¶ 230.

[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 disclosed this limitation by disclosing the failover of a processor to a hot standby processor in response to a loss of lockstep. For example, Safford disclosed "[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) ¶ 231. 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

	disclosed "**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.* 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"). As explained above, a POSITA would have understood "logical CPU 0" to be the boot-processor designation used by an operating system to determine which of its processors is the boot processor and/or to distinguish its boot processor from its non-boot processors. *See supra* § I.C.2; *see also Wolfe Decl.* (Ex. 1003) ¶ 231. In view of these citations and the knowledge that a boot processor was a common component identified as "logical CPU 0" in computer systems (*e.g.,* Safford's system), a POSITA would have understood Safford as disclosing failover of the boot processor (*e.g.,* logical CPU 0) to a spare processor in response to a loss of lockstep for the boot processor, especially in light of Cepulis' explanation of the meaning of the "logical CPU 0" boot-processor designator. *See Wolfe Decl.* (Ex. 1003) ¶ 231. Accordingly, Safford disclosed 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.*

[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 disclosed replacing the failed processor with a hot standby processor that is sitting idle in response to an error. For example, Safford **disclosed 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 disclosed that the processor assigned the role of spare is a hot standby held in reserve until a loss of lockstep is detected.

[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.	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.*

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	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
10/972,588	7624302	2114	9200



Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 56436 on 03/18/2016

- **Correspondence Address**
- **Maintenance Fee Address**

The address of record for Customer Number 56436 is:

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

PART 1 - ATTORNEY/APPLICANT COPY

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	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
10/972,588	7624302	2114	9200

OC000000042094518

Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 22879 on 06/14/2010

- Correspondence Address
- Maintenance Fee Address
- Power of Attorney Address

The address of record for Customer Number 22879 is:

22879
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS, CO 80528

PART 1 - ATTORNEY/APPLICANT COPY
page 1 of 1

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,624,302 B2
APPLICATION NO. : 10/972588
DATED : November 24, 2009
INVENTOR(S) : Scott L. Michaelis et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 20, line 13, in Claim 13, after "the" delete "s".

In column 20, line 16, in Claim 13, before "processor;" insert -- boot --.

In column 21, line 8, in Claim 26, before "means" insert -- system firmware --.

Signed and Sealed this

Sixteenth Day of February, 2010

David J. Kappos

David J. Kappos
Director of the United States Patent and Trademark Office

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

ATTORNEY DOCKET NUMBER: 200404973-1

IN RE APPLICATION OF: U.S. Patent No. 7,624,302

USPTO CONFIRMATION CODE: 1079

APPLICATION NO.: 10/972,588

FILED: Oct. 25, 2004

EXAMINER: DIEU MINH T LE

GROUP ART UNIT: 2114

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

37 CFR 1.322 & 37 CFR 1.323 REQUEST FOR CERTIFICATE OF CORRECTION

HONORABLE COMMISSIONER OF PATENTS & TRADEMARKS

Sir:

The following is a request for a certificate of correction in Serial Number 10/972,588, now Patent Number 7,624,302.

A certificate of correction under 35 USC 254 is respectfully requested in the above-identified patent.

☒ All errors were the fault of the USPTO, no fee required. In the event that a further fee is required, please charge the amount to Deposit Account No. 082025.

☐ All errors were the fault of the applicant and, accordingly, please charge $100.00 to our Deposit Account No. 082025. In the event that a further fee is required, please charge the amount to the same Deposit Account.

☐ The errors were the fault of both the applicant and USPTO and, accordingly, please charge $100.00 to our Deposit Account No. 082025. In the event that a further fee is required, please charge the amount to the same Deposit Account.

The exact locations where the errors appear in the patent and patent application are as follows:

In column 20, line 13, in Claim 13, after "the" delete "s".
(Amendments to Claims filed on Sep. 10, 2008, page 4, in claim 16, line 8)

In column 20, line 16, in Claim 13, before "processor;" insert - - boot - -.
(Amendments to Claims filed on Sep. 10, 2008, page 4, in claim 16, line 9)

In column 21, line 8, in Claim 26, before "means" insert - - system firmware - -.
(Amendments to Claims filed on Sep. 10, 2008, page 5, in claim 29, line 4)

The requested corrections are attached on Form PTO 1050.

Respectfully Submitted

1/7/2010
DATE

Name: Steven L. Webb
Registration No.: 44, 395

Attorney/Agent of Record

Hewlett-Packard Company
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
Fort Collins, CO 80528

PTO/SB/44 (04-07)
Approved for use through 04/30/2007. OMB 0651-0033
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.
(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,624,302

APPLICATION NO. : 10/972,588

ISSUE DATE : Nov. 24, 2009

INVENTOR(S) : Scott L. Michaelis et al.

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 20, line 13, in Claim 13, after "the" delete "s".

In column 20, line 16, in Claim 13, before "processor;" insert - - boot - -.

In column 21, line 8, in Claim 26, before "means" insert - - system firmware - -.

MAILING ADDRESS OF SENDER (Please do not use customer number below):

Hewlett-Packard Company
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
Fort Collins, CO 80528

This collection of information is required by 37 CFR 1.322, and 1.323, and 1.324. 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 1.0 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: Attention Certificate of Corrections Branch, 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:	6814910
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:	Steven L. Webb/Marci Santillanes
Filer Authorized By:	Steven L. Webb
Attorney Docket Number:	200404973-1
Receipt Date:	14-JAN-2010
Filing Date:	25-OCT-2004
Time Stamp:	18:47:41
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 Certificate of Correction	200404973-1.pdf	499942 c701486e5f67880347a88e29895e97677a6cd3c	no	3
Warnings:					
Information:					

Total Files Size (in bytes):	499942

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.

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.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/972,588	11/24/2009	7624302	200404973-1	1079

22879 7590 11/04/2009

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS, CO 80528

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 513 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

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 Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Scott L. Michaelis, Plano, TX;
Anurupa Rajkumari, Round Rock, TX;
William B. McHardy, Fort Collins, CO;

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 10/20/2009

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS, CO 80528

EXAMINER	
LE, DIEU MINH T	

ART UNIT	PAPER NUMBER
2114	

NOTIFICATION DATE	DELIVERY MODE
10/20/2009	ELECTRONIC

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.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
ipa.mail@hp.com
jessica.l.fusek@hp.com

UNITED STATES DEPARTMENT OF COMMERCE
U.S. Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.
10972588	10/25/04	MICHAELIS ET AL.	200404973-1

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS, CO 80528

EXAMINER
Dieu-Minh Le

ART UNIT	PAPER
2114	20091014

DATE MAILED:

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

Commissioner for Patents

In response to the communication filed on 10/05/2009, The reference cited in this IDS has been considered by the Examiner.

/Dieu-Minh Le/
Primary Examiner, Art Unit 2114

PTO-90C (Rev.04-03)

RECEIVED
CENTRAL FAX CENTER
OCT 05 2009

PTO/SB/08b (07-09)
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.

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	October 25, 2009
First Named Inventor	Scott L. Michaelis,et.al
Art Unit	2114
Examiner Name	Dieu-Minh Le
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]
/M.L./		Notice of Allowance, dated July 23, 2009, serial number 10/973,077, 20 pages	

Examiner Signature	/Dieu Minh Le/	Date Considered	10/14/2009

*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 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.

OCT 06 2009 PATENT & TRADEMARK OFFICE

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 07/14/2009

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

10/06/2009 RMEBRAH1 00000174 082025 10972588

01 FC:1501 1510.00 DA
02 FC:1504 300.00 DA

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.

Marci Santillanes	(Depositor's name)
Marci Santillanes	(Signature)
10.06.09	(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	10/14/2009

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: Hewlett-Packard Development Company, L.P.
(B) RESIDENCE: (CITY and STATE OR COUNTRY): Houston, Texas

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 08-2025 (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 10/5/09

Typed or printed name Steven L. Webb Registration No. 44,395

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.

PAGE 1/2 * RCVD AT 10/6/2009 10:11:25 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-6/10 * DNIS:2732885 * CSID:970 898 2590 * DURATION (mm-ss):01-14

OCT 06 2009 PATENT & TRADEMARK OFFICE

COPY

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 07/14/2009

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.

Marci Santillanes	(Depositor's name)
Marci Santillanes	(Signature)
10.06.09	(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	10/14/2009

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: Hewlett-Packard Development Company, L.P.
(B) RESIDENCE: (CITY and STATE OR COUNTRY): Houston, Texas

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 08-2025 (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 10/5/09

Typed or printed name Steven L. Webb Registration No. 44,395

This collection of information is required by 37 CFR 1.31. 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.

PAGE 2/2 * RCVD AT 10/6/2009 10:11:25 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-6/10 * DNIS:2732885 * CSID:970 898 2590 * DURATION (mm-ss):01-14

RECEIVED
CENTRAL FAX CENTER
OCT 05 2009

PTO/SB/08b (07-09)
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.

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	October 25, 2009
First Named Inventor	Scott L. Michaelis,et.al
Art Unit	2114
Examiner Name	Dieu-Minh Le
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 July 23, 2009, serial number 10/973,077, 20 pages	

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.

RECEIVED
CENTRAL FAX CENTER
OCT 05 2009

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 et al

Confirmation No.: 1079

Application No.: 10/972,588

Examiner: Dieu-Minh Le

Filing Date: Oct 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 $ 180.00 . 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.

☐ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450

Date of Deposit:

OR

☒ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571) 273-8300
Date of facsimile: 10.05.09
Typed Name: Marci Sandilanas
Signature: Marci Sandilanas

Respectfully submitted,
Scott L. Michaelis et al
By
Steven L Webb
Attorney/Agent for Applicant(s)
Reg No.: 44,395
Date: 10/5/09
Telephone: 970 898 3451

Rev 1508 (IISCon)

10/06/2009 HMARZI1 00000010 082025 10972588
01 FC:1806 180.00 DA

PAGE 1/23 * RCVD AT 10/5/2009 1:17:36 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/0 * DNIS:2738300 * CSID:970 898 2590 * DURATION (mm-ss):06-48

RECEIVED
CENTRAL FAX CENTER
OCT 05 2009

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 et al

Confirmation No.: 1079

Application No.: 10/972588

Examiner: Dieu-Minh Le

Filing Date: Oct 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

STATEMENT UNDER 37 CFR 1.97 (e)

The undersigned hereby certifies that either:

☐ Each item of information contained in the Information Disclosure Statement was cited for the first time in any communication from a foreign patent office in a counterpart foreign application, not more than three months prior to the filing of the statement, or

☒ 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 undersigned, after making reasonable inquiry, was known to any designated in 37 CFR 1.56 (c) more than three months prior to the filing of the Information Disclosure Statement.

☐ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450

Date of Deposit:

OR

☒ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571) 273-8300.

Date of facsimile: 10-05-09

Typed Name: Marci Santillanes

Signature: Marci Santillanes

Respectfully submitted,

Scott L. Michaelis et al

By

Steven L. Webb

Attorney/Agent for Applicant(s)

Reg No. : 44,395

Date : 10/5/09

Telephone : 970 898 3451

Rev 10/08 (Stat197e)

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 07/14/2009

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: 07/14/2009

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	10/14/2009

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 07/14/2009

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	10/14/2009

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 07/14/2009

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: 07/14/2009

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 RCE filed 5/12/09*.

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 5/12/09, 5/28/09, 5/29/09, 6/1/09, 6/2/09, and 6/3/09
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

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 for Serial No. 10/972,835 dated 11-20-2008 in the United States	
		Notice of Allowance for Serial No. 10/972,835 dated 8-1-2008 in the United States	
		Office Action for Serial No. 10/972,835 dated 3-3-2008 in the United States	
		Office Action for Serial No. 10/972,835 dated 7-12-2007 in the United States	

Examiner Signature	/Dieu Minh Le/	Date Considered	05/29/2009

*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] 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.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.L./

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]
		Office Action for Serial No. 10/973,003 dated 3-5-2008 in the United States	
		Office Action for Serial No. 10/973,003 dated 9-12-2007 in the United States	

Examiner Signature	/Dieu Minh Le/	Date Considered	06/17/2009

*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] 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.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.L./

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 for Serial No. 10/973,076 dated 11-28-2008 in the United States	
		Notice of Allowance for Serial No. 10/973,076 dated 7-16-2008 in the United States	
		Notice of Allowance for Serial No. 10/973,076 dated 6-3-2008 in the United States	
		Final Office Action for Serial No. 10/973,076 dated 2-27-2008 in the United States	
		Office Action for Serial No. 10/973,076 dated 9-12-2007 in the United States	
		Notice of Allowance for Serial No. 10/972,888 dated 11-14-2007 in the United States	
		Office Action for Serial No. 10/972,888 dated 7-18-2007 in the United States	

Examiner Signature	/Dieu Minh Le/	Date Considered	06/17/2009

*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] 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.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.L./

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 for Serial No. 10/973,004 dated 8-1-2007 in the United States	
		Office Action for Serial No. 10/973,004 dated 4-10-2007 in the United States	
		Final Office Action for Serial No. 10/973,075 dated 5-12-2009 in the United States	
		Office Action for Serial No. 10/973,075 dated 10-29-2008 in the United State	
		Office Action for Serial No. 10/973,075 dated 3-12-2008 in the United States	
		Office Action for Serial No. 10/973,075 dated 7-2-2007 in the United States	
		Notice of Allowance for Serial No. 10/972,796 dated 12-19-2007 in the United States	
		Office Action for Serial No. 10/972,796 dated 7-12-2007 in the United States	

Examiner Signature	/Dieu Minh Le/	Date Considered	06/17/2009

*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] 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.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.L./

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]
/M.L/		SHOICHI ARAKI, Real-Time Tracking of Multiple Moving Object Contours in a Moving Camera Image Sequence, Vol. E83-D IEICE Trans. Inf. & Syst, 7-7-2000, 9 pages	
no date		~~DONG KYU KIM, Active Object Tracking Using Image Mosaic Background, 4~~ pages	
/M.L/		JIANG-BIN ZHENG, Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 4-5 November 2002, 4 pages	✓
/M.L/		JONG RYUL KIM, Automatic Localization and Trackign of Moving Objects Using Adaptive Snake Algorithm, ICICS-PCM 2003, 15-18 December 2003, 5 pages, Singapore	
/M.L/		JIANHOA MENG, CVEPS-A Compressed Video Editing and Parsing System, 18-11-1996, Pages 1-11	

Examiner Signature	/Dieu Minh Le/	Date Considered	06/17/2009

*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] 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.

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 07/02/2009	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner) (Date)	1	1

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	315	spare adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:19
L2	229	(loss lose los$3 lost$3) same (lockstep lock-step "lock step")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:19
L3	95	(detect$3 track$3 monitor$3) same L2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:19
L4	22	L3 and L1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:19
L5	39197	(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	2009/07/01 14:31
L6	46413	multi-processor "multi processor" (plural$3 adj processor)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:31
L7	6914	L6 and L5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:31

L8	21	4 and 7	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:32
L9	33	(michaelis-scott$ or myer-sylvia$ or mchardy-william$).in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:32
L10	33	L9	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:32
L11	869	(714/11).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:32
L12	421	(714/12).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:32
L13	763	(714/13).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:32
L14	738	(714/10).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:32
L15	117	(712/31).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:32

L16	2421	L11 L12 L13 L14 L15	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:32
L17	2637	(boot$3 initializ$5 power-up) adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 14:34
L18	17	8 and 17	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 15:04
L19	16	((operating adj system) OS) and 18	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 15:36
L20	160944	firmware	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 15:39
L21	16	19 and L20	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 15:39
L22	6	(11 12 13 14 15) and 3 and 6 and 17	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 15:40
L23	13	("714"/$.ccls.) and 3 and 6 and 17	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 15:40

L24	0	("711"/$.ccls.) and 3 and 6 and 17	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/01 15:41
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	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

S50	1742	(boot$4 or initializ$4) adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 12:32
S51	94	S50 same S49	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 12:32
S52	16	(assign$4 giv$3 order$3 decid$3 mak$3) same role same S50	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 12:35
S53	2459	lockstep lock-step	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:00
S54	25133	(loss adj S53) LOL	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:00
S55	56	(detect$3 or track$4) adj S54	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:00
S56	13	S51 same S55	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:00
S57	0	S56 not S52	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01

S58	803	(714/11).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
S59	370	(714/12).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
S60	693	(714/13).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
S61	682	(714/10).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
S62	104	(712/31).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
S63	2237	S58 S59 S60 S61 S62	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:01
S64	803	(714/11).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
S65	289	spare adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02

S66	15	S55 and S50 and S65	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
S67	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
S68	92	(detect$3 track$3 monitor$3) same S67	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
S69	3	S68 and S64 and S66	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 13:02
S70	803	(714/11).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/04 13:46
S71	370	(714/12).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/04 13:46
S72	693	(714/13).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/04 13:46
S73	563	(714/10).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2008/09/04 13:46
S74	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
S75	9	S52 and S74	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 14:10
S76	0	S75 not S56	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/09/04 14:12

7/1/09 3:47:32 PM

C:\ Documents and Settings\ MLe\ My Documents\ EAST\ Workspaces\ rce310972588.wsp

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	6/30/09	dml
712	31	6/30/09	dml

SEARCH NOTES		
Search Notes	Date	Examiner
EAST search all files updated	6/30/09	dml
Inventor name search	6/30/09	dml
NPL search files [IEEE, Google Scholar]	7/1/09	dml

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
714	10-13	7/2/09	dml
712	31	7/2/09	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	869	(714/11).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/07/02 06:05
L2	421	(714/12).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/07/02 06:05
L3	763	(714/13).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/07/02 06:05
L4	619	(714/10).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/07/02 06:05
L5	117	(712/31).ccls.	US-PGPUB; USPAT; USOCR	OR	ON	2009/07/02 06:05
L6	38825	multi-processor "multi processor" (plural$3 adj processor)	US-PGPUB; USPAT; USOCR	OR	ON	2009/07/02 06:06
L7	33570	(pair spare slave redundan$4 backup back-up "back up" second$4) adj2 processor	US-PGPUB; USPAT; USOCR	OR	ON	2009/07/02 06:06
L8	1727	(boot$4 or initializ$4) adj processor	US-PGPUB; USPAT; USOCR	OR	ON	2009/07/02 06:07
L9	58002	(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	2009/07/02 06:07
L10	212	(loss lose los$3 lost$3) same (lockstep lock-step "lock step")	US-PGPUB; USPAT; USOCR	OR	ON	2009/07/02 06:08
L11	210	(detect$3 track$3 monitor$3) same 8	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:12
L12	50	11 and 6 and 7	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:12

L13	4	12 and 9 and 10 and 1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:12
L14	4	12 and 9 and 10 and 2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:12
L15	0	12 and 9 and 10 and 3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:13
L16	0	12 and 9 and 10 and 4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:13
L17	2	12 and 9 and 10 and 5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:13
L18	13	("714"/$.ccls.) and 12 and 9 and 10	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:13
L19	0	("711"/$.ccls.) and 12 and 9 and 10	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:13
L20	2	("712"/$.ccls.) and 12 and 9 and 10	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2009/07/02 06:14

7/ 2/ 09 6:14:21 AM
C:\ Documents and Settings\ MLe\ My Documents\ EAST\ Workspaces\ rce310972588.wsp

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	/Dieu Minh Le/	Date Considered	05/29/2009

*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] 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.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.L./

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]
		Final Office Action for Serial No. 10/972,835 dated 11-21-2007 in the United States	
		Office Action for Serial No. 10/973,075 dated 12-13-2007 in the United States	

Examiner Signature	/Dieu Minh Le/	Date Considered	06/17/2009

*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] 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.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /M.L./

Web Images Video Maps News Shopping Gmail more ▼

Sign in

detecting loss of lockstep multi-processor boot processor switching spare proces | Search | Advanced Search | Preferences

Web Show options... Results 1 - 10 of about 601 for **detecting loss of lockstep multi-processor boot processor switching spare processor**. (0.58 seconds)

1. System and method for configuring **lockstep** mode of a **processor** ...

A system comprising: a **processor** module that supports **lockstep** mode of operation ; ... "System and Method for **Switching** the Role of **Boot Processor** to a **Spare Processor** Responsive to **Detection** of **Loss of Lockstep** in a **Boot Processor**". ... and method for terminating **lock-step** sequences in a **multiprocessor** system ...
www.patentstorm.us/patents/7308566/claims.html - Similar

2. System and method for configuring **lockstep** mode of a **processor** ...

Dec 11, 2007 ... A system comprises a **processor** module that supports **lockstep** mode of operation. ... in a **Multi-Processor** System a **Spare Processor** That is in **Lockstep** for Use ... and Method for **Switching** the Role of **Boot Processor** to a **Spare Processor** Responsive to **Detection** of **Loss of Lockstep** in a **Boot Processor**". ...
www.freepatentsonline.com/7308566.html - Similar
by SL Michaelis - 2007

3. Self-checked, **lock step processor** pairs - Patent 6233702

A **multiprocessor** system includes a number of sub-**processor** systems, ... system for a **multiple processor** telecommunications **switching** control, February, 1983, De et al. .... 4907228 is also an example of this pair-and-**spare** technique, ... the local router will subsequently **detect** the **loss** of keep-alive symbols, ...
www.freepatentsonline.com/6233702.html - Similar
by RW Horst - 2001 - Cited by 11 - Related articles
More results from www.freepatentsonline.com »

4. [PDF] Software Optimization Guide for the AMD Hammer **Processor**

File Format: PDF/Adobe Acrobat
Chapter 11, "**Multiprocessor** Considerations," discusses **processor**/core selection and related ..... Many compilers have an "assume no aliasing" optimization **switch**. ..... The application is also typically responsible for **detecting** the ..... counters in a **processor** and these counters are incremented in **lockstep**. ...
www.amd.com/us-en/assets/content_type/white.../40546.pdf - Similar

5. **Processor** Editorial Article - Product Releases

May 8, 2009 ... Marathon Technologies everRun VM **Lockstep** & everRun 2G .... Manager is connected between a network patch panel and a **switch** or router, ... Updated every four hours, a list of precompiled malware-**detection** rules lets the ClamAV ... With an Intel Xeon Nehalem 5500 Series **processor**, the Stirling 118m ...
www.**processor**.com/editorial/article.asp?article...asp - Cached - Similar

6. MD-11 Electronic Instrument System - Digital Avionics Systems ...

Should a failure occur, the Master Caution or Master Warning **switch**/light .... Monitoring within each DEU is capable of **detecting** a **loss** of DU ... Each **processor** board is capable of running at a sustained rate of 3.6 ... d) 16Kbytes **Boot** PROM (UVPROM) .... **lock-step** processing to implement software partitioning. ...
ieeexplore.ieee.org/iel2/1033/6983/00262146.pdf?arnumber... - Similar
by J Morgan - 1992 - Cited by 4 - Related articles

7. Unixware 7 Non stop Clusters

A CPU dies, but its twin, which has been running in **lockstep** all this time, ... the hardware identically, and the **spare** sits around waiting for the need. .... NUMA clusters work by extending memory across **multiple** machines. .... Well, gee, if **processor** 1 is done, then it is available for more work, isn't it? ...
aplawrence.com/UW/clusters.html - Cached - Similar

8. Error **detection**/correction and fault **detection**/recovery inventions ...

Jul 19, 2007 ... The number of **spare** servers is reduced for improving the .... 20070174689 - Computer platform embedded operating system backup **switching** handling method and system: ... 20070174687 - Systems and methods for maintaining **lock step** ..... 20070168760 - Saving state data in parallel in a **multi-processor** ...
www.freshpatents.com/Error-detection-correction-and-fault-detection-recovery-d20070705ntc714.php - Cached - Similar

9. [PDF] HP ROM-Based Setup Utility User Guide

File Format: PDF/Adobe Acrobat - View as HTML
Selecting the primary **boot** controller. •. Configuring online **spare** memory .... **Lockstep** Mode provides increased data protection by enabling ..... **Multi-Processor** Specification (MPS) Table Mode (on page 105) ..... ROM-based utilities generically **switch** to run in either Windows® or VT100 mode to provide maximum ...
h20000.www2.hp.com/bc/docs/support/.../c00191707.pdf - Similar

10. NEC Fault-Tolerant Linux Server

A stock Linux kernel provides too little error **detection** and recovery for ... Both modules run the same instructions in **lockstep**, checking each ... After pressing the power **switch**, located under a hinged plastic ... We compared this server to a **spare** generic server having a single Athlon XP 2100+ **processor** using ...
www.linuxjournal.com/article/6709 - Cached - Similar

Gooooooooooogle
1 2 3 4 5 6 7 8 9 10 Next

detecting loss of lockstep multi-processor boot processor switching spare proces Search

Search within results - Language Tools - Search Help - Dissatisfied? Help us improve - Try Google Experimental

Google Home - Advertising Programs - Business Solutions - Privacy - About Google

IEEE Xplore® Digital Library
Home | Login | Logout | Access Information | Alerts | Purchase History | Help
Welcome United States Patent and Trademark Office
Citations | 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

Wed, 1 Jul 2009, 4:03:16 PM EST

Search Query Display

Run Search | Reset

Recent Search Queries		Results
#1	(((detecting monitoring tracking)<in>metadata) <and> ((loss of lockstep)<in>metadata))<and> ((boot processor) <in>metadata)	0
#2	(((switching changing swapping)<in>metadata) <and> ((boot processor)<in>metadata))<and> ((firmware)<in>metadata)	0
#3	(((spare processor)<in>metadata) <and> ((system identifier) <in>metadata))<and> ((shutting down os)<in>metadata)	0
#4	(((lol multi-processor)<in>metadata) <and> ((firmware) <in>metadata))<and> ((switching os)<in>metadata)	0
#5	(((system identifier)<in>metadata) <and> ((different os) <in>metadata))<and> ((lol boot processor)<in>metadata)	0

Clear Session History

Help Contact Us Privacy & Security IEEE.org
© Copyright 2009 IEEE – All Rights Reserved
Indexed by Inspec®

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	07/02/2009					
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/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]
		Final Office Action for Serial No. 10/972,835 dated 11-21-2007 in the United States	
		Office Action for Serial No. 10/973,075 dated 12-13-2007 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 Acknowledgement Receipt	
EFS ID:	5450360
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:	03-JUN-2009
Filing Date:	25-OCT-2004
Time Stamp:	21:26:30
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	NPL Documents	10973075-OA-12-13-07.pdf	645056	no	18
Warnings:					
Information:					

2	NPL Documents	10972835-FOA-11-21-2007.pdf	652102 9bc894b32fed5098b91e5721e9a5ee329548d7ce	no	17
Warnings:					
Information:					
3	Transmittal Letter	E-IDS.pdf	560184 6a3b0785b66a5cdb8736d1d8794f79921888988b	no	1
Warnings:					
Information:					
4	Information Disclosure Statement (IDS) Filed (SB/08)	sb0008b-6.pdf	190983 3e196d1802f0c1761284a3842ffc2e9df8bcac29	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
		Total Files Size (in bytes):	2048325		

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 : June 3, 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 for Serial No. 10/973,004 dated 8-1-2007 in the United States	
		Office Action for Serial No. 10/973,004 dated 4-10-2007 in the United States	
		Final Office Action for Serial No. 10/973,075 dated 5-12-2009 in the United States	
		Office Action for Serial No. 10/973,075 dated 10-29-2008 in the United State	
		Office Action for Serial No. 10/973,075 dated 3-12-2008 in the United States	
		Office Action for Serial No. 10/973,075 dated 7-2-2007 in the United States	
		Notice of Allowance for Serial No. 10/972,796 dated 12-19-2007 in the United States	
		Office Action for Serial No. 10/972,796 dated 7-12-2007 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.

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 : June 2, 2009
Telephone : (303) 298-9888

Rev 10/08 (S-IDS)

Electronic Acknowledgement Receipt	
EFS ID:	5433682
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:	02-JUN-2009
Filing Date:	25-OCT-2004
Time Stamp:	10:53:12
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	NPL Documents	10972796-OA-7-12-07.pdf	913945 8d9724670d9e1c4998c30b513126f923488 43a5	no	23
Warnings:					
Information:					

2	NPL Documents	10972796-NOA-12-19-07.pdf	393944 [illegible]	no	8
Warnings:					
Information:					
3	NPL Documents	10973075-FOA-5-12-09.pdf	1124143 [illegible]	no	28
Warnings:					
Information:					
4	NPL Documents	10973075-OA-7-2-07.pdf	746284 [illegible]	no	21
Warnings:					
Information:					
5	NPL Documents	10973075-OA-10-29-08.pdf	1015037 [illegible]	no	25
Warnings:					
Information:					
6	NPL Documents	10973075-OA-12-13-07.pdf	645056 [illegible]	no	18
Warnings:					
Information:					
7	NPL Documents	10973004-NOA-8-1-07.pdf	493543 [illegible]	no	12
Warnings:					
Information:					
8	NPL Documents	10973004-OA-4-10-07.pdf	494591 [illegible]	no	13
Warnings:					
Information:					
9	Information Disclosure Statement (IDS) Filed (SB/08)	sb0008b-4.pdf	192604 [illegible]	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
10	Transmittal Letter	E-IDS.pdf	554045 [illegible]	no	1

Warnings:	
Information:	
Total Files Size (in bytes):	6573192

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.

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]
		SHOICHI ARAKI, Real-Time Tracking of Multiple Moving Object Contours in a Moving Camera Image Sequence, Vol. E83-D IEICE Trans. Inf. & Syst, 7-7-2000, 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, 4-5 November 2002, 4 pages	
		JONG RYUL KIM, Automatic Localization and Trackign of Moving Objects Using Adaptive Snake Algorithm, ICICS-PCM 2003, 15-18 December 2003, 5 pages, Singapore	
		JIANHOA MENG, CVEPS-A Compressed Video Editing and Parsing System, 18-11-1996, Pages 1-11	

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 Acknowledgement Receipt

EFS ID:	5440472
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:	02-JUN-2009
Filing Date:	25-OCT-2004
Time Stamp:	18:12:03
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	554045 9b6a3e0d88868577d8d3b3802a57aae19d772e9a	no	1

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	sb0008b-5.pdf	192377 d885648592906aeb4bebf9b93fd14c0c1e2c94f	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	ARAKI.pdf	665693 d1206680a1383db55592855209827e71b995f4e3	no	9
Warnings:					
Information:					
4	NPL Documents	DONG.pdf	341377 0191074218685219fd6918fcf.8d2263e63a6468ee	no	4
Warnings:					
Information:					
5	NPL Documents	JIANG-BIN.pdf	262550 2e8e413e16e6e07dc213f9a56022ca1a97e19b13	no	4
Warnings:					
Information:					
6	NPL Documents	JIANHAO.pdf	891903 9c5795ee6e3e08aaab6f5921989ec251968ad9	no	11
Warnings:					
Information:					
7	NPL Documents	JONG.pdf	377312 5a67249bbbf3aaab1ac98d6e8888d862752c92f6	no	5
Warnings:					
Information:					
		Total Files Size (in bytes):	3285257		

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 : June 2, 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 for Serial No. 10/973,076 dated 11-28-2008 in the United States	
		Notice of Allowance for Serial No. 10/973,076 dated 7-16-2008 in the United States	
		Notice of Allowance for Serial No. 10/973,076 dated 6-3-2008 in the United States	
		Final Office Action for Serial No. 10/973,076 dated 2-27-2008 in the United States	
		Office Action for Serial No. 10/973,076 dated 9-12-2007 in the United States	
		Notice of Allowance for Serial No. 10/972,888 dated 11-14-2007 in the United States	
		Office Action for Serial No. 10/972,888 dated 7-18-2007 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.

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 : June 1, 2009
Telephone : (303) 298-9888

Rev 10/08 (S-IDS)

Electronic Acknowledgement Receipt

EFS ID:	5426049
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:	01-JUN-2009
Filing Date:	25-OCT-2004
Time Stamp:	00:58:50
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	Information Disclosure Statement (IDS) Filed (SB/08)	sb0008b-3.pdf	192644 cd998c1114a6365ef3d8010b38831fe97139 4dd6a	no	1

Warnings:	
Information:	

This is not an USPTO supplied IDS fillable form					
2	NPL Documents	10972888-NOA-11-14-07.pdf	343428 ce4d2287ccc312a139e6e7e4e8bad915a2c820651	no	6
Warnings:					
Information:					
3	NPL Documents	10972888-OA7-18-07.pdf	757427 5801a99cc8aea7e7f7f4137f8d36a78f694f2956f2e	no	20
Warnings:					
Information:					
4	NPL Documents	10973076-FOA-2-27-08.pdf	673034 c3f3437e6e6ce4f6b315c93aa61621af95a6d5e	no	18
Warnings:					
Information:					
5	NPL Documents	10973076-NOA-6-3-08.pdf	1093613 a2f48a8b13dc66ef731e857b195cde22c05f605cc6	no	26
Warnings:					
Information:					
6	NPL Documents	10973076-NOA-7-16-08.pdf	1096462 438e3934e9554e5986d18dc196a8217746fccc9	no	20
Warnings:					
Information:					
7	NPL Documents	10973076-NOA-11-28-08.pdf	1538148 65a0f6ca8162234b9a892e527e4a15be99d8c198	no	29
Warnings:					
Information:					
8	NPL Documents	10973076-OA-9-12-07.pdf	945952 ab199af1f59dda2a7e9b0a4ecc8356e6ed61d1d9982	no	30
Warnings:					
Information:					
9	Transmittal Letter	E-IDS.pdf	547906 9ec5812fc2355d390bfe2345e86e5597dbe5856c65ab	no	1
Warnings:					
Information:					
		Total Files Size (in bytes):	7188614		

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.

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]
		Office Action for Serial No. 10/973,003 dated 3-5-2008 in the United States	
		Office Action for Serial No. 10/973,003 dated 9-12-2007 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 Acknowledgement Receipt

EFS ID:	5417501
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:	29-MAY-2009
Filing Date:	25-OCT-2004
Time Stamp:	08:35:50
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	NPL Documents	10973003_3_5_2008.pdf	636651 ec:c0c500b61c00ec53c2029c63a0e4964386	no	20
Warnings:					
Information:					

2	NPL Documents	10973003_9_12_2007.pdf	826362 [illegible]	no	26
Warnings:					
Information:					
3	Transmittal Letter	E-IDS.pdf	541767 [illegible]	no	1
Warnings:					
Information:					
4	Information Disclosure Statement (IDS) Filed (SB/08)	sb0008b-2.pdf	189925 [illegible]	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
		Total Files Size (in bytes):	2194705		

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 29, 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 for Serial No. 10/972,835 dated 11-20-2008 in the United States	
		Notice of Allowance for Serial No. 10/972,835 dated 8-1-2008 in the United States	
		Office Action for Serial No. 10/972,835 dated 3-3-2008 in the United States	
		Office Action for Serial No. 10/972,835 dated 7-12-2007 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 Acknowledgement Receipt

EFS ID:	5409696
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:	09:14:01
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 087f6f6549b6b948fc8f65c02143088165d55c1678a	no	1

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	sb0008b-1.pdf	190653 8a0c906a89592f94de7a0ee8a9d228d2be9 :17a91	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	10972835-NOA-8-1-2008.pdf	1240979 618ceab2b3971ac349aa3d51eb6db42b1 cc884e	no	25
Warnings:					
Information:					
4	NPL Documents	10972835-NOA-11-20-2008.pdf	930084 9b798d492cbeeb0e9a7 f5d4888e7a6c 098cc	no	15
Warnings:					
Information:					
5	NPL Documents	10972835-OA-3-3-2008.pdf	943392 5482d3316e1d3386d37cdd4c2168c0f1 9d3c	no	21
Warnings:					
Information:					
6	NPL Documents	10972835-OA-7-12-2007.pdf	738688 6f594e9d4f95d28427b8e924ac75486c 8fc9	no	19
Warnings:					
Information:					
		Total Files Size (in bytes):	4579428		

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/08a (01-09)
Approved for use through 02/28/2009. 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

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- 2004-0078651	02-21-2006	Kevin David Safford	
		US- 2006-0248384	11-02-2006	Kevin David Safford	
		US- 6,330,356	12-11-2001	SUNDARESWARAN	
		US- 6,141,770	10-31-2000	STEPHEN FUCHS	
		US- 6,263,452	07-17-2001	DOUGLAS E. JEWETT	
		US- 6,643,387	11-04-2003	Sriram Sethuraman	
		US- 6,735,253	05-11-2004	SHIH-FU CHANG	
		US- 6,983,018	01-03-2006	CHIH-LUNG	
		US- 7,042,943	05-09-2006	Barin Geoffry Haskell	
		US-			
		US-			
		US-			
		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]
		WO 01 45421	06-21-2001	Koninklijke		

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.

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
21. Juni 2001 (21.06.2001)

PCT

(10) Internationale Veröffentlichungsnummer
WO 01/45421 A1

(51) Internationale Patentklassifikation⁷: H04N 7/26

(21) Internationales Aktenzeichen: PCT/EP00/11653

(22) Internationales Anmeldedatum:
23. November 2000 (23.11.2000)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
99204369.5 17. Dezember 1999 (17.12.1999) EP

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(72) Erfinder; und

(75) Erfinder/Anmelder *(nur für US)*: DE HAAN, Gerard [NL/NL]; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). WITTEBROOD, Rimmert, B. [NL/NL]; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). SCHUTTEN, Robert, J. [NL/NL]; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(74) Anwalt: STEENBEEK, Leonardus, J.; Internationaal Octrooibureau B.V., Prof Holstlaan 6, NL-5656 AA Eindhoven (NL).

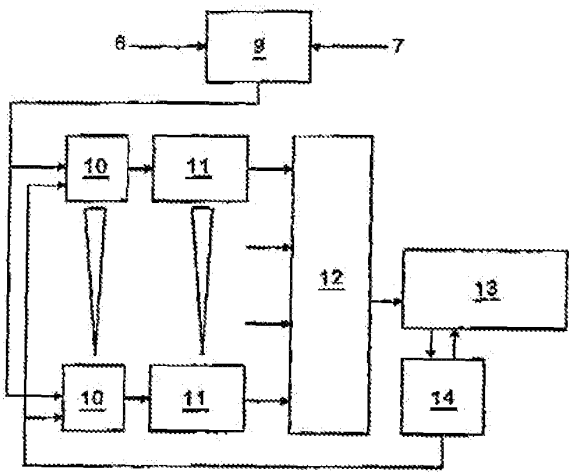
(81) Bestimmungsstaaten *(national)*: JP, KR, US.

(84) Bestimmungsstaaten *(regional)*: europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

[Fortsetzung auf der nächsten Seite]

(54) Title: MOTION ESTIMATION IN VIDEO IMAGES

(54) Bezeichnung: BEWEGUNGSSCHÄTZUNG IN VIDEOBILDERN



(57) Abstract: The invention relates to an object-oriented motion estimation method for the interpolation between video image data sets. According to said method, two or more motion models are used to describe the displacement of image objects between a first and a second video image. The motion models are determined by sets of parameters from which displacement vectors for the image objects can be calculated. One of the motion models is required for taking into account the parts of the image that are static, i.e., stationary. The consistency of the allocation of the moveable image objects to the corresponding image parts in terms of time and space is taken into account when the image area is segmented.

[Fortsetzung auf der nächsten Seite]

WO 01/45421 A1

PAGE 42/62 * RCVD AT 6/21/2006 11:39:02 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/8 * DNIS:2738300 * CSID:6508575487 * DURATION (mm-ss):18-08

Veröffentlicht:
— *Mit internationalem Recherchenbericht.*
— *Vor Ablauf der für Änderungen der Ansprüche geltenden Frist; Veröffentlichung wird wiederholt, falls Änderungen eintreffen.*

Zur Erklärung der Zweibuchstaben-Codes, und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

(57) Zusammenfassung: Die Erfindung betrifft ein objekt-orientiertes Bewegungsschätzungsverfahren für die Interpolation zwischen Videobilddatensätzen. Es werden dabei zwei oder mehr Bewegungsmodelle verwendet, um die Verschiebung von Bildobjekten zwischen einem ersten und einem zweiten Videobild zu beschreiben. Die Bewegungsmodelle werden durch Parametersätze bestimmt, aus welchen sich Verschiebungsvektoren für die Bildobjekte berechnen lassen. Eines der Bewegungsmodelle wird benützt, um diejenigen Bildteile zu berücksichtigen, die statisch, also unbeweglich sind. Bei der Segmentierung des Bildbereiches wird die zeitliche und räumliche Konsistenz der Zuordnung von beweglichen Bildobjekten zu den entsprechenden Bildteilen beachtet.

PAGE 43/62 * RCVD AT 6/21/2006 11:39:02 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/8 * DNIS:2738300 * CSID:6508575487 * DURATION (mm-ss):18-08

Bewegungsschätzung in Videobildern

Die Erfindung betrifft ein Verfahren zur Bewegungsschätzung in Videobilddaten, wobei zunächst ausgehend von einem ersten und einem zweiten Videobild Parametersätze von zwei oder mehr Bewegungsmodellen ermittelt werden und wobei dann den Bewegungsmodellen Bildobjekte zugeordnet werden. Die Erfindung betrifft ferner eine Vorrichtung zur Durchführung des Verfahrens, eine gemäß dem Verfahren arbeitende Vorrichtung zur Darstellung von Videobildern und ein Computerprogrammprodukt zur Bewegungsschätzung.

Fortschritte in der Multimedia-Technik haben zur Entwicklung einer Vielzahl von Videoformaten und Wiedergabestandards geführt. Diese unterscheiden sich unter anderem auch in der Bildrate, also in der Zahl der Einzelbilder pro Zeiteinheit. Soll eine Videosequenz auf einem PC- oder TV-Bildschirm wiedergegeben werden, so ist es erforderlich, eine Anpassung an die Bildrate des Wiedergabegeräts vorzunehmen. Hierfür geeignete Schnittstellen arbeiten mit Konversionsverfahren unterschiedlicher Komplexität. Die einfachste Methode besteht darin, bei der Wiedergabe je nach gewünschter Bildrate Einzelbilder der Videosequenz zu wiederholen oder auszulassen. Bei der Darstellung derartig behandelter Videodaten treten jedoch unerwünschte Artefakte auf. Es kommt je nach dem Verhältnis der zugrundeliegenden Bildraten zu unerwünschten Darstellungsfehlern. Die Wiedergabe erscheint stockend und ungleichmäßig, so daß die in der Videosequenz dargestellten Bewegungen unnatürlich wirken. Aufwendigere Verfahren nehmen eine Interpolation zwischen aufeinanderfolgenden Videobildern vor, wobei ein Algorithmus zur Bewegungsschätzung zum Einsatz kommt, welcher die Verschiebungen einzelner Bildelemente von einem Einzelbild zum nächsten erkennt und daraus Bilddaten erzeugt, die zeitlich zwischen den Einzelbildern der Videosequenz liegen. Die Anwendung derartiger Verfahren im Bereich der Heim-Endgeräte erfordert, daß die zugrundeliegenden Algorithmen eine qualitativ hochwertige Bildratenkonversion liefern und dabei nur einen geringen Rechenaufwand benötigen, da der Leistungsfähigkeit der digitalen Signalverarbeitungselektronik von Heimgeräten Grenzen gesetzt sind.

Verfahren zur Bewegungsschätzung der eingangs genannten Art eignen sich außer für die Bildratenkonversion auch für die Kodierung und Kompression bei der Übertragung von Videodaten sowie für die Tiefeschätzung in der 3D Bilddatenverarbeitung und für Disparitätsschätzungen bei Stereobildern.

Ein derartiges Verfahren schlägt die WO 99/16251 vor. Es handelt sich hierbei um ein leistungsfähiges, objektorientiertes Verfahren zur Bewegungsschätzung, bei dem zwei oder mehr Bewegungsmodelle verwendet werden, um die Verschiebung von Bildobjekten zwischen einem aktuellen und einem vorhergehenden Videobild zu beschreiben. Die Bewegungsmodelle werden durch Parametersätze bestimmt, aus welchen sich Verschiebungsvektoren berechnen lassen. Eines der Bewegungsmodelle wird benötigt, um diejenigen Bildteile zu berücksichtigen, die statisch sind. Der zugehörige Verschiebungsvektor ist also der Nullvektor. Die Parametersätze der übrigen Bewegungsmodelle werden ermittelt, indem die Anpassungsfehler der Bewegungsmodelle bei der Beschreibung der Verschiebung der Bildobjekte zwischen aufeinanderfolgenden Videobildern bewertet werden. Für die Interpolation ist es dann erforderlich, die Bilddaten zu segmentieren und dabei einzelnen Bildobjekten geeignete Bewegungsmodelle zuzuordnen. Das Ergebnis der Segmentierung sind separate Objekte, also Bildteile die vom vorhergehenden zum aktuellen Videobild eine ähnliche oder vergleichbare Verschiebung ausführen.

Das vorbekannte Verfahren zur Bewegungsschätzung stellt eine leistungsfähige Alternative zu den ansonsten üblichen blockorientierten Verfahren dar, da die Zahl der unabhängig voneinander beweglichen Objekte in normalen Videosequenzen gering ist und deswegen auch nur eine entsprechend kleine Zahl von Bewegungsmodellen verarbeitet werden muß. Daraus resultiert eine geringe Zahl von Rechenoperationen, was das Verfahren auch im Heimbereich universell einsetzbar macht.

Die grundlegende Aufgabe der vorliegenden Erfindung ist es, das vorbekannte Bewegungsschätzungsverfahren weiter zu verbessern und dabei gleichzeitig die Komplexität weiter zu reduzieren.

Bei dem Verfahren zur Bewegungsschätzung ist die Segmentierung des Videobildes der schwierigste Schritt. Ziel ist es, jeder Gruppe von Bildpunkten eindeutig ein Bewegungsmodell zuzuordnen. Bei dem vorbekannten Verfahren wird dies dadurch erreicht, daß jedem der typischerweise acht mal acht Bildpunkte großen Blöcke dasjenige

Bewegungsmodell zugeordnet wird, das die Verschiebung im jeweiligen Bildbereich am besten beschreibt. Hierfür wird ein Anpassungsfehler bestimmt, welcher als Summe der absoluten Differenzen der Intensitäten von gemäß dem entsprechenden Bewegungsmodell relativ zueinander verschobenen Bildpunkten berechnet wird. Jedem Block wird dann dasjenige Bewegungsmodell zugeordnet, bei dem der Anpassungsfehler am kleinsten ist.

In der Praxis hat es sich gezeigt, daß es bei dieser Art der Segmentierung, insbesondere wenn eine große Zahl von individuell beweglichen Objekten in der Videosequenz dargestellt werden, zu Zuordnungsfehlern und unerwünschten Bildartefakten kommt.

Gemäß der zugrundeliegenden Aufgabenstellung wird die Qualität der Segmentierung bei einem Verfahren der eingangs genannten Art dadurch verbessert, daß die Bildobjekte aus einer Mehrzahl von Blöcken zusammengesetzt werden, in die der gesamte Bildbereich unterteilt wird, und daß für die Zuordnung der Bewegungsmodelle zu den Blöcken die Übereinstimmung der Bewegungsmodelle mit der Verschiebung innerhalb der jeweiligen Blöcke sowie die räumliche und/oder die zeitliche Konsistenz der Zuordnung bewertet werden.

Erfindungsgemäß werden die Blöcke nicht unbedingt denjenigen Bewegungsmodellen zugeordnet, für die sich der geringste Anpassungsfehler ergibt. Zusätzlich wird die räumliche Konsistenz der Zuordnung bewertet, indem die Bewegung in räumlich benachbarten Bildbereichen berücksichtigt wird. Typische Videosequenzen bestehen aus einer großen Zahl von Einzelbildern. Das Verfahren zur Bewegungsschätzung wird dabei auf Paare von jeweils aufeinanderfolgenden Bildern angewandt. Gemäß der Erfindung werden die einzelnen Bildpaare nicht mehr unabhängig voneinander behandelt, sondern es wird bei der Zuordnung der Bewegungsmodelle zu den Blöcken bewertet, welche Zuordnung bei früheren Bildpaaren der Videosequenz getroffen wurde. Auf diese Weise wird eine verbesserte räumliche und zeitliche Konsistenz bei der Segmentierung erreicht. Insgesamt ergibt sich somit ein rekursives Segmentierungsverfahren, mit welchem eine Interpolation mit deutlich verbesserter Bildqualität ermöglicht wird. Bei wiederholter Anwendung des Verfahrens auf eine Videosequenz ist eine rasche Konvergenz bei der Bestimmung der einzelnen Bildobjekte festzustellen.

Blöcken im Sinne der vorliegenden Erfindung sind, wie oben erwähnt, typischerweise rechteckige oder quadratische Anordnungen von mehreren Bildpunkten. Im Prinzip können die Blöcke aber auch aus jeweils nur einem einzelnen Bildpunkt bestehen.

Die Zahl der Rechenoperationen hängt bei den Verfahren zur Bewegungsschätzung gemäß der Erfindung stark von der Zahl der Parameter, durch welche

die Bewegungsmodelle bestimmt werden, ab. Ist die Zahl der Parameter sehr gering, sind die Bewegungsmodelle also entsprechend einfach, können unter Umständen komplexere Bewegungen nicht beschrieben werden, ohne die Zahl der Bewegungsmodelle zu erhöhen. Dadurch wird jedoch die Zahl der Rechenoperationen wiederum erhöht und außerdem müssen einzelne Bildobjekte in separate Unterobjekte mit jeweils unterschiedlichen Bewegungsmodellen aufgeteilt werden. Dies führt leicht zu Inkonsistenzen und unerwünschten Bildartefakten. Vorteilhafterweise sind die Verfahren zur Bewegungsschätzung gemäß der Erfindung dadurch zu verbessern, daß die Parametersätze der zwei oder mehr Bewegungsmodelle aus jeweils zwei bis vier Parametern bestehen und der Parametersatz eines zusätzlichen Bewegungsmodells aus sechs oder mehr Parametern besteht.

Die Bestimmung von zwei bis vier Parametern ist mit nur geringem Rechenaufwand verbunden. Durch derartig einfache Bewegungsmodelle lassen sich grundlegende Bewegungstypen wie Schwenks, Drehungen und Zoom problemlos beschreiben. Falls im Bildbereich komplexere Bewegungen auftreten, können diese durch Hinzufügen nur eines zusätzlichen Bewegungsmodells mit sechs oder mehr Parametern beschrieben werden, wobei der benötigte Rechenaufwand nur geringfügig ansteigt.

Es ist dabei zweckmäßig, den Parametersatz des zusätzlichen Bewegungsmodelles bei der Zuordnung der Bewegungsmodelle zu den Bildobjekten zu bestimmen.

Wie oben beschrieben, können die Parametersätze der Bewegungsmodelle durch Minimierung eines Anpassungsfehlers, der in ausgewählten Bildbereichen berechnet wird, bestimmt werden. Hochdimensionale Minimierungsverfahren, wie Sie für die Bestimmung von sechs oder mehr Parameter benötigt würden, sind jedoch extrem rechenzeitintensiv. Wesentlich effektiver ist es, die sechs oder mehr Parameter des zusätzlichen Bewegungsmodells erst bei der Segmentierung des Bildbereiches zu ermitteln. In diesem Fall ist das zusätzliche Bewegungsmodell als globales Bewegungsmodell zu verstehen, welches die übrigen Bewegungsmodelle im gesamten Bildbereich ergänzt und dabei lokale Abweichungen der Bewegungsmodelle von der wahren Bewegung der Bildobjekte ausgleicht.

Um Rechenzeit zu sparen, ist es weiter vorteilhaft, den Parametersatz des zusätzlichen Bewegungsmodell ausgehend von den Parametern bei früheren Videobilddaten zur verfeinern. Auf diese Weise wird die Bestimmung der Parameter des komplexen Bewegungsmodells rekursiv verbessert. Experimente haben gezeigt, daß durch das zusätzliche Bewegungsmodell eine deutliche Verbesserung bei der Qualität der Interpolation

erreicht wird, wobei aufgrund der raschen Konvergenz bei der rekursiven Parameterbestimmung der Rechenaufwand nur geringfügig zunimmt.

Für die Durchführung des erfindungsgemäßen Verfahrens eignet sich eine Vorrichtung zur Bewegungsschätzung in Videobilddaten, mit einem digitalen Bildspeicher, in welchem eine aktuelles und ein vorhergehendes Videobild speicherbar sind, mit Mitteln zur Bestimmung von Parametersätzen von zwei oder mehr Bewegungsmodellen und mit Mitteln zur Zuordnung der Bewegungsmodellen zu Bildobjekten, wobei die Vorrichtung Mittel zur Unterteilung des gesamten Bildbereiches in Blöcke, aus denen die Bildobjekte zusammengesetzt werden und zur Zuordnung der Blöcke zu den Bewegungsmodellen aufweist und wobei die Übereinstimmung der Bewegungsmodelle mit der Verschiebung innerhalb der jeweiligen Blöcke sowie die räumliche und zeitliche Konsistenz der Zuordnung bewertet werden. Derartige Vorrichtungen können vorteilhafterweise in Fernsehgeräte, Monitore, Video- oder DVD-Abspielgeräte usw. eingebaut werden. Der digitale Bildspeicher der erfindungsgemäßen Vorrichtung muß dabei nicht notwendigerweise eine ausreichende Kapazität haben, um das erste und das zweite Videobild gleichzeitig aufzunehmen. Die Speicherung der jeweiligen Bilder nacheinander reicht für das erfindungsgemäße Verfahren aus.

Vorrichtungen zur Darstellung von Videobildern, wie beispielsweise Fernsehgeräte, Monitore usw., mit einem digitalen Bildspeicher, in dem Videobilddaten speicherbar sind, mit elektronischen Mitteln zur Verarbeitung der in dem Bildspeicher gespeicherten Bilddaten und zur Darstellung von Videobildern auf einer Anzeigevorrichtung, wobei die Mittel zur Verarbeitung der Bilddaten Mittel zur Bestimmung von Parametersätzen von zwei oder mehr Bewegungsmodellen und Mittel zur Zuordnung der Bewegungsmodelle zu Bildobjekten aufweisen, können vorteilhafterweise von dem erfindungsgemäßen Verfahren profitieren, wenn die Mittel zur Verarbeitung der Bilddaten ferner Mittel zur Unterteilung des gesamten Bildbereiches in Blöcke, aus denen die Bildobjekte zusammengesetzt werden, und zur Zuordnung der Blöcke zu den Bewegungsmodellen aufweisen, wobei die Übereinstimmung der Bewegungsmodelle mit der Verschiebung innerhalb der jeweiligen Blöcke sowie die räumliche und zeitliche Konsistenz der Zuordnung bewertet werden. Herkömmliche digital arbeitende Fernsehgeräte und Monitore können auf einfache Weise nach dem erfindungsgemäßen Verfahren betrieben werden, wodurch die Qualität der Bilddarstellung verbessert wird. Anzeigevorrichtungen im obigen Sinne sind dabei zum Beispiel die bei Fernsehgeräten und Monitoren üblichen Kathodenstrahlröhren oder

Punktmatrixdisplays. Andere Anzeigevorrichtungen zur visuellen Ausgabe von digitalen Bilddaten sind genauso denkbar.

Ein entsprechendes Computerprogrammprodukt zur Bewegungsschätzung in Videobilddaten, welches als Eingabe ein aktuelles und ein vorhergehendes Videobild erhält und davon ausgehend zunächst Parametersätze von zwei oder mehr Bewegungsmodellen berechnet, dann den Bewegungsmodellen Bildobjekte zuordnet und schließlich Bewegungsdaten ausgibt, welche die Verschiebung der Bildobjekte vom vorhergehenden zum aktuellen Bild beschreiben, unterteilt den gesamten Bildbereich in Blöcke, aus denen die Bildobjekte zusammengesetzt werden, wobei es für die Zuordnung der Blöcke zu den Bewegungsmodellen die Übereinstimmung der Bewegungsmodelle mit der Verschiebung innerhalb der jeweiligen Blöcke sowie die räumliche und zeitlich Konsistenz der Zuordnung bewertet. Ein derartiges Computerprogrammprodukt kann beispielsweise eine Software umfassen, welche zur Programmierung von Videogeräten, Fernsehern usw. geeignet ist. Herkömmliche digital arbeitende Geräte können vorteilhafterweise mittels dieser Software erfindungsgemäß verbessert werden. Die Software kann dabei auf Datenträgern, wie Diskette oder CD-ROM vorliegen, oder kann über ein Datennetz (Internet) zum Herunterladen bereitgestellt werden.

Im folgenden werden, auch anhand der Figuren, Ausführungsbeispiele der Erfindung erläutert. Es zeigen:

Fig. 1 Blockdiagramm eines Bewegungsschätzungsverfahrens gemäß der Erfindung;

Fig. 2 Blockdiagramm einer erfindungsgemäßen Vorrichtung zur Darstellung von Videobildern.

Bei der Ermittlung der Parametersätze für die Bewegungsmodelle gemäß der Erfindung wird ein Auswahlkriterium auf ausgewählte Bildbereiche angewandt. Das Auswahlkriterium besteht dabei zum Beispiel in der Bewertung eines Anpassungsfehlers ε. Dieser wird als Summe von absoluten Differenzen von einzelnen bewegungskompensierten Pixelintensitäten zwischen aktuellem und vorhergehendem Videobild folgendermaßen berechnet:

$$\varepsilon(\vec{C}_o, n) = \sum_{\vec{x} \in I(n)} W_o(\vec{x}) \cdot |F_s(\vec{x}, n) - F_s(\vec{x} - \vec{C}_o(\vec{x}, n), n-1)|$$

Es erfolgt eine Summierung über Bildkoordinaten $\vec{x} = \begin{pmatrix} x \\ y \end{pmatrix}$, die in einer Menge $I(n)$ von ausgewählten Bildbereichen enthalten sind. An diesen Bildkoordinaten werden die absoluten Differenzen zwischen den Pixelintensitäten im aktuellen bzw. vorhergehenden Videobild aufsummiert. $F_s(\vec{x},n)$ ist dabei die Pixelintensität an der Bildkoordinate $\vec{x}$ in einem Videobild mit reduziertem Bildraster. Es hat sich nämlich gezeigt, daß bei der Parameterbestimmung die Verwendung eines in der Auflösung reduzierten Bildes (subsampled image) völlig ausreichend ist. Dies führt vorteilhafterweise zu einer erheblichen Reduktion des Rechenaufwandes. Durch den fortlaufenden Index n wird die Nummer des Einzelbildes und damit der Zeitpunkt innerhalb der Videosequenz bezeichnet. $\vec{C}_o(\vec{x},n)$ bezeichnet für das Bildpaar n als aktuelles und $n-1$ als vorhergehendes Videobild den Verschiebungsvektor an der Bildkoordinate $\vec{x}$ gemäß dem Bewegungsmodell mit dem Index o. $W_o(\vec{x})$ stellt einen Wichtungsfaktor dar, mit dem berücksichtigt wird, welchem Bewegungsmodell o die Bildkoordinate $\vec{x}$ bei früheren Bilddaten der Videosequenz zugeordnet wurde. Damit läßt sich eine Verknüpfung zwischen Parameterbestimmung und Bildsegmentierung herstellen, was Vorteile im Hinblick auf die zeitliche Konsistenz der Bewegungsschätzung und die Leistungsfähigkeit des Verfahrens hat.

Mittels des folgenden Bewegungsmodells lassen sich ausgehend von vier Parametern Verschiebungsvektoren berechnen:

$$\vec{C}_o(\vec{x},n) = \begin{pmatrix} s_x(o,n) + x \cdot d_x(o,n) \\ s_y(o,n) + y \cdot d_y(o,n) \end{pmatrix}$$

Es handelt sich hierbei um ein einfaches lineares Modell erster Ordnung, mit dem Translationen und Skalierungen beschreibbar sind. Das Modell wird bestimmt durch den Parametersatz $\vec{P}_o(n) = \left(s_x(o,n), d_x(o,n), s_y(o,n), d_y(o,n)\right)^T$. Der Parametersatz wird so bestimmt, daß der obige Anpassungsfehler für das entsprechende Bewegungsmodell o einen minimalen Wert annimmt. Bei dem Bewegungsschätzungsverfahren gemäß der Erfindung kommen stets wenigstens zwei Bewegungsmodelle zum Einsatz, wovon eines immer den Nullvektor als Parametersatz hat, so daß durch dieses Bewegungsmodell mit dem Verschiebungsvektor $\vec{C}_0(\vec{x},n) = \vec{0}$ die stationären Bildbereiche beschrieben werden.

Der nächste Schritt bei der Bewegungsschätzung gemäß der Erfindung ist die Bildsegmentierung, also die Zuordnung von Bildbereichen zu den Bewegungsmodellen. Hierzu wird zunächst der gesamte Bildbereich in Blöcke unterteilt. In der Praxis haben sich

quadratische Blöcke aus 8 mal 8 Pixeln bewährt. Für alle Bildkoordinaten innerhalb des Blocks an der Stelle $\vec{X}$ gilt dann $\vec{x} \in B(\vec{X})$. Für jeden Block wird wiederum ein Anpassungsfehler an ein Bewegungsmodell o berechnet:

$$\varepsilon_o(\vec{X},n) = \sum_{\vec{x} \in B(\vec{X})} |F_s(\vec{x}+(1-\alpha)\vec{C}_o(\vec{x},n),n) - F_s(\vec{x}-\alpha\vec{C}_o(\vec{x},n),n-1)|$$

Der Zeitpunkt, zudem die Segmentierung gültig sein soll, wird durch α festgelegt. Im einfachsten Fall wird dem Block $\vec{X}$ dasjenige Bewegungsmodell o zugeordnet, für das $\varepsilon_o(\vec{X},n)$ minimal ist. Die Zuordnung wird dann in der Segmentierungsmaske $M(\vec{X},n)$ abgelegt.

Für die erfindungsgemäße Berücksichtigung der räumlichen und zeitlichen Konsistenz der Bildsegmentierung, also der Zuordnung von Bildbereichen zu Bewegungsmodellen, wird dem Anpassungfehler $\varepsilon_o(\vec{X},n)$ ein geeigneter Straf-Faktor $P(\vec{X},n)$ hinzugefügt:

$$\varepsilon_o(\vec{X},n) = \sum_{\vec{x} \in B(\vec{X})} |F_s(\vec{x}+(1-\alpha)\vec{C}_o(\vec{x},n),n) - F_s(\vec{x}-\alpha\vec{C}_o(\vec{x},n),n-1)| + P(\vec{X},n)$$

Der Straf-Faktor nimmt nach der folgenden Regel verschiedene Werte an:

$$P(\vec{X},n) = \begin{cases} P_s, \text{für } M(\vec{X}+\vec{\delta},n) = o \\ P_t, \text{für } M(\vec{X}-\vec{\delta},n-1) = o \\ P_k \quad \text{sonst} \end{cases} \quad \text{mit} \quad \vec{\delta} = \begin{pmatrix} i \\ j \end{pmatrix}, \; i,j = 0,\pm 1$$

Durch relativ kleine Strafwerte P_s und P_t werden diejenigen Zuordnungen bevorzugt, bei denen zeitlich und räumlich benachbarte Blöcke dem gleichen Bewegungsmodell o zugeordnet sind, während alle übrigen Zuordnungen durch einen vergleichsweise großen Wert P_k benachteiligt werden.

Die Figur 1 zeigt schematisch den Ablauf eines Bewegungsschätzungsverfahrens gemäß der Erfindung. Ausgehend von einem vorhergehenden Videobild 6 und einem aktuellen Videobild 7 werden in einem ersten Schritt 9 für die Bestimmung der Parametersätze Bildbereiche ausgewählt. Für eine Mehrzahl von Bewegungsmodellen werden alle diese Bildbereiche mit Wichtungsfaktoren 10 versehen und dann in einem Verfahrensschritt 11, bei dem die Parameter der Bewegungsmodelle nach einem Auswahlkriterium ermittelt werden, weiterverarbeitet. Ausgehend von den vollständig

PAGE 51/62 * RCVD AT 6/21/2006 11:39:02 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/8 * DNIS:2738300 * CSID:6508575487 * DURATION (mm-ss):18-08

bestimmten Bewegungsmodellen wird dann in einem Schritt 12 der gesamte Bildbereich in Blöcke unterteilt und für jeden Block werden die den einzelnen Bewegungsmodellen entsprechenden Verschiebungsvektoren berechnet. Danach erfolgt in 13 die Segmentierung des Bildbereiches, wobei die Zuordnung der Blöcke zu den Bewegungsmodellen vorgenommen wird. In einer dabei entstehenden Segmentierungsmaske 14 werden die Zuordnungen, die beim nächsten Bildpaar in die Wichtung 10 Eingang finden, gespeichert. Auf die gespeicherte Segmentierungsmaske 14 wird dann während der Segmentierung zur Berücksichtigung der zeitlichen und räumlichen Konsistenz zugegriffen.

Die Figur 2 zeigt schematisch den Aufbau eines digital arbeitenden Anzeigegerätes, bei dem es sich z.B. um ein Fernsehgerät oder einen Videomonitor handeln kann. Der Vorrichtung wird ein Videosignal 20 zugeführt, das in einer digitalen Bildverarbeitungseinheit 21 gespeichert und aufbereitet wird. Hierzu weist die Bildverarbeitungseinheit einen Bildspeicher 22, einen Prozessor 23 und einen Programmspeicher 24 auf. Gegebenenfalls können diese Elemente auch, zumindest teilweise, in einem diskreten Bauteil vereinigt sein. Auf dem Prozessor 23 läuft ein in dem Programmspeicher 24 abgelegtes Programm ab, durch das die erfindungsgemäße Bildverarbeitung gesteuert wird. Eine Bilddarstellungseinheit 25 erhält von der Bildverarbeitungseinheit 21 aufbereitete Bilddaten 26 und erzeugt daraus ein Signal 27 zur Ansteuerung einer Kathodenstrahlröhre 28, über welche die Videobilder visuell ausgegeben werden.

PATENTANSPRÜCHE:

1. Verfahren zur Bewegungsschätzung in Videobilddaten, wobei zunächst ausgehend von einem ersten und einem zweiten Videobild (6,7) Parametersätze von zwei oder mehr Bewegungsmodellen ermittelt werden (11) und wobei dann den Bewegungsmodellen Bildobjekte zugeordnet werden (13),
dadurch gekennzeichnet, daß
die Bildobjekte aus einer Mehrzahl von Blöcken zusammengesetzt werden, in die der gesamte Bildbereich unterteilt wird (12), und daß für die Zuordnung der Bewegungsmodelle zu den Blöcken die Übereinstimmung der Bewegungsmodelle mit der Verschiebung innerhalb der jeweiligen Blöcke sowie die räumliche und/oder die zeitliche Konsistenz der Zuordnung bewertet werden.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß für die Bewertung der räumlichen Konsistenz benachbarte Blöcke berücksichtigt werden.

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß für die Bewertung der zeitlichen Konsistenz frühere Videobilddaten in einer Sequenz von Videobildern berücksichtigt werden.

4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Parametersätze der zwei oder mehr Bewegungsmodelle aus jeweils zwei bis vier Parametern bestehen und der Parametersatz eines zusätzlichen Bewegungsmodell aus sechs oder mehr Parametern besteht.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, daß der Parametersatz des zusätzlichen Bewegungsmodells bei der Zuordnung der Bewegungsmodelle zu den Bildobjekten bestimmt wird.

6. Vorrichtung zur Bewegungsschätzung in Videobilddaten, mit einem digitalen Bildspeicher, in dem ein erstes und ein zweites Videobild speicherbar sind, mit Mitteln zur

Bestimmung von Parametersätzen von zwei oder mehr Bewegungsmodellen und mit Mitteln zur Zuordnung der Bewegungsmodelle zu Bildobjekten, gekennzeichnet durch Mittel zur Unterteilung des gesamten Bildbereiches in Blöcke, aus denen die Bildobjekte zusammengesetzt werden, und zur Zuordnung der Blöcke zu den Bewegungsmodellen, wobei die Übereinstimmung der Bewegungsmodelle mit der Verschiebung innerhalb der jeweiligen Blöcke sowie die räumliche und zeitliche Konsistenz der Zuordnung bewertet werden.

7. Vorrichtung zur Darstellung von Videobildern, insbesondere Fernsehgerät oder Monitor, mit einem digitalen Bildspeicher (22), in dem Videobilddaten speicherbar sind, mit elektronischen Mitteln (21,25) zur Verarbeitung der in dem Bildspeicher gespeicherten Bilddaten und zur Darstellung von Videobildern auf einer Anzeigevorrichtung (28), wobei die Mittel (21) zur Verarbeitung der Bilddaten Mittel zur Bestimmung von Parametersätzen von zwei oder mehr Bewegungsmodellen und Mittel zur Zuordnung der Bewegungsmodelle zu Bildobjekten aufweisen, dadurch gekennzeichnet, daß die Mittel (21) zur Verarbeitung der Bilddaten ferner Mittel zur Unterteilung des gesamten Bildbereiches in Blöcke, aus denen die Bildobjekte zusammengesetzt werden, und zur Zuordnung der Blöcke zu den Bewegungsmodellen aufweisen, wobei die Übereinstimmung der Bewegungsmodelle mit der Verschiebung innerhalb der jeweiligen Blöcke sowie die räumliche und zeitliche Konsistenz der Zuordnung bewertet werden.

8. Computerprogrammprodukt zur Bewegungsschätzung in Videobilddaten, welches als Eingabe ein Paar von Videobilddatensätzen erhält und davon ausgehend zunächst Parametersätze von zwei oder mehr Bewegungsmodellen berechnet, dann den Bewegungsmodellen Bildobjekte zuordnet und schließlich Bewegungsdaten ausgibt, welche die Verschiebung der Bildobjekte vom einen zum anderen Bild beschreiben, dadurch gekennzeichnet, daß es den gesamten Bildbereich in Blöcke unterteilt, aus denen die Bildobjekte zusammengesetzt werden, wobei es für die Zuordnung der Blöcke zu den Bewegungsmodellen die Übereinstimmung der Bewegungsmodelle mit der Verschiebung innerhalb der jeweiligen Blöcke sowie die räumliche und zeitliche Konsistenz der Zuordnung bewertet.

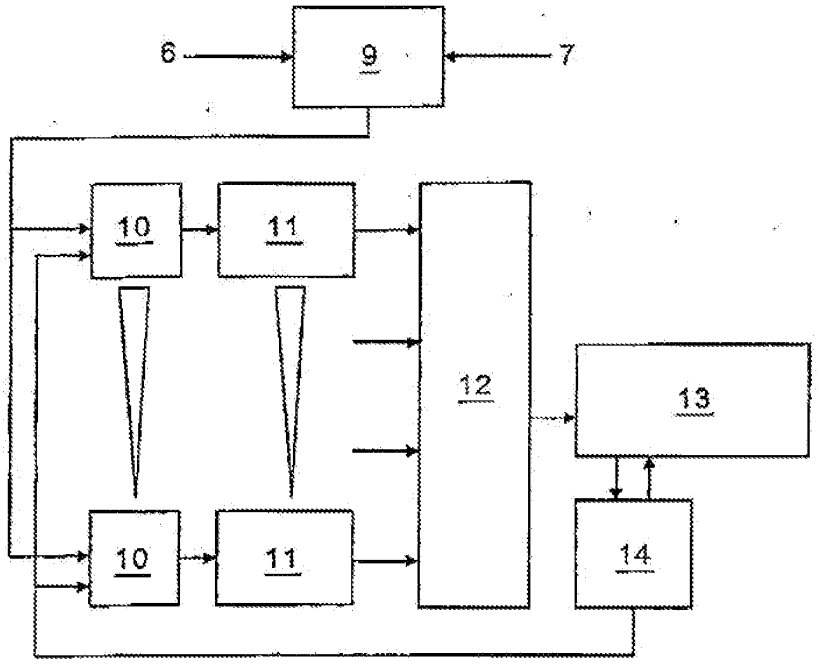


FIG. 1

PAGE 55/62 * RCVD AT 6/21/2006 11:39:02 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/8 * DNIS:2738300 * CSID:6508575487 * DURATION (mm-ss):18-08

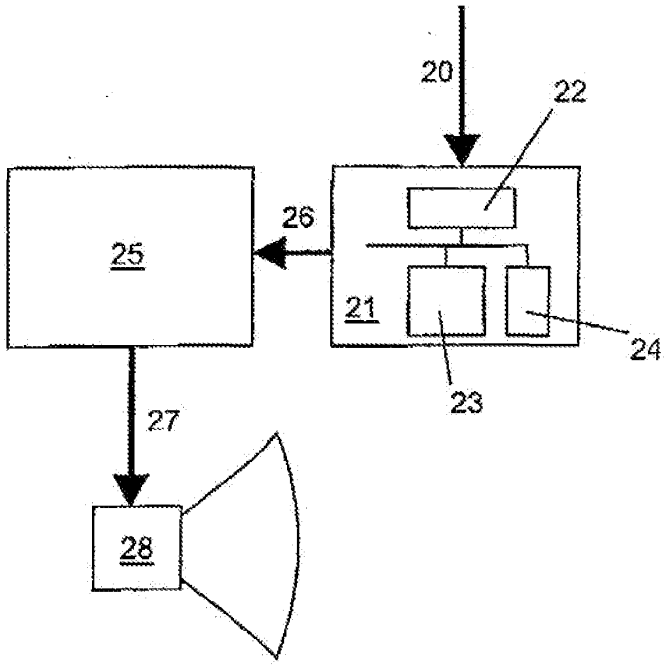


FIG. 2

INTERNATIONAL SEARCH REPORT

Intern nal Application No
PCT/EP 00/11653

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N7/26

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 99 16251 A (KONINKL PHILIPS ELECTRONICS NV ;PHILIPS SVENSKA AB (SE)) 1 April 1999 (1999-04-01) cited in the application claims 1-6	1-8
Y	BLACK M J ET AL: "THE ROBUST ESTIMATION OF MULTIPLE MOTIONS: PARAMETRIC AND PIECEWISE-SMOOTH FLOW FIELDS" COMPUTER VISION AND IMAGE UNDERSTANDING,US,ACADEMIC PRESS, vol. 63, no. 1, 1996, pages 75-104, XP000582617 ISSN: 1077-3142 page 76, right-hand column, paragraph 2 -page 77, left-hand column, paragraph 2.1 page 80, paragraph 2.2 -/--	1-8

[X] Further documents are listed in the continuation of box C. [X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
9 March 2001	12/04/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Foglia, P

Form PCT/ISA/210 (second sheet) (July 1992)

PAGE 57/62 * RCVD AT 6/21/2006 11:39:02 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/8 * DNIS:2738300 * CSID:6508575487 * DURATION (mm-ss):18-08

INTERNATIONAL SEARCH REPORT

Intern. nal Application No
PCT/EP 00/11653

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	WEISS Y ET AL: "A UNIFIED MIXTURE FRAMEWORK FOR MOTION SEGMENTATION: INCORPORATING SPATIAL COHERENCE AND ESTIMATING THE NUMBER OF MODELS" PROCEEDINGS OF THE IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION,US,LOS ALAMITOS, IEEE COMP. SOC. PRESS, 18 June 1996 (1996-06-18), pages 321-326, XP000640250 ISBN: 0-8186-7258-7 page 322, paragraph 2.1 -page 323, paragraph 22	1-8
A	WANG J Y A ET AL: "SPATIO-TEMPORAL SEGMENTATION OF VIDEO DATA" PROCEEDINGS OF THE SPIE, 7 February 1994 (1994-02-07), XP000574646 page 121, paragraph 2.1 -page 122, paragraph 2.2	1-8
A	ELIAS ET AL.: "Image Sequence Segmentation using Mixture-Models with Uniqueness and Spatio-Temporal Consistency Constraints" IEE COLLOQUIUM ON MOTION ANALYSIS AND TRACKING, 10 May 1999 (1999-05-10), pages 6/1-6/6, XP002162507 paragraph '03.2!	1-8
A	US 5 999 651 A (CHANG YUH-LIN ET AL) 7 December 1999 (1999-12-07) column 9, line 39 -column 10, line 45; figures 1,3A,3B	1-8
A	STEINBACH ET AL.: "Using Multiple Global Motion Models for Improved Block-Based Video Coding" ICIP 99, vol. 2, 24 October 1999 (1999-10-24), pages 56-60, XP002162482 page 57, paragraph 2 -page 58, paragraph 4; figure 1	1-8
A	GB 2 317 525 A (NOKIA MOBILE PHONES LTD) 25 March 1998 (1998-03-25) the whole document	1-8

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

PAGE 58/62 * RCVD AT 6/21/2006 11:39:02 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/8 * DNIS:2738300 * CSID:5508575487 * DURATION (mm-ss):18-08

INTERNATIONAL SEARCH REPORT

information on patent family members

Internal Application No
PCT/EP 00/11653

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9916251	A	01-04-1999	EP	0940041 A	08-09-1999
			EP	0943209 A	22-09-1999
			WO	9916248 A	01-04-1999
US 5999651	A	07-12-1999	NONE		
GB 2317525	A	25-03-1998	AU	4133197 A	14-04-1998
			EP	0927494 A	07-07-1999
			WO	9812877 A	26-03-1998

Form PCT/ISA/210 (patent family annex) (July 1992)

INTERNATIONALER RECHERCHENBERICHT

Intern. ...nales Aktenzeichen
PCT/EP 00/11653

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
IPK 7 H04N7/26

Nach der Internationalen Patentklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)
IPK 7 H04N

Recherchierte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)
EPO-Internal, INSPEC, WPI Data, PAJ

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
Y	WO 99 16251 A (KONINKL PHILIPS ELECTRONICS NV ;PHILIPS SVENSKA AB (SE)) 1. April 1999 (1999-04-01) in der Anmeldung erwähnt Ansprüche 1-6	1-8
Y	BLACK M J ET AL: "THE ROBUST ESTIMATION OF MULTIPLE MOTIONS: PARAMETRIC AND PIECEWISE-SMOOTH FLOW FIELDS" COMPUTER VISION AND IMAGE UNDERSTANDING,US,ACADEMIC PRESS, Bd. 63, Nr. 1, 1996, Seiten 75-104, XP000582617 ISSN: 1077-3142 Seite 76, rechte Spalte, Absatz 2 -Seite 77, linke Spalte, Absatz 2.1 Seite 80, Absatz 2.2	1-8

-/--

[X] Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen

[X] Siehe Anhang Patentfamilie

* Besondere Kategorien von angegebenen Veröffentlichungen :
"A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist
"E" älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist
"L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)
"O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht
"P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist

"T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist
"X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden
"Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist
"&" Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche	Absendedatum des internationalen Recherchenberichts
9. März 2001	12/04/2001

Name und Postanschrift der Internationalen Recherchenbehörde	Bevollmächtigter Bediensteter
Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Foglia, P

Formblatt PCT/ISA/210 (Blatt 2) (Juli 1992)

PAGE 60/62 * RCVD AT 6/21/2006 11:39:02 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/8 * DNIS:2738300 * CSID:6508575487 * DURATION (mm-ss):18-08

INTERNATIONALER RECHERCHENBERICHT

Interna...nales Aktenzeichen
PCT/EP 00/11653

C.(Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	WEISS Y ET AL: "A UNIFIED MIXTURE FRAMEWORK FOR MOTION SEGMENTATION: INCORPORATING SPATIAL COHERENCE AND ESTIMATING THE NUMBER OF MODELS" PROCEEDINGS OF THE IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION,US,LOS ALAMITOS, IEEE COMP. SOC. PRESS, 18. Juni 1996 (1996-06-18), Seiten 321-326, XP000640250 ISBN: 0-8186-7258-7 Seite 322, Absatz 2.1 -Seite 323, Absatz 22	1-8
A	WANG J Y A ET AL: "SPATIO-TEMPORAL SEGMENTATION OF VIDEO DATA" PROCEEDINGS OF THE SPIE, 7. Februar 1994 (1994-02-07), XP000574646 Seite 121, Absatz 2.1 -Seite 122, Absatz 2.2	1-8
A	ELIAS ET AL.: "Image Sequence Segmentation using Mixture-Models with Uniqueness and Spatio-Temporal Consistency Constraints" IEE COLLOQUIUM ON MOTION ANALYSIS AND TRACKING, 10. Mai 1999 (1999-05-10), Seiten 6/1-6/6, XP002162507 Absatz '03.2!	1-8
A	US 5 999 651 A (CHANG YUH-LIN ET AL) 7. Dezember 1999 (1999-12-07) Spalte 9, Zeile 39 -Spalte 10, Zeile 45; Abbildungen 1,3A,3B	1-8
A	STEINBACH ET AL.: "Using Multiple Global Motion Models for Improved Block-Based Video Coding" ICIP 99, Bd. 2, 24. Oktober 1999 (1999-10-24), Seiten 56-60, XP002162482 Seite 57, Absatz 2 -Seite 58, Absatz 4; Abbildung 1	1-8
A	GB 2 317 525 A (NOKIA MOBILE PHONES LTD) 25. März 1998 (1998-03-25) das ganze Dokument	1-8

1

Formblatt PCT/ISA/210 (Fortsetzung von Blatt 2) (Juli 1992)

PAGE 61/62 * RCVD AT 6/21/2006 11:39:02 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/8 * DNIS:2738300 * CSID:6508575487 * DURATION (mm-ss):18-08

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Intern... Aktenzeichen
PCT/EP 00/11653

Im Recherchenbericht angeführtes Patentdokument		Datum der Veröffentlichung	Mitglied(er) der Patentfamilie		Datum der Veröffentlichung
WO 9916251	A	01-04-1999	EP	0940041 A	08-09-1999
			EP	0943209 A	22-09-1999
			WO	9916248 A	01-04-1999
US 5999651	A	07-12-1999	KEINE		
GB 2317525	A	25-03-1998	AU	4133197 A	14-04-1998
			EP	0927494 A	07-07-1999
			WO	9812877 A	26-03-1998

Formblatt PCT/ISA/210 (Anhang Patentfamilie)(Juli 1992)

PAGE 62/62 * RCVD AT 6/21/2006 11:39:02 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/8 * DNIS:2738300 * CSID:6508575487 * DURATION (mm-ss):18-08

EXHIBIT
1006

US007251746B2

(12) **United States Patent**
Fox et al.

(10) **Patent No.: US 7,251,746 B2**
(45) **Date of Patent: Jul. 31, 2007**

(54) **AUTONOMOUS FAIL-OVER TO HOT-SPARE PROCESSOR USING SMI**

(75) Inventors: **Thomas James Fox**, Apex, NC (US); **Eric R. Kern**, Durham, NC (US); **Michael Scott Rollins**, Durham, NC (US)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 519 days.

(21) Appl. No.: **10/761,721**

(22) Filed: **Jan. 21, 2004**

(65) **Prior Publication Data**

US 2005/0172164 A1 Aug. 4, 2005

(51) **Int. Cl.**
G06F 11/00 (2006.01)

(52) **U.S. Cl.** **714/13**; 714/11

(58) **Field of Classification Search** 714/13, 714/11, 12

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,251,299 A * 10/1993 Masuda et al. 714/10
5,898,829 A * 4/1999 Morikawa 714/47
6,138,247 A * 10/2000 McKay et al. 714/10
6,240,526 B1 * 5/2001 Petivan et al. 714/11
6,931,568 B2 * 8/2005 Abbondanzio et al. 714/11

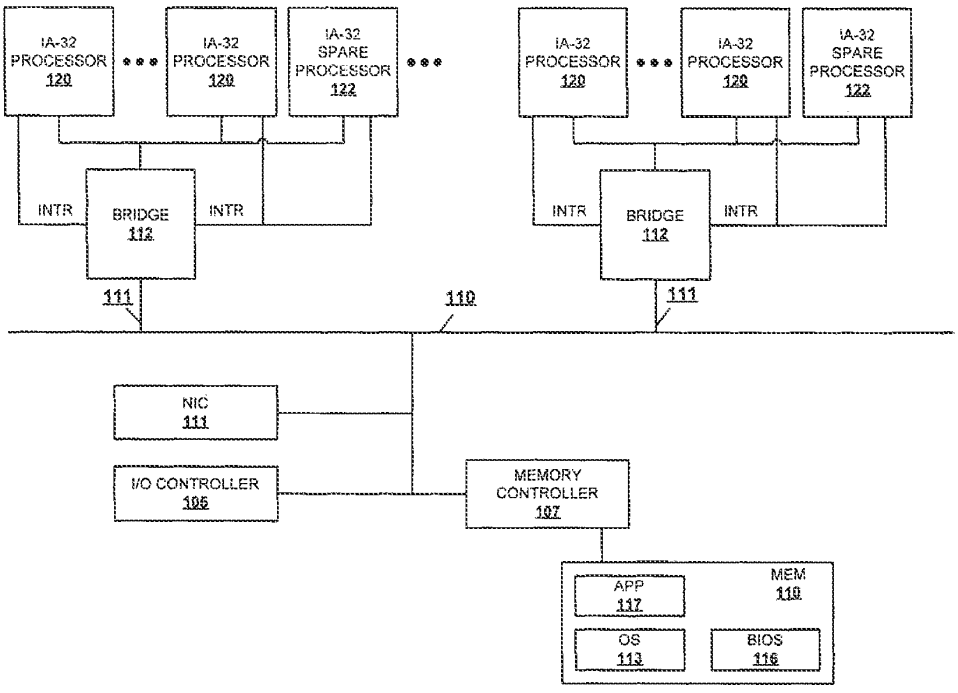
* cited by examiner

Primary Examiner—Dieu-Minh Le
(74) *Attorney, Agent, or Firm*—Cynthia S. Byrd; Dillon & Yudell LLP

(57) **ABSTRACT**

A method and system for dynamically replacing a failing processor in a server system configured with IA-32 architecture without requiring hardware changes to the IA-32 architecture or administrative effort. At least one processor of the multiprocessor system (MP) is initially provided as a reserve (or hot-spare) processor that remains in an idle, off, or low-power mode. While in that mode, the OS is prevented from initially utilizing the hot-spare processor. When a processor failure is detected, SMI code running on a good processor instructs the OS to hold off allocating processes to the failing processor. Contemporaneously, the SMI (and OS) activates and completes an initialization of the hot-spare processor to prepare it to begin receiving the held-off processes. 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.

19 Claims, 4 Drawing Sheets



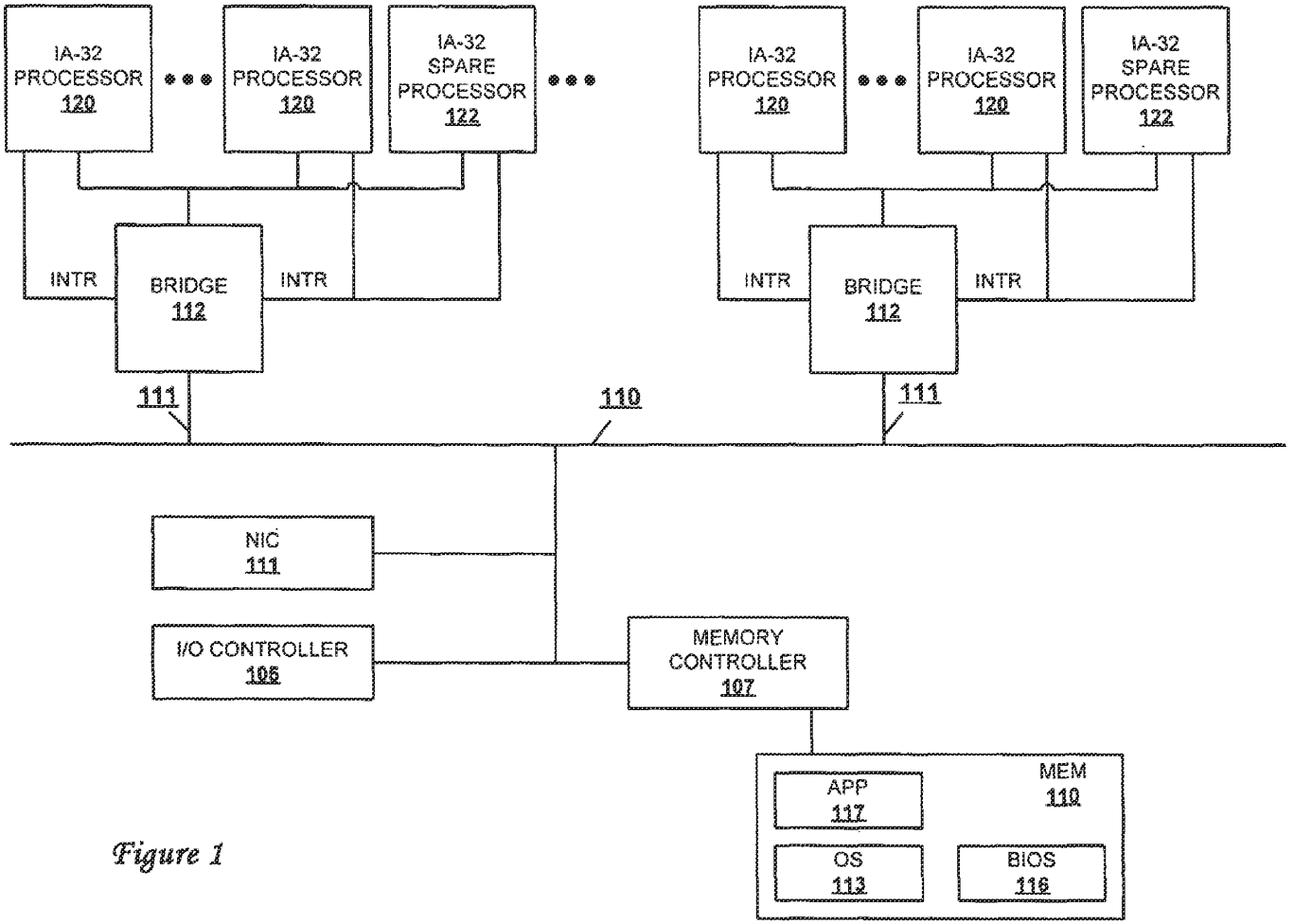

IA-32 PROCESSOR 120
IA-32 PROCESSOR 120
IA-32 SPARE PROCESSOR 122
IA-32 PROCESSOR 120
IA-32 PROCESSOR 120
IA-32 SPARE PROCESSOR 122
INTR
BRIDGE 112
INTR
INTR
BRIDGE 112
INTR
111
110
111
NIC 111
I/O CONTROLLER 106
MEMORY CONTROLLER 107
MEM 110
APP 117
OS 113
BIOS 116

Figure 1

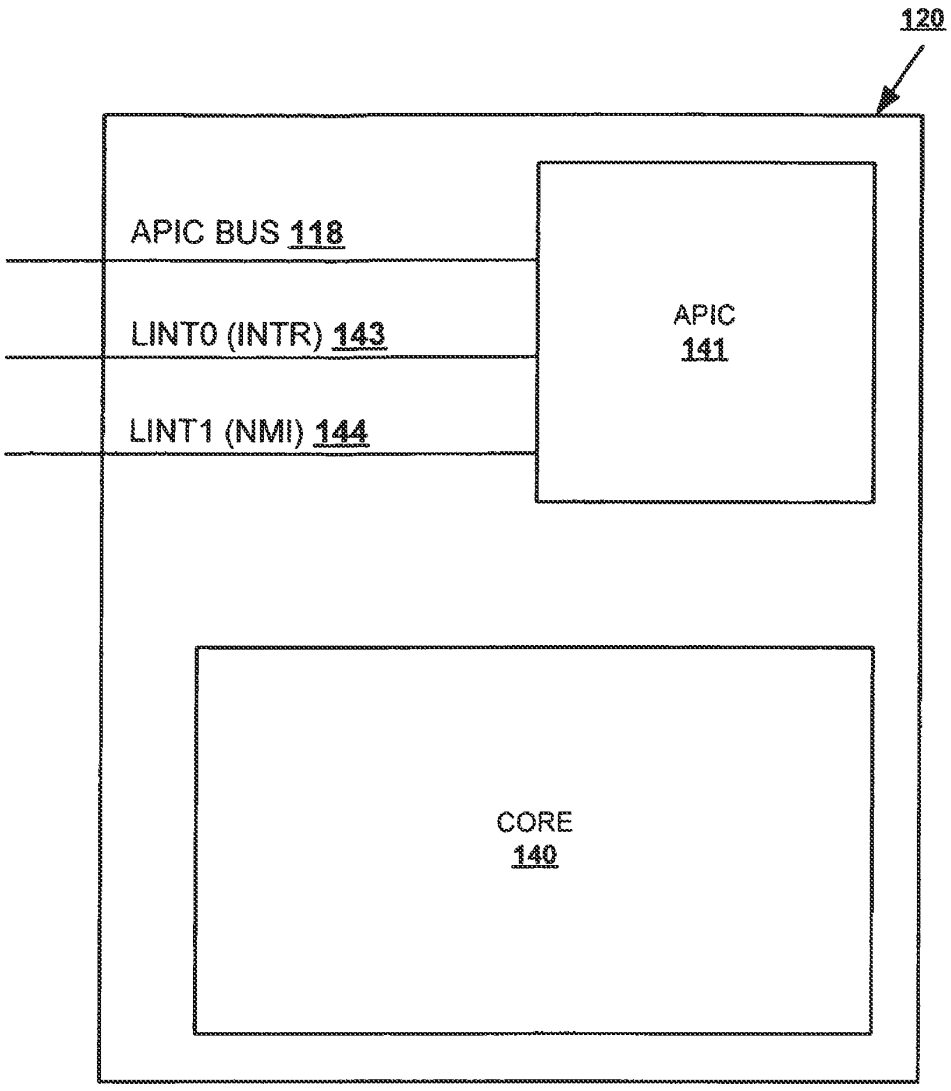

120
APIC BUS 118
LINT0 (INTR) 143
LINT1 (NMI) 144
APIC
141
CORE
140

Figure 2A

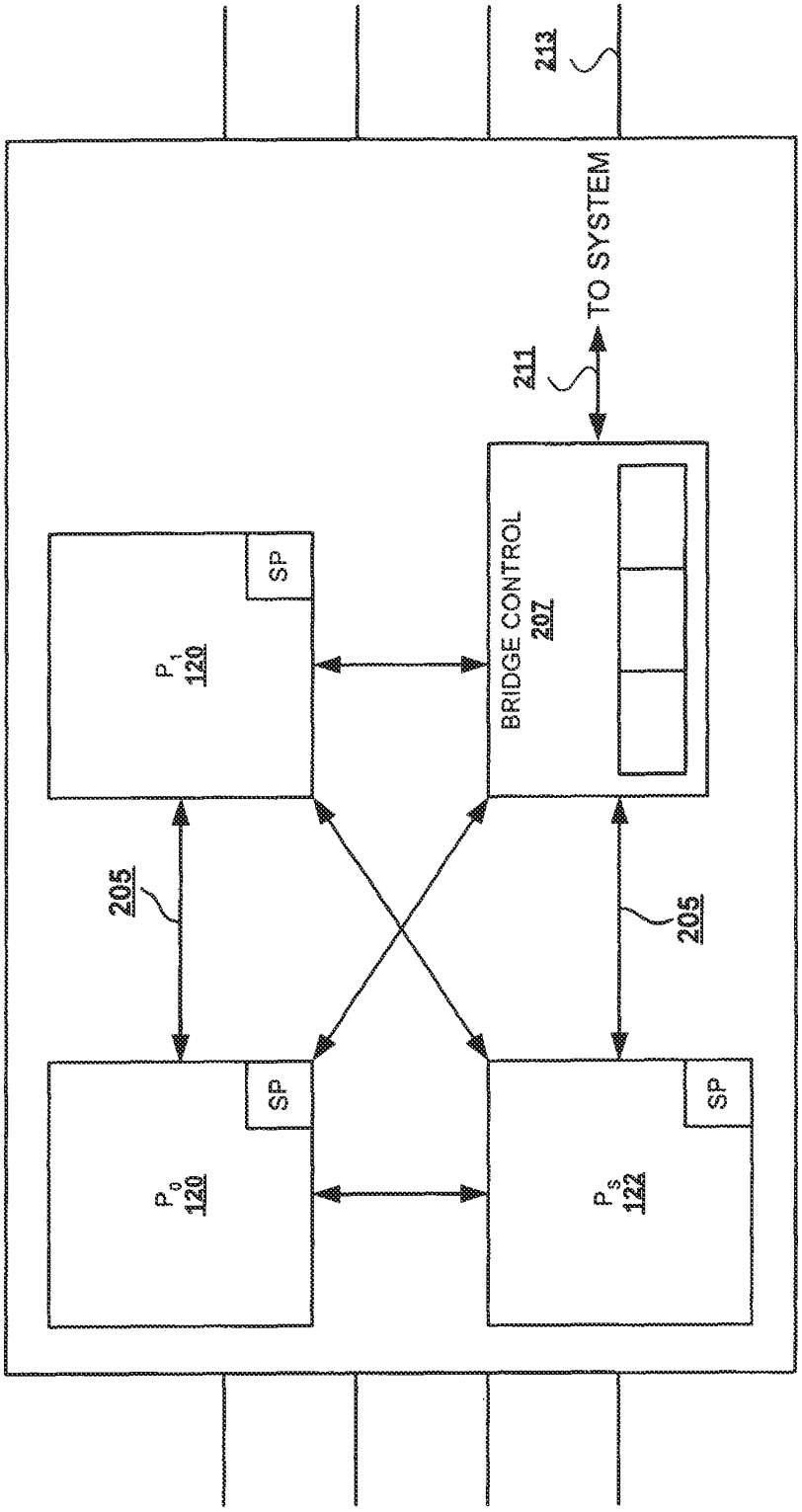

213
TO SYSTEM
211
BRIDGE CONTROL
207
SP
P_1
120
205
205
SP
P_0
120
SP
P_S
122

Figure 2B

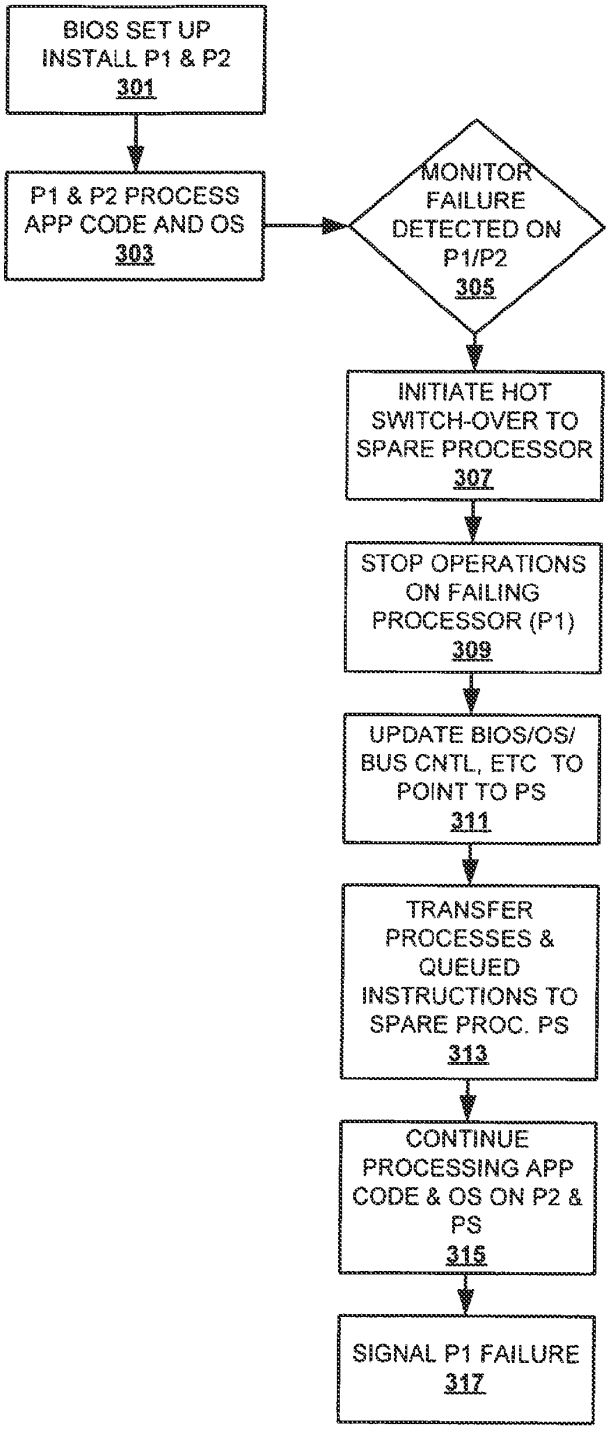

BIOS SET UP
INSTALL P1 & P2
301
P1 & P2 PROCESS
APP CODE AND OS
303
MONITOR
FAILURE
DETECTED ON
P1/P2
305
INITIATE HOT
SWITCH-OVER TO
SPARE PROCESSOR
307
STOP OPERATIONS
ON FAILING
PROCESSOR (P1)
309
UPDATE BIOS/OS/
BUS CNTL, ETC TO
POINT TO PS
311
TRANSFER
PROCESSES &
QUEUED
INSTRUCTIONS TO
SPARE PROC. PS
313
CONTINUE
PROCESSING APP
CODE & OS ON P2 &
PS
315
SIGNAL P1 FAILURE
317

Figure 3

AUTONOMOUS FAIL-OVER TO HOT-SPARE PROCESSOR USING SMI

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention relates generally to data processing systems and in particular to system response to processor failure in a multi-processor data processing system. Still more particularly, the present invention relates to a method and system for dynamically activating a spare processor when processor failure is detected in a multi-processor data processing system.

2. Description of the Related Art

Conventional data processing systems are often configured with multiple processors interconnected to each other and other system components. These multiple processors may exist on a single chip or may be manufactured on separate chips. The processors operate in tandem to efficiently complete tasks associated with application code being executed. Those skilled in the art are familiar with the various configuration and operations of multiprocessor systems (MPs).

Occasionally, while a system is running and the processors are executing application code, one (or more) of the processors may fail (i.e., begin to provide inaccurate processing results and/or begin operating outside of pre-established or "ideal" operating parameters, etc.). When such processor failure occurs, the system's technician has to replace the failing processor (or processor chip) with a new one in order to maintain the level of processing desired for the system. This changing of processors is normally a manual operation, which requires the technician to halt all executing processes (across the system including the operating system (OS)), shut down the MP, obtain a new/replacement processor, complete the switch out of the failing processors, reboot the system, and then restart the executing processes across the MP.

During the system reboot, the replacement processor is recognized by the MP's BIOS (basic input/output system) and activated for operation within the system. Conventionally, the failed (or failing) processor is physically detached (or removed) from the system bus (or interconnect), and the replacement processor is connected (plugged-in) to the interconnect in place of the removed processor. This replacement method is convenient when non-critical processes are being completed on the MP; however, the time required to replace the processor and downtime in processing is unacceptable for critical processes that require continuous up-time of the MP.

Also, with current replacement methods, a separate replacement processor is required to be plugged-in after the failure condition is detected. This requires a technician to swap out the failed processor with the replacement, and as described above, the OS and executing processes are halted until the swap of the processors is completed.

The traditional method of responding to processor failure severely limits the ability of larger systems (e.g., multiprocessor server systems) with non-failing processors to continue executing despite the presence of the failing processor. In lager server systems that require continuous up-time, replacement of a failing processor has to be completed without shutting down the entire system. Typically, in a server system, when a processor begins to fail, the processor must be taken out of the processing pipeline and replaced by another processor to avoid the entire MP crashing. Depending on the built-in redundancy and complexity of the MP, such a temporary removal may have wide ranging effects, from slightly degrading the overall performance of the MP to temporarily removing the MP from service.

Currently, manufacturers of server systems provide different types of server architectures, with common architectures being the S/390 architecture and the Intel Architecture-32 bit (IA-32). The S/390 architecture has a machine instruction for switching to a backup processor, while IA-32 does not have a similar machine instruction. Rather, IA-32 is designed with the functionality to generate an SMI (Systems Management Interrupt) after a CPU fault. Generation of SMIs for standard system management tasks is unique to the IA-32 and the process is described in detailed in U.S. Pat. No. 6,625,679.

Realizing that shutting down the entire MP and then restarting all processes is an unacceptable method of handling single processor failures, manufacturers designed some conventional MPs with a failure response mechanism that involves a hot-spare processor and hardware changes to the system architecture to support processor failure conditions. Implementation of the failure response mechanism is based on the type of server architecture.

U.S. Pat. No. 6,115,829 provides a hot spare processor for solving processor problems relating to the S/390 architecture. The solution involves the utilization of a hardware instruction built into the processor that is usable only by millicode. Since the problems in the S/390 architecture are specific to that architecture, the above solution is not available to the IA-32 architecture, which has a different processor configuration and exhibits a different set of processor problems. Those skilled in the art are familiar with the functional and architectural differences in the two types of architectures and appreciate that different response methods unique to each architecture must be implemented for processor failure.

As another example, U.S. Pat. No. 4,819,232 provides a hardware instruction for software programs to utilize when completing fault recovery to a spare processor from the primary processor. Implementation of this process in an IA-32 system would require architecture (hardware) changes to current IA-32. Another patent, U.S. Pat. No. 5,155,729 provides a redundant processor that engages in a ping-ponging process with the primary processor during a hot swap condition. This process is also specific to the S/390 architecture and not provided within the IA-32 architecture. Finally, U.S. Pat. No. 6,370,657 places the system into standby prior to hot-swapping the processors; however, the response mechanism does not provide a hot-spare nor does it provide a means to keep the OS running during the switch between processors.

With server systems, processor downtime is an undesirable condition, and thus a fast fault-response mechanism/scheme is required. It would be desirable for such a scheme to include the ability to detect when one of the multiple processors has failed. Additionally, when a processor failure has been detected, it would also be desirable for the fault-response mechanism to quickly respond to the failure by providing a replacement processor without the system having to suspend processing and with minimal disruption to the overall system. The present invention recognizes that it would be desirable to provide a processor failure response mechanism that provides a replacement processor in a seamless manner so that executing processes and the OS continue executing during dynamic replacement of the failed processor.

SUMMARY OF THE INVENTION

Disclosed is a method and system for dynamically replacing a failing processor in a server system configured with IA-32 architecture without requiring hardware changes to the IA-32 architecture or administrative effort. The method enables a system to remain operational, i.e., continue executing operating system (OS) processes and application processes on the other processors, while the failing processor is automatically de-activated and a replacement processor is dynamically activated. Negligible system downtime occurs during the transfer of executing threads/processes to the replacement processor, unlike other methods which require temporary halting of the OS processing and, in some instances, halting all other processes across the system.

Implementation of the invention involves the utilization of Systems Management Interrupt (SMI) functionality provided within basic input output system (BIOS) of the IA-32 architecture. At least one processor of the multiprocessor system (MP) is initially provided as a reserve (or hot-spare) processor that remains in an idle, off, or low-power mode. While in that mode, the OS is prevented from initially utilizing the hot-spare processor. When a processor failure is detected, SMI code running on a good processor instructs the OS to hold off allocating processes to the failing processor. Contemporaneously, the SMI (and OS) activates and completes an initialization of the hot-spare processor to prepare it to begin receiving the held-off processes. 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.

Thus, a processor is able to autonomically fail-over to a hot-spare processor without affecting the OS or executing applications. The invention substantially eliminates the problems related to high availability during single processor failure without crashing all of the processors. Further, the invention completes this processor replacement without requiring specialized OS or middleware features. Finally, the invention provides a method for dynamic, hot swap of spare processors without the need of processor architecture changes. That is, no special hardware instruction is built into the processor, as in other methods.

The above as well as additional objects, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

FIG. **1** is a block diagram of a multiprocessor computer system (MP) designed according to IA-32 architecture and a spare processor according to one illustrative embodiment of the present invention;

FIG. **2**A is a block diagram depicting two major internal components of one of the IA-32 processors of FIG. **1**A, according to one illustrative embodiment of the present invention;

FIG. **2**B is a block diagram of a multiprocessor chip designed with dual IA-32 processors and a spare IA-32 processor to enable dynamic switching out of a failed processor during failure response in accordance with one embodiment of the present invention; and

FIG. **3** is a flow chart illustrating the process of detecting and responding to processor failure with the spare processor configuration of FIGS. **1** and **2**B according to one embodiment of the present invention.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENT(S)

The present invention provides a method for dynamically replacing failing processors in a IA-32 configured multiprocessor computer system (MP) by providing the MP with a spare processor that is initially held-off and is later brought into the group of executing processors whenever one of the main processors is detected as failing. The 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 autonomically fail-over to a hot-spare processor without affecting the OS or executing applications. The failure response method of the invention is completed without requiring special hardware instructions built into the processors.

With reference now to the figures and in particular to FIG. **1**, there is illustrated a multiprocessor system (MP) designed with a spare processor and other components that enable the implementation of the various features of the invention. MP **100** is designed according to IA-32 architecture and comprises an IA-32 processor system **120** coupled to system components by a bridge **112**. The IA-32 processor system **11** may include one or more processors **120**, **122** that perform various computing functions. The processors **14** are coupled to a common bus **114**, and may be coupled to a local advanced programmable interrupt controller (APIC) bus (described below). The processors **120**, **122** share access to the common bus **114**, and may also share other resources such as memory, input/output (I/O) devices, and interrupt handlers, for example. The system components provide enhanced functionality, and can be used with the existing IA-32 processors **120**, **122**, provided that appropriate hardware and/or software is used to ensure compatibility between the IA-32 processor system and the system components

MP **100** also comprises a spare processor **122**. The processors are interconnected to each other via processor interconnect **114**. Each processor **120**, **122** comprises service processor logic (not shown), which communicates with other service processors of other processors as well as bus/interconnect logic (e.g., bridge **112**) to allow the processors **120**, **122** to operate in a coherent manner with each other and the rest of the MP **100**.

As shown by FIG. **2**A, each processor **120**, **122** includes a core unit **140** that controls processing on the processor **120**, **122** and an APIC **141** that processes external and normal interrupts that the processor **120**, **122** may receive at its interrupt pins **143** and **144**. APIC **141** also includes a connection to the APIC bus **18**, used to deliver interrupts to the application processors. The interrupt pins **143** and **144** may be programmed to receive specific types of interrupts.

The system components include one or more I/O controllers **105**, which are coupled to a memory controller (MC) **107**. The I/O controllers **105** are coupled to I/O device buses **109**. The buses **109** may include a peripheral computer interface (PCI) bus, for example, the buses **109** may connect standard computer peripherals such as a keyboard.

Coupled to MC **107** is a system memory **110**, within which is stored software for controlling the MP and executing application processes. These software include: the OS **113**, BIOS **116**, which includes SMI, and program applications **117**. SMI notifies the OS of additional processor resources within the MP (i.e., increase/decrease in number of processors) as well as addition/removal of other system resources (i.e., memory and I/O, etc.).

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**.

The bridge **112** takes the interrupts that are being delivered over the system bus **110** and signals the interrupt to the appropriate IA-32 processor **120**, **122**. To ensure that a particular interrupt reaches the appropriate IA-32 processor **120**, **122**, all interrupt transactions on the system bus **110** contain target node **111** and IA-32 processor identification.

Software may initialize the I/O controllers **105** to transfer an interrupt transaction in a manner that identifies the IA-32 processor **120**, **122** to be interrupted. The bridge **112** that is connected to the same processor bus **114** as the destination IA-32 processor **120**, **122** recognizes the interrupt transaction, and asserts an interrupt pin at the targeted IA-32 processor **120**, **122**.

During operation, the OS and software code from the program application are executed by the processors to complete specific tasks or functions associated with the application. The processors are interconnected to the other components (memory, etc.) via system interconnect **110**. Interconnect fabric **110** includes wires and control logic for routing communication between the components as well as controlling the response of the OS of MP **100** to changes in the hardware configuration. In the present invention, BIOS comprises processor failure response logic **207**, which generates the SMI, and configuration setting logic **209**.

FIG. **2**B provides a block diagram representation of a multiprocessor chip designed with main IA-32 processors **120** and a spare processor **122**. Also provided with MP chip **200** is a register **209** for tracking active processors on MP chip **200**, as described below. In one alternate implementation, processors **120**, **122** are interconnected to each other via processor-to-processor buses **203**. Processors **120**, **122** are also connected to bridge (bus controller) **207** via individual control buses **205**. Control buses enable transfer of control data from the bus controller that indicates which processor is active within MP chip **200**. Control bus then provides a connector **211** that couples MP chip **200** to larger processing system such as MP **100** of FIG. **1** via pins **213**.

When the MP is initially activated for processing, one IA-32 processor is held off as a spare processor. Holding off the processor involves having the basic input output system (BIOS) recognize the processor as a spare during BIOS boot (i.e., power on self test (POST), etc. of the MP and having the BIOS respond by not enabling the spare processor to be initially allocated for OS and application processing. This particular set-up of the MP may require some administrative input and support. However, in one embodiment, the processor initialization/identification code includes a software tag to identify the processor as a spare processor to the OS and other components. The other processors are initiated and identified to the OS for regular processing.

In one embodiment, during boot, BIOS utilizes advanced configuration and power interface (ACPI) specification and/or the S3 state to put the processor in a low-power, standby, or "off" state. The S3 state is a computer sleep state defined by the ACPI specification in which the system is not powered nor running, but the memory remains intact, powered and in a continuous refresh cycle. This initial removal of the spare processor from the group of executing processors occurs before the BIOS hands over control to the OS. In another embodiment the system is made to enable the "hot spare processor" feature on-the-fly by the OS with a driver-OS-SMI handshake. Tracking which processors are active may be completed by an software table of active processors, which is available to be read by the OS, when determining work load allocation to each of the active processors.

Notably, the present invention maintains system uptime even while a processor in the MP is failing. This feature of the invention is particularly applicable to operations in a critical server environment that requires continuous uptime of the system.

The spare processor remains hidden until one of the primary processors fails and the spare processor is needed to replace the failing processor. As illustrated in FIG. **2**B, the spare processor may be built on the same board as the main processors. However, in another implementation, the spare processor is provided on a separate processing board to make serviceability easier. For example, with the spare processor operation for Series x440 models, the bottom-most board with CPU's is identified as the ideal board on which to reserve a hot-spare CPU(s), since the bottom CPU board is not as accessible. This enables a failed CPU to be disabled while the OS remains running.

The spare processor is interconnected similarly to each primary processor with the exception that BIOS does not initialize the spare nor initially present the spare processor to the OS following the POST. During POST, BIOS hides the spare processor from the OS. This is accomplished either through a setup of the MP table and/or an ACPI setup. The BIOS brings the spare processor all the way online, then puts it into a spin lock (i.e., a tight software loop in the SMI handler). This has the effect of completely hiding the processor from the OS until the spare processor is needed. Once processor failure occurs and the spare processor is activated, the spare processor is then placed under complete OS control.

Possible triggers for the failure-response features of the invention includes the following: (1) CPU thermal trends are observed by a system management utility, which may be hardware (HW), firmware (FW), and/or software (SW). In one implementation, the system manager utility also provides an SMI handler itself; (2) The CPU temperature goes over or near some operational breaking point; and (3) The system management entity causes an SMI to be invoked, indicating which CPU is to be off-loaded to the hot spare CPU.

Implementation of the invention involves the utilization of Systems Management Interrupt (SMI) functionality provided within basic input output system (BIOS) of the IA-32 systems. At least one processor of the multiprocessor system is reserved as a hot-spare processor that remains in an idle, off, or low-power mode. While in that mode, the OS is prevented from initially utilizing the hot-spare processor. When a processor failure is detected, SMI code running on a good processor instructs the OS to hold off allocating processes to the failing processor. Contemporaneously, the SMI (and OS) activates and completes an initialization of the hot-spare processor to prepare it to begin receiving the held-off processes. 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.

The SMI is generated by hardware during various processor failure conditions. During an SMI, the system state of each processor is completely saved to memory and the caches and pipelines are flushed. A BIOS interrupt handler (i.e., the SMI handler) then takes control of the system until a resume (RSM) instruction is performed by the OS. The SMI handler is utilized to allow BIOS to swap out processors. The BIOS places the failing processor in a halt state and contemporaneously brings the standby processor out of the held-off state and into an active state. Additionally, the BIOS switches the memory save state from the old processor to the new processor. In one embodiment, the above processes are accomplished by providing a software register of active processors in the OS and setting the register to reflect the spare processor as active and the failing processor as inactive.

In the present invention also, SMI allows BIOS to detect a failure and control the reallocation of processing from a good CPU to the hot spare CPU. The SMIs are generated through hardware, and an internal monitoring mechanism is provided for each processor or group of processors that generates the SMI. In the described embodiment, the specific processor-fault SMI is generated when a processor is failing. The SMI identifies to the OS and the BIOS which one of the processors is failing. SMIs are invoked periodically and/or asynchronously by the hardware or other system events, and possibly by software control. A description of the functionality and utilization of the SMI and SMI handler is described in detail in U.S. Pat. No. 6,463,492.

Whenever a CPU fault occurs, an SMI is automatically generated by the existing hardware, as described in the above referenced patent. The SMI is utilized as a trigger that initiates the activation of the spare processor by the BIOS. Notably, unlike with other conventional methods, no hardware switching logic is needed to accomplish the fail-over process of the invention. That is, the invention provides a software-only solution for processor failure response.

Thus, a processor is able to autonomically fail-over to a hot-spare processor without affecting the OS or executing applications. The invention substantially eliminates the problems related to high availability during single processor failure without crashing all of the processors. Further, the invention completes this solution in a way that also does not require specialized OS or middleware features. Finally, the invention provides a method for hot swap spare processors without the need of processor architecture changes. That is, no special hardware instruction is required built into the processor as in other methods.

FIG. **3** is a flow chart illustrating the process of failure detection and response according to one implementation of the invention. For simplicity, the process is described with reference to the MP and processor configuration of FIGS. **1** and **2**A, **2**B. As provided at block **301**, the process begins with the BIOS setting up and installing the main processors, P**0** and P**1**, during the initial POST of the MP. The presence of the spare processor is registered in the BIOS and that processor is initially left in power-down (held-off) or inactive mode. The OS and applications are not provided with an indication of the spare processor, so no operations are initially allocated to that processor. The primary processors commence processing application code and OS operations on the MP, as shown at block **303**. The system monitors the processors for a detection of failure on one of the processors, and a determination is made at block **305** whether a failure is detected. In one implementation, the processors detect failure within their subsystem, generating a processor fault SMI, and message the failure to the SMI handle. If no failure occurs, the processing of application and OS continues as normal on the main processors.

If failure is detected, however, the receipt of a processor fault SMI by the OS immediately activates a switch over response to replace the failing processor with the spare processor as indicated at block **307**. The spare processor is brought out of the held-off state. Concurrently, processes (instructions, threads, etc.) initially allocated to the failing processor are held by the OS and processing on the failing processor is stopped as shown at block **309**. Once the spare processor has been successfully brought on line, the BIOS (table) is updated and the OS, bus controllers, etc. are messaged to change their configurations to reflect a processor change from the failing processor to the spare processor as shown at block **311**. The processes initially allocated to the failing processor are sent to the spare processor as indicated at block **313**, and the application code and OS continue processing on the processors that are now functional (i.e., the other main processor and the spare processor) as provided at block **315**. Notably, for this seamless transfer to be possible, all processors in the system must be identical or process instructions in an identical manner. Otherwise a degradation or improvement in processing may become noticeable and affect overall throughput. Once the fail-over to the spare processor has been completed, a signal/message is generated indicating (to the administrator) that the processor has failed and is replaced with a spare, as shown at block **317**. Then the process ends as depicted at block **319**.

The invention provides a method for hot swap spare processors without the need of processor architecture changes, i.e., without requiring a hardware instruction built into the processor. The invention provides high availability of processing resources and eliminates delays due to personal switching out failing processors manually. Further, the invention completes this solution in a way that also does not require specialized OS or middleware features.

The present invention provides several advantages over other available methods including: (1) no specialized software is needed because SMI is in BIOS, and the reaction to a failing processor is automatic. Also, (2) using an SMI, a service processor is able to identify a failing processor & halt its execution from an outside perspective, unlike the limitation with a software limitation. Furthermore, (3) an alert about a hot-spare processor switching event can be generated.

The other software methods do not utilize SMIs or SMI functionality and encounter several drawbacks, including (1) customers must purchase the OS/software license that provides the hot-spare-processor feature, since not all OSes support this feature; and (2) the OS that has to make the decision to switch to a hot spare processor may be running decision threads atop the faulting processor, leading to incorrect code execution.

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

1. In a multiprocessor data processing system (MP) configured according to IA-32 architecture, a method for dynamically providing spare processor resources when an operating processor fails, said method comprising:

holding-off a spare processor during a POST (power on self test), wherein said spare processor is available within said MP along with at least two operating processors, and said spare processor is not allocated

any processing load by the operating system (OS) following the POST, wherein said MP includes a basic input/output system (BIOS) and a processor register linked to said BIOS, which indicates which processors among said operating processors and said spare processors are currently available to said OS for allocating load, wherein said holding off of the spare processor comprises:

setting a bit within said register corresponding to each of said operating processors to an active state during said initial POST, wherein said active state indicates to said OS that the corresponding operating processor is available for allocating load; and

setting a bit corresponding to said spare processor to an inactive state; and

when one of the operating processors is determined to be failing, dynamically activating said spare processor to replace the failing operating processor, wherein the processing load of the failing processor is automatically sent to the spare processor for processing.

2. The method of claim **1**, wherein said dynamically activating step comprises:

re-setting a bit corresponding to the failing processor to an inactive state; and

setting said bit corresponding to said spare processor to an active state.

3. The method of claim **1**, wherein said holding off step comprises:

placing said spare processor in a low-power, standby state during system boot utilizing advanced configuration and power interface (ACPI) of the BIOS; and

subsequently handing off control of said MP from said BIOS to the OS.

4. The method of claim **1**, wherein said holding off step comprises:

placing said spare processor in an S3 state during system boot utilizing advanced configuration and power interface (ACPI) of the BIOS; and

subsequently handing off control of said MP from said BIOS to the OS.

5. The method of claim **1**, wherein said holding off step further comprises:

bringing said spare processor online during BIOS boot and then placing the spare processor into a spin lock.

6. The method of claim **1**, further comprising generating a message signaling said processor failure to a system administrator.

7. The method of claim **1**, wherein said dynamically activating step comprises completing a driver-OS-SMI handshake operation.

8. In a multiprocessor data processing system (MP) configured according to IA-32 architecture, a method for dynamically providing spare processor resources when an operating processor fails, said method comprising:

holding-off a spare processor during a POST (power on self test), wherein said spare processor is available within said MP along with at least two operating processors, and said spare processor is not allocated any processing load by the operating system (OS) following the POST; and

when one of the operating processors is determined to be failing, dynamically activating said spare processor to replace the failing operating processor, wherein the processing load of the failing processor is automatically sent to the spare processor for processing, wherein said dynamically activating includes:

activating a system management interrupt (SMI) when said failing processor is determined to be failing, wherein said SMI messages said OS to hold off allocation of said processing load to said failing processor;

saving a system state of each processor to memory;

activating an SMI handler to temporarily take control of said system from said OS while said spare processor is being activated;

swapping out the failing processor for the spare processor;

updating system configuration parameters, including bus controller, memory and OS parameters, to reflect switch over to said spare processor from said failing processor; and

restoring control to said OS once said spare processor has been activated.

9. The method of claim **8**, further comprising enabling OS updates to an active processor lists, wherein said OS begins allocating threads previously running on said failed processor to said spare processor.

10. A multiprocessor data processing system (MP) configured according to IA-32 architecture, comprising:

a plurality of processors having at least two main operating processors and a spare processor;

a memory coupled to said plurality of processors;

an operating system (OS) that controls work allocation to each of said plurality of processors;

a basic input output system (BIOS) that supports system management interrupt (SMI);

means for holding-off the spare processor during a POST (power on self test), wherein said spare processor is not allocated any processing load by the operating system (OS) following a power-on self test (POST), said means including:

a processor register linked to said BIOS, which indicates which processors among said operating processors and said spare processors are currently available to said OS for allocating load; and

means for setting a bit within said register corresponding to each of said operating processors to an active state during said initial POST, wherein said active state indicates to said OS that the corresponding operating processor is available for allocating processing load, wherein further a bit of said spare processor is set to an inactive states; and

means, when one of the operating processors is determined to be failing, for dynamically activating said spare processor to replace the failing operating processor, wherein the processing load of the failing processor is automatically sent to the spare processor for processing.

11. The MP of claim **10**, wherein said means for dynamically activating comprises:

means for re-setting a bit corresponding to the failing processor to an inactive state; and

means for setting the bit corresponding to said spare processor to an active state.

12. A multiprocessor data processing system (MP) configured according to IA-32 architecture, comprising:

a plurality of processors having at least two main operating processors and a spare processor;

a memory coupled to said plurality of processors;

an operating system (OS) that controls work allocation to each of said plurality of processors;

a basic input output system (BIOS) that supports system management interrupt (SMI);

means for holding-off the spare processor during a POST (power on self test), wherein said spare processor is not allocated any processing load by the operating system (OS) following a power-on self test (POST); and

means, when one of the operating processors is determined to be failing, for dynamically activating said spare processor to replace the failing operating processor, wherein the processing load of the failing processor is automatically sent to the spare processor for processing, said means including:

means for activating a system management interrupt (SMI) when said failing processor is determined to be failing, wherein said SMI messages said OS to hold off allocation of said processing load to said failing processor.

means for saving a system state of each processor to memory;

means for activating an SMI handler to temporarily take control of said system from said OS while said spare processor is being activated;

means for swapping out the failing processor for the spare processor;

means for updating system configuration parameters, including bus controller, memory and OS parameters, to reflect switch over to said spare processor from said failing processor; and

means for restoring control to said OS once said spare processor has been activated.

13. The MP of claim **12**, wherein said means for holding off comprises:

means for placing said spare processor in a low-power, standby state during system boot utilizing advanced configuration and power interface (ACPI) of the BIOS; and

means for subsequently handing off control of said MP from said BIOS to the OS.

14. The MP of claim **12**, wherein said means for holding off comprises:

means for placing said spare processor in an S3 state during system boot utilizing advanced configuration and power interface (ACPI) of the BIOS; and

means for subsequently handing off control of said MP from said BIOS to the OS.

15. The MP of claim **12**, wherein said means for holding off further comprises BIOS means for bringing said spare processor online and then placing the spare processor into a spin lock.

16. The MP of claim **12**, further comprising generating a message signaling said processor failure to a system administrator.

17. The MP of claim **12**, wherein said means for dynamically activating comprises means for completing a driver-OS-SMI handshake operation.

18. The MP of claim **12**, further comprising means for enabling OS updates to an active processor lists, wherein said OS begins allocating threads previously running on said failed processor to said spare processor.

19. The MP of claim **12**, wherein further:

said plurality of processors are configured with said at least two main processors on a first processing board and said spare processor on a second processing board, wherein said second processing board is a processing board among a plurality of processing boards containing a processor within the MP that is most difficult to access.

* * * * *

EXHIBIT
1015

intel

MultiProcessor Specification

Version 1.4

May 1997

intel

B
Operating System Programming Guidelines

The goal of the MP specification is to transfer enough information about the hardware environment to the operating system that a single, shrink-wrapped, operating-system binary can boot-up and fully utilize a wide variety of multiprocessor systems. The following sections explain how the operating system can take advantage of this specification to handle these operations:

1. Operating-system boot-up.
2. Self configuration.
3. Interrupt mode initialization.
4. Application processor startup.
5. Application processor shutdown.
6. Dynamic interrupt masking.
7. Support for unequal processors.

B.1 Operating System Boot-up

While all processors in an MP-compliant system are functionally identical, one of the processors will be designated as the boot processor (BSP) at system initialization by the system hardware or by the system hardware in conjunction with the BIOS. The rest of the processors are designated as the application processors (APs). The BSP is responsible for booting the operating system. Once the MP operating system is up and running, the BSP functions as an AP.

Usually a processor is designated as the BSP because it is capable of controlling all system hardware, including AP startup and shutdown. The operating system must determine and remember the APIC ID of the designated BSP, so it can keep the BSP operating as the last running processor during system shutdown. The BSP is not necessarily the first processor, especially in fault-tolerant MP systems in which any available processor can be designated as the BSP.

At the time that the first instruction of the operating system is executed, the APs are in the following state:

- The APs have been restrained (either by the BIOS or by the hardware) from executing operating system code.
- The APs are in a halted condition with interrupts disabled. This means that the AP's local APICs are passively monitoring the APIC bus and will react only to INIT or STARTUP interprocessor interrupts (IPIs).

The operating system's first task is to determine whether the system conforms to the MP specification. This is done by searching for the MP floating pointer structure. If a valid floating pointer structure is detected, it indicates that the system is MP-compliant, and the operating system should continue to look for the MP configuration table. If the system is not MP-compliant, the operating system may attempt other means of MP system detection, if it is capable of doing so, or treat the system as a uniprocessor system.

B.2 Operating System Booting and Self-configuration

An MP configuration table is required by the MP specification, with the exception of the default system configurations defined in Chapter 5. The table should be treated as read-only by the operating system. If the MP configuration table exists, the BSP should access the processor entries in the table to configure the operating system.

The BSP should later configure the operating system based on the bus, I/O APIC, IRQs, and system interrupt assignment entries of the configuration table. Note that certain types of buses are mutually exclusive, such as EISA with MCA, or ISA with MCA. The operating system may report such errors in the configuration table, if they occur.

If the MP configuration table does not exist, the BSP configures the operating system for the default system configuration indicated by the default configuration bits of the MP feature information bytes. In this case, only two processors and one I/O APIC exist in the system; both processors have the same type and features.

The operating system contains a set of predefined internal configuration tables that represent the default configurations described in Chapter 5. For MP-compliant systems that use one of the default configurations, the operating system derives the required configuration information from the corresponding predefined table.

To wake up the AP, the BSP should use the universal algorithm defined in Section B.4.

B.3 Interrupt Mode Initialization and Handling

At the time the operating system boots, the interrupt structure is configured for DOS compatibility. The system may be running either in PIC Mode or in Virtual Wire Mode. If it is in PIC Mode, the NMI and INTR will bypass the BSP's local APIC when the Interrupt Mode Configuration Register (IMCR) has a value of zero. The operating system should not try to read the IMCR because it may not exist.

The operating system should switch over to Symmetric I/O Mode to start multiprocessor operation. If the IMCRP bit of the MP feature information bytes is set, the operating system must set the IMCR to APIC mode. The operating system should not write to the IMCR unless the IMCRP bit is set.

Then the operating system should enable its own local APIC, thereby allowing IPI communications with other APIC-based processors. At this time, the APs' local APICs have interrupts disabled. Interrupts must remain disabled at the APs' local APICs while the BSP is enabling the I/O APIC and bringing the system to the normal operating state. Otherwise, an I/O interrupt may be delivered to the uninitialized AP, resulting in the loss of the interrupt.

It is the responsibility of the operating system to assign unique IDs to I/O APIC units.

B.4 Application Processor Startup

An AP may be started either by the BSP or by another active AP. The operating system causes application processors to start executing their initial tasks in the operating system code by using the following universal algorithm. The algorithm detailed below consists of a sequence of interprocessor interrupts and short programmatic delays to allow the APs to respond to the wakeup commands. The algorithm shown here in pseudo-code assumes that the BSP is starting an AP for documentation convenience. The BSP must initialize BIOS shutdown code to 0AH and the warm reset vector (DWORD based at 40:67) to point to the AP startup code prior to executing the following sequence:

```
BSP sends AP an INIT IPI
BSP DELAYs (10mSec)
If (APIC_VERSION is not an 82489DX) {
    BSP sends AP a STARTUP IPI
    BSP DELAYs (200uSEC)
    BSP sends AP a STARTUP IPI
    BSP DELAYs (200uSEC)
}
BSP verifies synchronization with executing AP
```

Example B-1. Universal Start-up Algorithm

If the MP configuration table exists, it provides the IDs of the application processor local APICs. These IDs should be used as the destination addresses in targeted IPIs.

If the MP configuration table does not exist on an MP-compliant system, the system must be of default configuration type. The MP specification requires local APIC IDs to be numbered sequentially, starting at zero for all default configurations. As a result, the BSP can determine the AP's local APIC ID in default, two-processor configurations by reading its own local APIC ID. Since there are only two possible local APIC IDs in this case, zero and one, when the APIC ID of the BSP is one, the APIC ID of the AP is zero, and vice versa. This is important, because a BSP cannot start up an AP unless it already knows the local APIC ID.

Both INIT IPI and STARTUP IPI are open-ended commands. The operating system is responsible for determining whether its local APIC unit successfully dispatches one of these commands. The operating system must do so for an INIT and STARTUP IPI because the APIC either does not automatically retry or guarantee delivery for one of these special messages.

A local APIC unit indicates successful dispatch of an IPI by resetting the Delivery Status bit in the Interrupt Command Register (ICR). The operating system polls the delivery status bit after sending an INIT or STARTUP IPI until the command has been dispatched.

A period of 20 microseconds should be sufficient for IPI dispatch to complete under normal operating conditions. If the IPI is not successfully dispatched, the operating system can abort the command. Alternatively, the operating system can retry the IPI by writing the lower 32-bit double word of the ICR. This "time-out" mechanism can be implemented through an external interrupt, if interrupts are enabled on the processor, or through execution of an instruction or time-stamp counter spin loop.

Optionally, the operating system can make use of the information provided by the integrated APIC error register. Integrated local APIC units on Intel Architecture processors provide an APIC error register that indicates the reason for non-delivery of an APIC message. Reasons for non-delivery include SEND_ACCEPT and RECEIVE_ACCEPT errors that are generated when no processor responds to a message on the APIC bus. An example of this type of failure includes sending a STARTUP IPI to a destination APIC ID for a non-existent processor. Other errors are usually indicative of hardware problems and include SEND and RECEIVE CHECKSUM errors.

The operating system is responsible for determining whether the IPI was received by the targeted AP and executed successfully. For example, the operating system can define a status flag for each processor. After being awakened, a processor sets its status flag, indicating to the operating system that it is present and running. If the status flag does not change value after a certain time, the operating system should treat the processor as not present or not functional.

The following two sections describe specifics relating to the use of INIT and STARTUP IPIs that can be used to guide the implementation of the universal algorithm presented here.

B.4.1 USING INIT IPI

INIT IPIs can be used with systems based on the 82489DX APIC, or on systems that are based on multiple Pentium (735/90, 815/100) processors. INIT IPI is an Interprocessor Interrupt with trigger mode set to level and delivery mode set to "101" (bits 8 to 10 of the ICR). INIT IPIs should always be programmed as level triggered; the operating system must perform two writes to the ICR to assert and then deassert this delivery mode.

An INIT IPI is an IPI that has its delivery mode set to RESET. Upon receiving an INIT IPI, a local APIC causes an INIT at its processor. The processor resets its state, except that caches, floating point unit, and write buffers are not cleared. Then the processor starts executing from a fixed location, which is the reset vector location. To cause the processor to jump to a different location, the INIT IPI must be used as part of a warm-reset.

The warm reset is a feature of every standard PC/AT BIOS. It allows the INIT signal to be asserted without actually causing the processor to run through its entire BIOS initialization procedure (POST). This feature is used, for example, to return an 80286 processor to Real Mode. The key components of warm reset are the following:

- Shutdown code. One of the first actions of the BIOS POST procedure is to read the shutdown code from location 0Fh of the CMOS RAM. This code can have any of several values that indicate the reason that an INIT was performed. A value of 0Ah indicates a warm reset.
- Warm-reset vector. When POST finds a shutdown code of 0Ah, it executes an indirect jump via the warm-reset vector, which is a doubleword pointer in system RAM location 40:67h.

By putting an appropriate pointer in the warm-reset vector, setting the shutdown code to 0Ah, then causing an INIT, the BIOS (or the operating system) can cause the current processor to jump immediately to any location. Because all processors in an MP system share the same system memory, and because the INIT IPI gives one processor the power to cause an INIT at another, the operating system can cause any processor to jump immediately to any location.

B.4.2 USING STARTUP IPI

STARTUP IPIs are used with systems based on Intel processors with local APIC versions of 1.x or higher. These local APICs recognize the STARTUP IPI, which is an APIC Interprocessor Interrupt with trigger mode set to edge and delivery mode set to "110" (bits 8 through 10 of the ICR).

The STARTUP IPI causes the target processor to start executing in Real Mode from address 000VV000h, where VV is an 8-bit vector that is part of the IPI message. Startup vectors are limited to a 4-kilobyte page boundary in the first megabyte of the address space. Vectors A0-BF are reserved; do not use vectors in this range. STARTUP IPIs are not maskable, do not cause any change of state in the target processor (except for the change to the instruction pointer), and can be issued only one time after RESET or after an INIT IPI reception or pin assertion. A STARTUP IPI neither requires the targeted APIC to be enabled nor the interrupt table to be programmed. If the target processor is in the halted state immediately after RESET or INIT, a STARTUP IPI causes it to leave that state and start executing. The effect is to set CS:IP to VV00:0000h.

For an operating system to use a STARTUP IPI to wake up an AP, the address of the AP initialization routine (or of a branch to that routine) must be in the form of 000VV000h. Sending a STARTUP IPI with VV as its vector causes the AP to jump immediately to and begin executing the operating system's AP initialization routine.

The operating system should not issue a STARTUP IPI to an 82489DX since the STARTUP IPI will be ignored instead of forcing the targeted processor to execute from the given address.

B.5 AP Shutdown Handling

An AP may be shut down by itself, by the BSP, or by another active AP. Shutting down an AP with an active task or a bound device driver is not permitted. Only the BSP may shut itself down, and it must be the last processor to shut down.

There are several possible alternatives for shutdown handling. Two of the possibilities are presented here by way of example.

In the simplest case, the operating system can define an IPI for shutting down and taking APs off-line. The exact usage and behavior of an IPI that is defined for this purpose is operating system specific and is not defined in this specification.

Should the BSP need to take an AP off-line or to place the AP back into a HALT state, the BSP can also use the INIT IPI with Warm Restart. The operating system places the address of a HLT instruction in the warm-reset vector (40:67), sets the CMOS shut-down code to 0Ah, then sends an INIT IPI to the AP. The INIT IPI causes the AP to enter the BIOS POST routine, where it immediately jumps to the warm-reset vector and executes the operating system's HLT instruction. Only one processor can execute the shutdown routine at any given time, due to the use of the shutdown code. The operating system must not rely on any code being executed after the delivery

EXHIBIT
1008

US 20040123201A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0123201 A1

Nguyen et al. (43) Pub. Date: Jun. 24, 2004

(54) ON-DIE MECHANISM FOR HIGH-RELIABILITY PROCESSOR

(76) Inventors: Hang T. Nguyen, Tempe, AZ (US); Steven J. Tu, Phoenix, AZ (US); Alexander J. Honcharik, Holden, MA (US); Sujat Jamil, Chandler, AZ (US)

Correspondence Address:
Leo V. Novakoski
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026 (US)

(21) Appl. No.: 10/324,957

(22) Filed: Dec. 19, 2002

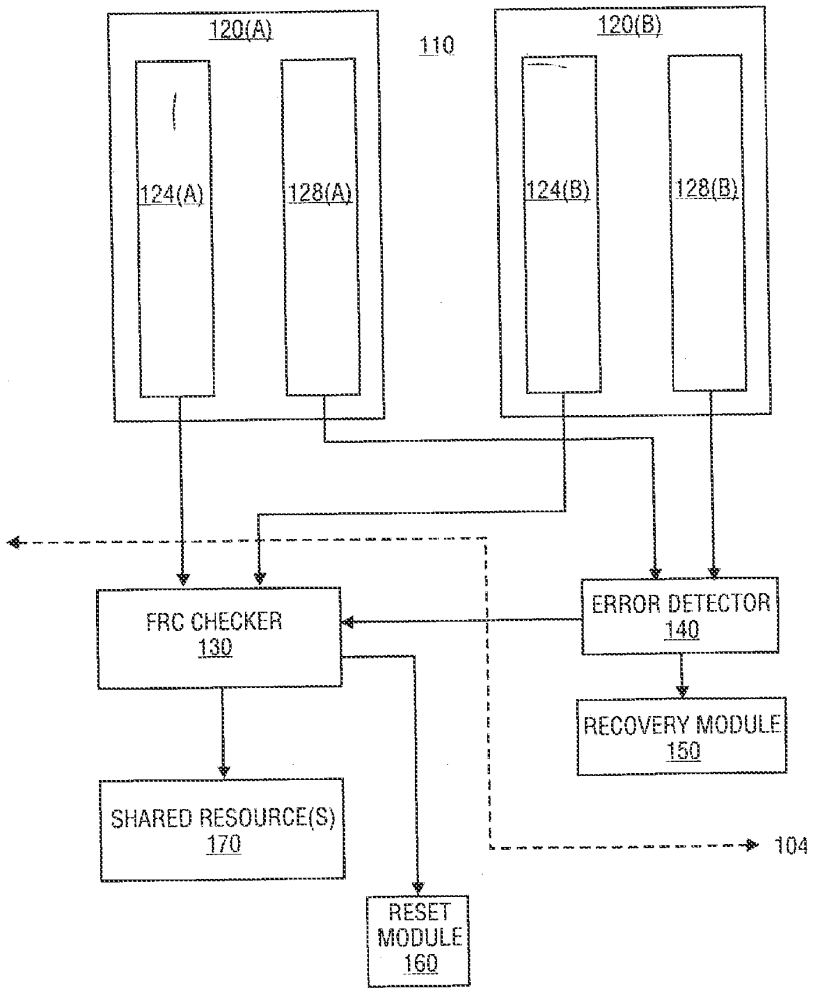
Publication Classification

(51) Int. Cl.[7] G01R 31/28; G06F 11/00

(52) U.S. Cl. 714/736

(57) ABSTRACT

A processor includes first and second execution cores that operate in a redundant (FRC) mode, an FRC check unit to compare results from the first and second execution cores, and an error check unit to detect recoverable errors in the first and second cores. The error detector disables the FRC checker, responsive to detection of a recoverable error. A multi-mode embodiment of the processor implements a multi-core mode in addition to the FRC mode. An arbitration unit regulates access to resources shared by the first and second execution cores in multi-core mode. The FRC checker is located proximate to the arbitration unit in the multi-mode embodiment.



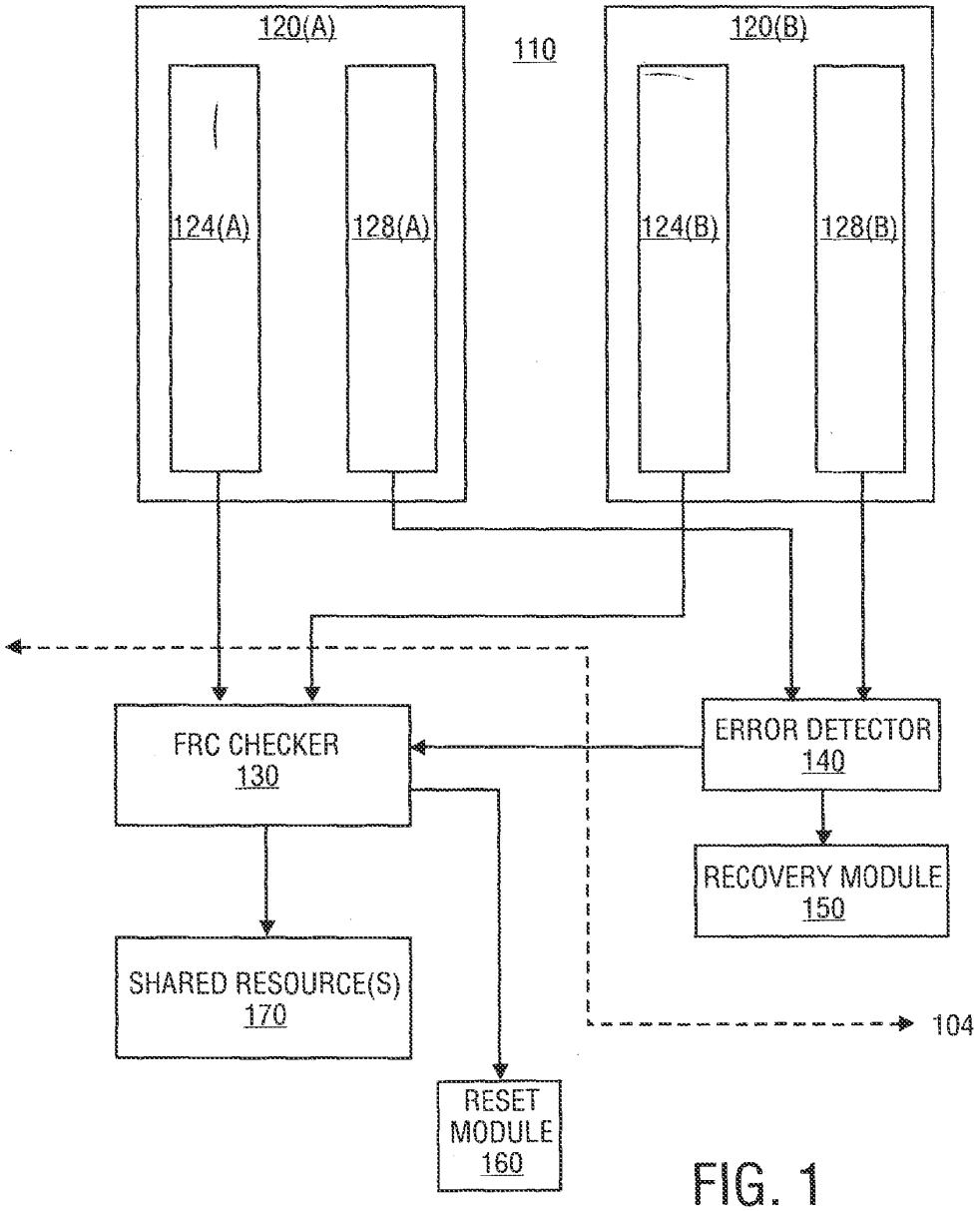

120(A)
110
120(B)
124(A)
128(A)
124(B)
128(B)
FRC CHECKER
130
ERROR DETECTOR
140
RECOVERY MODULE
150
SHARED RESOURCE(S)
170
104
RESET
MODULE
160

FIG. 1

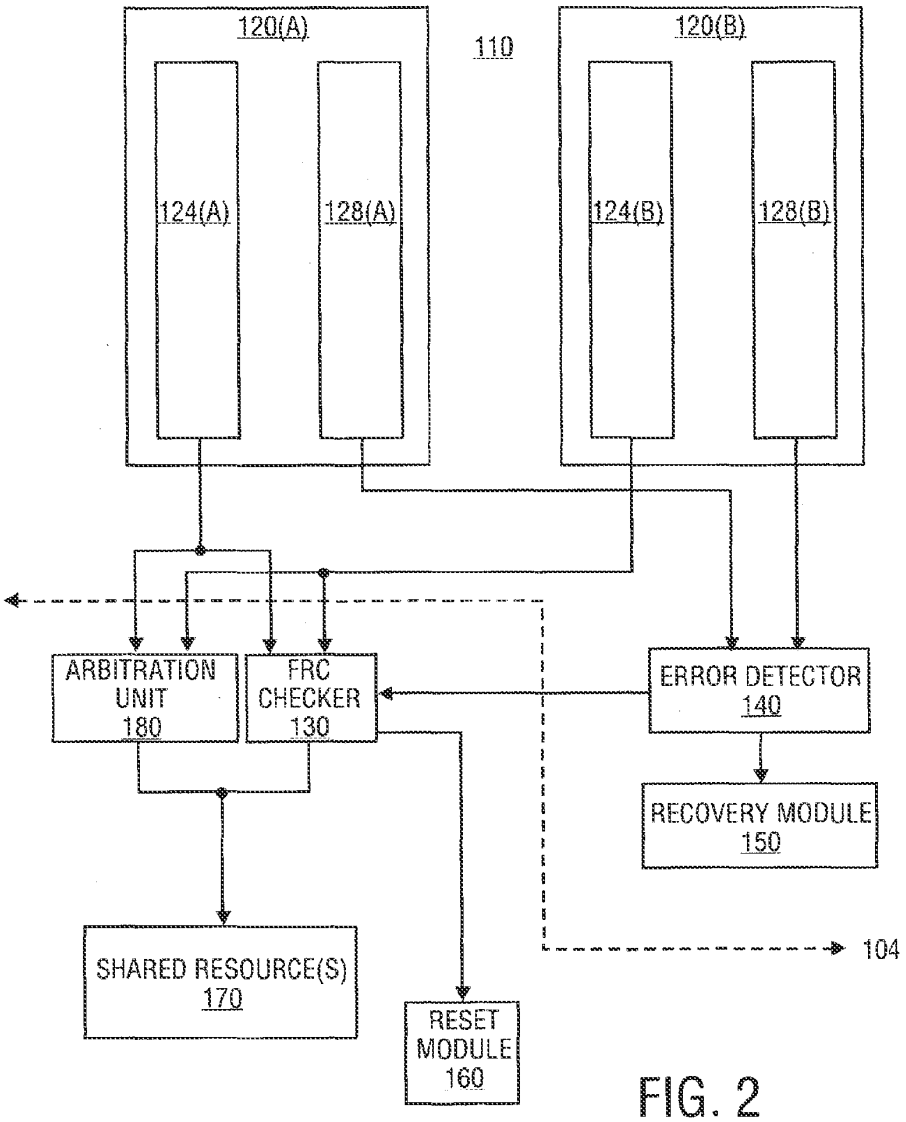

120(A)
110
120(B)
124(A)
128(A)
124(B)
128(B)
ARBITRATION UNIT 180
FRC CHECKER 130
ERROR DETECTOR 140
RECOVERY MODULE 150
SHARED RESOURCE(S) 170
RESET MODULE 160
104

FIG. 2

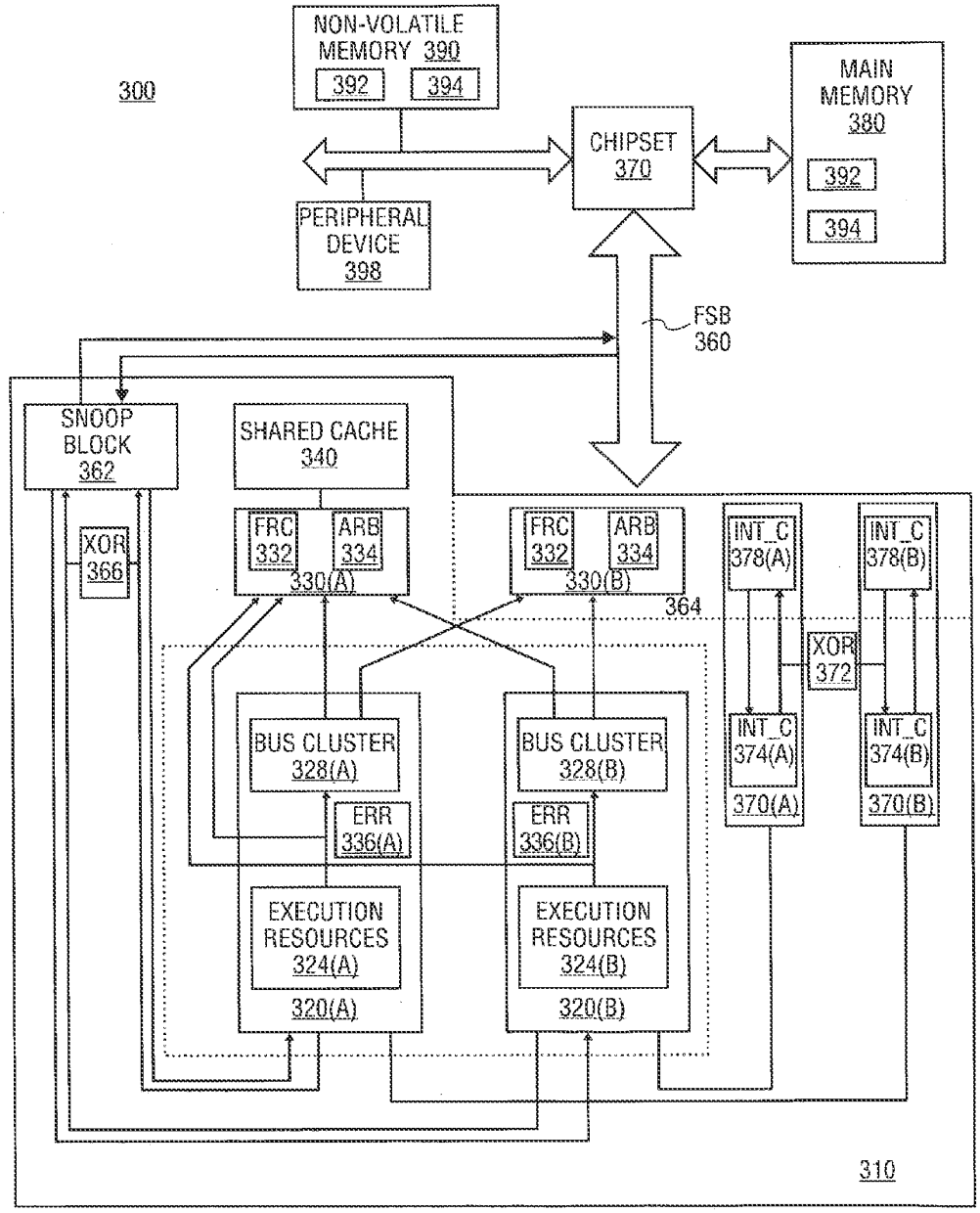

300
NON-VOLATILE MEMORY 390
392
394
CHIPSET 370
MAIN MEMORY 380
392
394
PERIPHERAL DEVICE 398
FSB 360
SNOOP BLOCK 362
SHARED CACHE 340
XOR 366
FRC 332
ARB 334
330(A)
FRC 332
ARB 334
330(B)
364
INT_C 378(A)
INT_C 378(B)
XOR 372
BUS CLUSTER 328(A)
BUS CLUSTER 328(B)
INT_C 374(A)
INT_C 374(B)
ERR 336(A)
ERR 336(B)
370(A)
370(B)
EXECUTION RESOURCES 324(A)
EXECUTION RESOURCES 324(B)
320(A)
320(B)
310

FIG. 3A

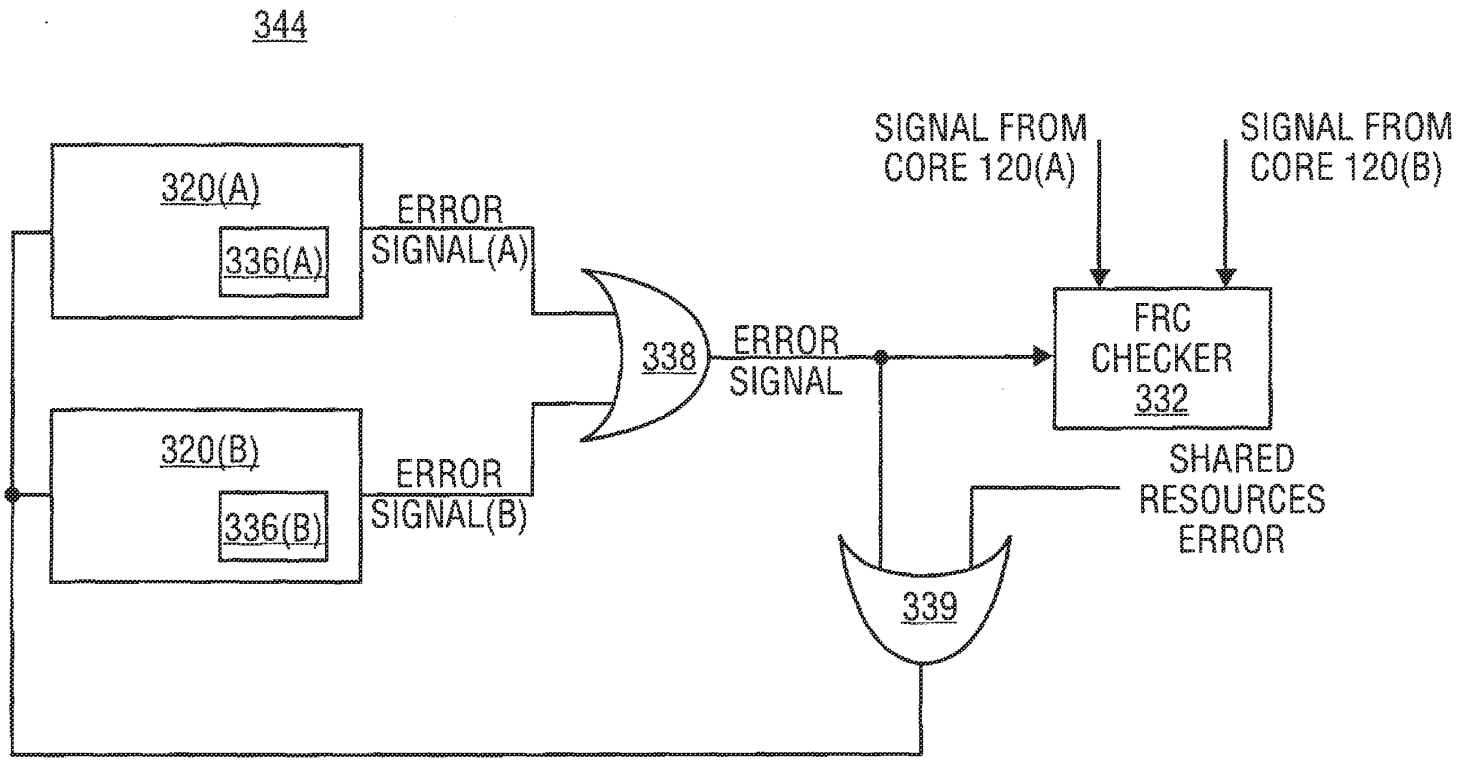

344
320(A)
336(A)
ERROR SIGNAL(A)
320(B)
336(B)
ERROR SIGNAL(B)
338
ERROR SIGNAL
SIGNAL FROM CORE 120(A)
SIGNAL FROM CORE 120(B)
FRC CHECKER 332
SHARED RESOURCES ERROR
339

FIG. 3B

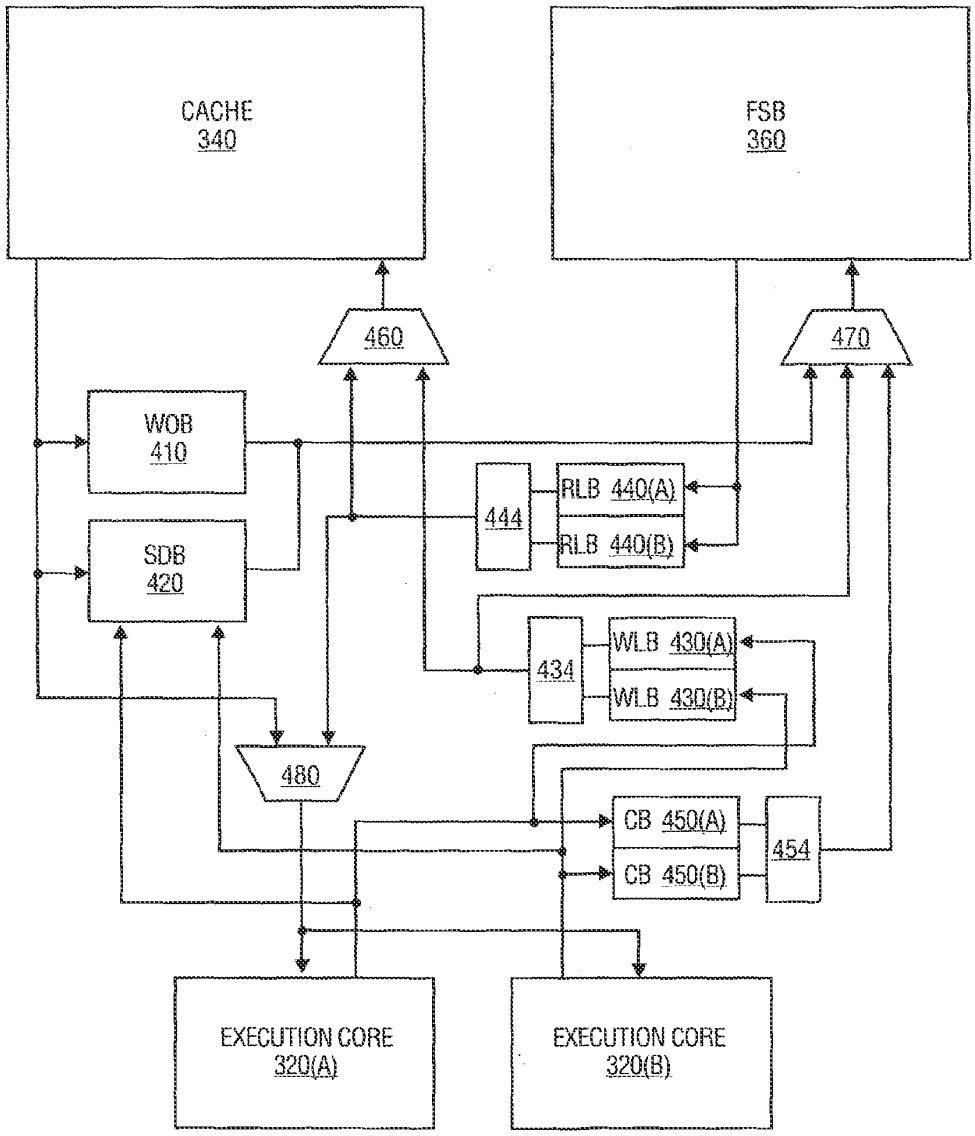

CACHE
340
FSB
360
460
470
WOB
410
SDB
420
444
RLB 440(A)
RLB 440(B)
434
WLB 430(A)
WLB 430(B)
480
CB 450(A)
CB 450(B)
454
EXECUTION CORE
320(A)
EXECUTION CORE
320(B)

FIG. 4

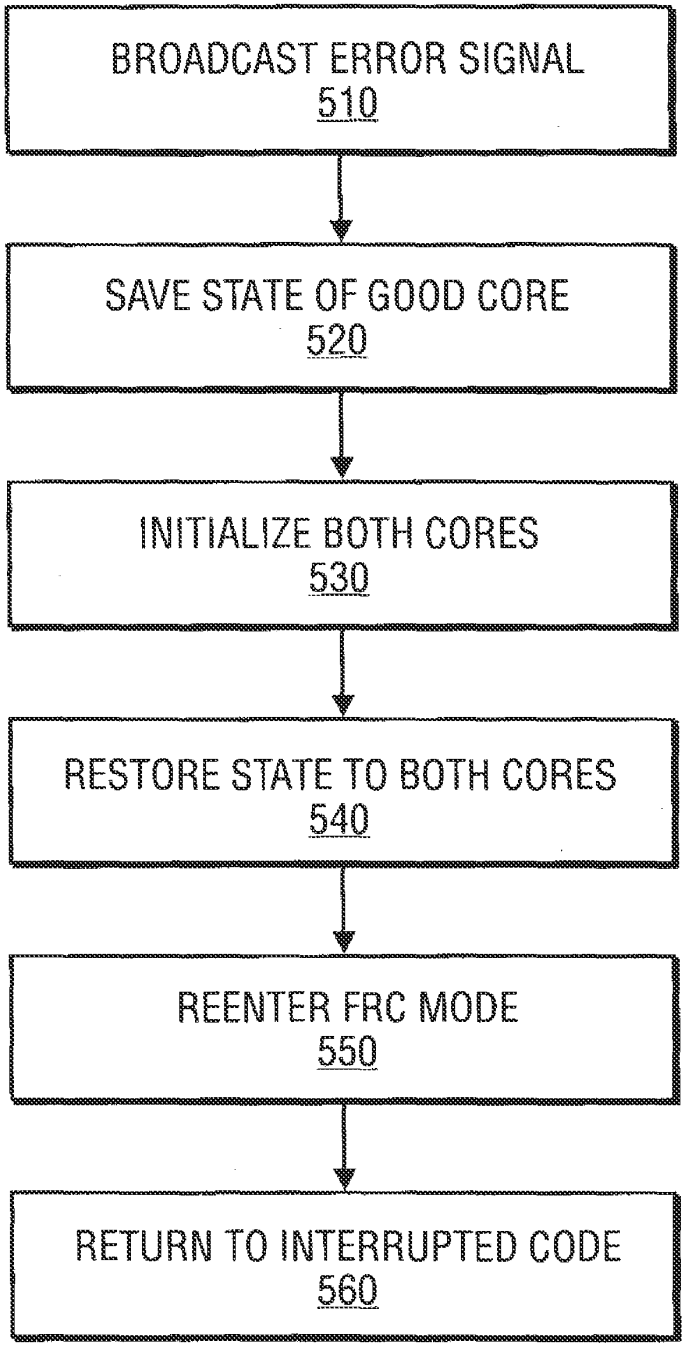

BROADCAST ERROR SIGNAL
510
SAVE STATE OF GOOD CORE
520
INITIALIZE BOTH CORES
530
RESTORE STATE TO BOTH CORES
540
REENTER FRC MODE
550
RETURN TO INTERRUPTED CODE
560

FIG. 5

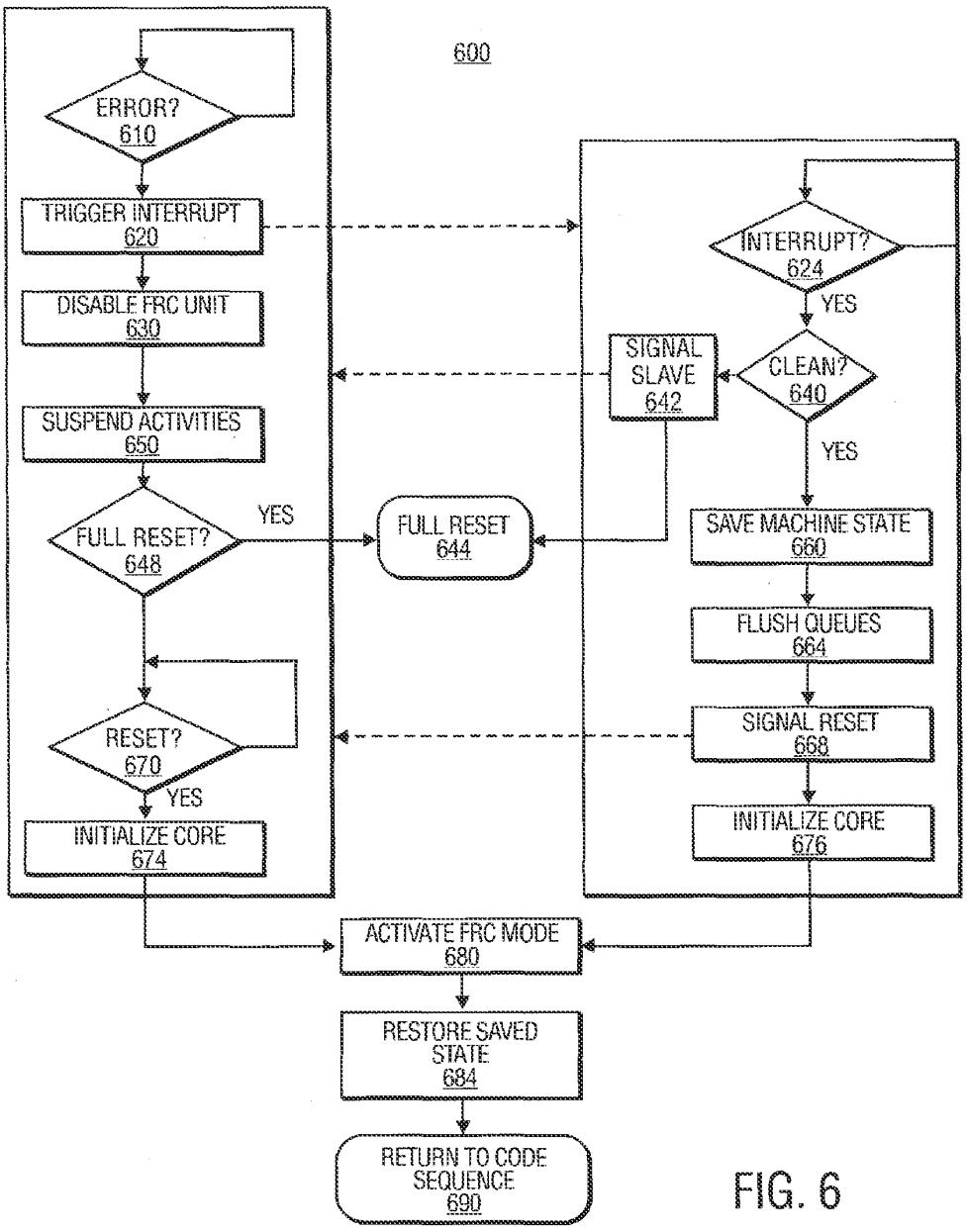

600
ERROR? 610
TRIGGER INTERRUPT 620
DISABLE FRC UNIT 630
SUSPEND ACTIVITIES 650
FULL RESET? 648
YES
FULL RESET 644
RESET? 670
YES
INITIALIZE CORE 674
INTERRUPT? 624
YES
SIGNAL SLAVE 642
CLEAN? 640
YES
SAVE MACHINE STATE 660
FLUSH QUEUES 664
SIGNAL RESET 668
INITIALIZE CORE 676
ACTIVATE FRC MODE 680
RESTORE SAVED STATE 684
RETURN TO CODE SEQUENCE 690

FIG. 6

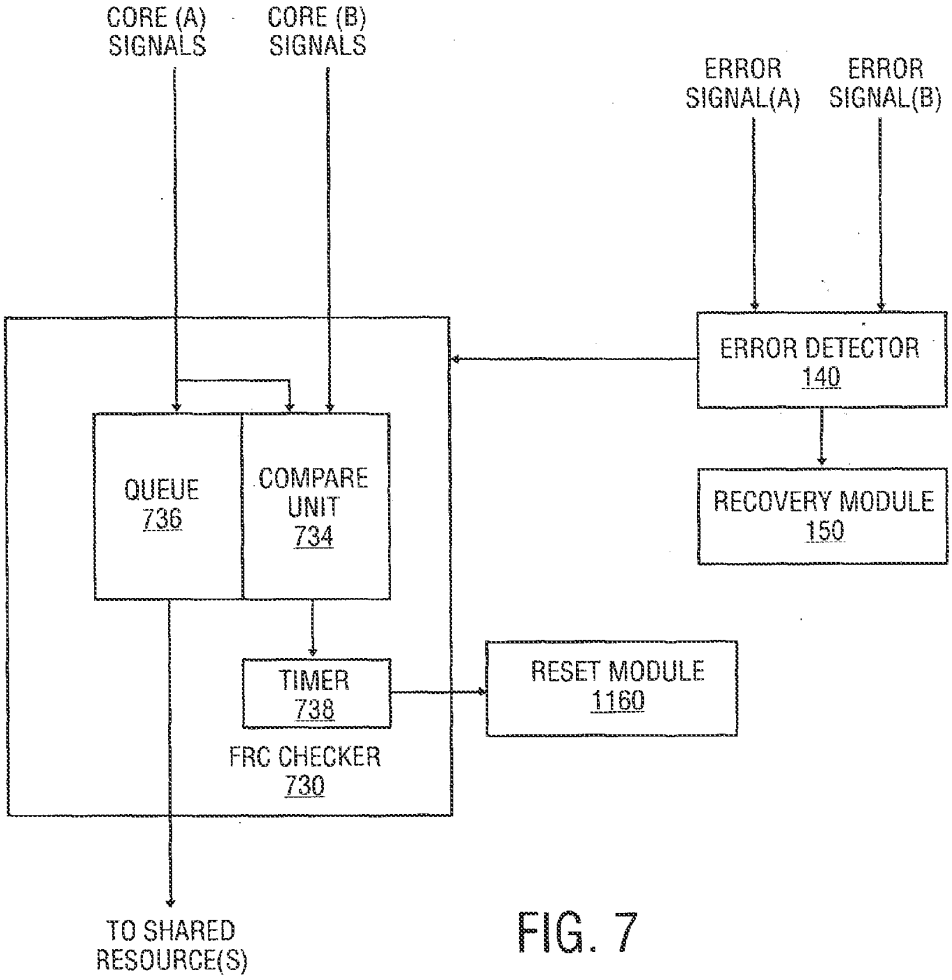

CORE (A) SIGNALS
CORE (B) SIGNALS
ERROR SIGNAL(A)
ERROR SIGNAL(B)
ERROR DETECTOR 140
QUEUE 736
COMPARE UNIT 734
RECOVERY MODULE 150
TIMER 738
RESET MODULE 1160
FRC CHECKER 730
TO SHARED RESOURCE(S)

FIG. 7

ON-DIE MECHANISM FOR HIGH-RELIABILITY PROCESSOR

BACKGROUND OF THE INVENTION

[0001] 1. Technical Field

[0002] The present invention relates to microprocessors and, in particular, to mechanisms for handling errors in FRC-enabled processors.

[0003] 2. Background Art

[0004] Servers and other high-end computing and communication systems are designed to provide high levels of reliability and availability. Soft errors pose a major challenge to both of these properties. Soft errors result from collisions between high-energy particles, e.g. alpha particles, and charge storing nodes. They are prevalent in storage arrays, such as caches, TLBs, and the like, which include large numbers of charge storing nodes. They also occur in random state elements and logic. Rates of occurrence of soft errors (soft error rates or SERs) will likely increase as device geometry decreases and device densities increase.

[0005] Highly reliable systems include safeguards to detect and manage soft errors, before they lead to silent, e.g. undetected, data corruption (SDC). However, to the extent error detection/handling mechanisms that support high-reliability operations take a system away from its normal operations, the system's availability is reduced. For example, one such mechanism resets the system to its last known valid state if an error is detected. The system is unavailable to carry out its assigned task while it is engaged in the reset operation.

[0006] One well-known mechanism for detecting soft errors is functional redundancy checking (FRC). A single processor enabled for FRC may include replicated instruction execution cores on which the same instruction code is run. Depending on the particular embodiment, 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.). FRC-hardware compares results generated by each core, and if a discrepancy is detected, the FRC system passes control to an error-handling routine. The point(s) at which results from different execution cores are compared represents the FRC-boundary for the system. Errors that are not detected at the FRC boundary can lead to SDC.

[0007] Since FRC errors indicate only that the execution cores disagree on a result, FRC errors are detectable but not recoverable. As noted above, the FRC error handling routine typically resets the system to the last known point of reliable data. This reset mechanism is relatively time consuming. It takes the system away from its normal operations, reducing system availability.

[0008] FRC is only one mechanism for handling soft errors, and for random logic and random state elements, it is the primary mechanism. Array structures present a different picture. Array structures typically include parity and/or ECC hardware, which detect soft errors by examining properties of the data. In many cases, the system can correct errors created by data corruption using relatively fast hardware or software mechanisms. 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. Handling these otherwise recoverable errors through a reset mechanism reduces system availability.

[0009] The present invention addresses mechanisms for combining recoverable and non-recoverable error handling mechanisms efficiently in FRC-enabled processors.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The present invention may be understood with reference to the following drawings, in which like elements are indicated by like numbers. These drawings are provided to illustrate selected embodiments of the present invention and are not intended to limit the scope of the invention.

[0011] **FIG. 1** is a block diagram of a processor including dual execution cores and FRC-detection and handling logic.

[0012] **FIG. 2** is a block diagram of an embodiment of the processor of **FIG. 1** that is capable of operating in multiple modes.

[0013] **FIG. 3A** is a block diagram of one embodiment of a computing system that implements the multi mode processor of **FIG. 2**.

[0014] **FIG. 3B** is a block diagram of a mechanism for signaling recoverable errors in the computing system of **FIG. 3A**.

[0015] **FIG. 4** is a block diagram representing the data paths of the computing system of **FIG. 3A**.

[0016] **FIG. 5** is a flowchart representing one embodiment of a mechanism for recovering from soft error in an execution core.

[0017] **FIG. 6** is a flowchart representing one embodiment of a mechanism for recovering from a soft error in a multi execution core processor.

[0018] **FIG. 7** is a block diagram representing one embodiment of an FRC checker that mitigates race conditions between recoverable and non-recoverable error mechanisms.

DETAILED DESCRIPTION OF THE INVENTION

[0019] The following discussion sets forth numerous specific details to provide a thorough understanding of the invention. However, those of ordinary skill in the art, having the benefit of this disclosure, will appreciate that the invention may be practiced without these specific details. In addition, various well-known methods, procedures, components, and circuits have not been described in detail in order to focus attention on the features of the present invention. For example, aspects of the present invention are illustrated using a dual-core processor, but persons skilled in the art will recognize that more than two cores may be used with appropriate modifications of the reset and recovery mechanisms.

[0020] **FIG. 1** is a block diagram representing one embodiment of an FRC-enabled processor **110** in accordance with the present invention. Processor **110** includes first and second execution cores **120**(*a*), **120**(*b*) (generically, execution core **120**), an FRC checker **130**, an error detector

140, a recovery module **150**, a reset module **160**, and shared resources **170**. A portion of the FRC-boundary is indicated by dashed line **104**. For purposes of illustration, recovery module **150** and reset module **160** are shown as part of processor **110**. These modules may be implemented in whole or in part as hardware, firmware or software and located on or off of the processor die. Similarly, shared resources **170** may include components that are on the processor die as well as components that are on one or more different die.

[**0021**] 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. Data pipeline **124** represents the logic that operates on various types of data as it moves through processor **110** toward FRC checker **130**. 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. Error pipeline **128** represents the logic that operates on various types of data to detect errors in the data and provide appropriate signals to error detector **140**. For example, the signals may be one or more bits (flags) representing the parity or ECC status of data retrieved from various storage arrays (not shown) of processor **110**. Soft errors in these arrays may appear as parity or ECC error flags when the corrupted data is accessed.

[**0022**] If an error reaches error detector **140** from either core **120**, recovery module **150** is activated to implement a recovery routine. Recovery can be implemented with relatively low latency by hardware, software, firmware or some combination of these. For example, there is an extremely small probability that data is corrupted in both execution cores **120** at the same (or nearly the same) time. This leaves an uncorrupted copy of the data available to restore data integrity to processor **110**. However, an FRC error will be triggered if the corrupted data from one execution core and an uncorrupted version of the data from the other execution are allowed to reach FRC checker **130** before recovery module **150** is activated. Since FRC errors are not recoverable, reset module **160** resets the system, if FRC checker **130** signals an FRC error before the underlying parity/ECC error is detected.

[**0023**] Not all FRC-errors are traceable to underlying parity/ECC or other correctible soft errors. For those that are, it is faster for error detector **140** to address the underlying soft error than for FRC-checker to address the FRC error that results when the corrupted data reaches FRC boundary **104**. As noted above, the reset process has a significantly longer latency than the recovery process, and it is to be avoided if the error can be corrected by recovery module **150**. In addition, reset usually brings the entire system down, whereas recovery only results in a temporary performance loss. For this reason, FRC checker **130** is temporarily disabled if error detector **140** detects an error in either error pipeline **128**, since execution cores **120** are no longer in lock step.

[**0024**] Execution cores **120** operate in lock step during normal FRC mode, but data pipeline **124** and error pipeline **128** may operate relatively independently. For example, ECC hardware is relatively complex and, consequently, relatively slow, especially for 2-bit errors. A flag signaling such an error may reach error detector **140** before, after, at the same time as the data with which it is associated reaches FRC checker **130**. This flexibility is generally beneficial. For example, it allows data to be used speculatively before its error status is determined. Since soft errors are relatively rare, and error pipeline **128** is generally as fast as data pipeline **124**, this flexibility is net positive. As long as the error flag arrives at error detector **140** in time to disable FRC checker **130** before it acts on a mismatch attributable to the corrupted data, the relatively low latency recovery routine is engaged.

[**0025**] As discussed below, processor **110** may implement strategies to mitigate the race between recoverable and non-recoverable error mechanisms. For example, a streamlined signaling mechanism may be used in FRC mode to speed disabling of FRC checker **130** in the event of a non-FRC error. In addition, FRC errors may be delayed for an interval prior to reset, in case a late arriving recoverable error signal obviates the need for reset.

[**0026**] For one embodiment of the invention, processor **110** is capable of operating in a high reliability (e.g. FRC) mode or a high performance (e.g. multi-core) mode. The operating mode may be selected, for example, when a computing system that includes processor **110** is booted or reset. In FRC mode, execution cores **120**(*a*) and **120**(*b*) may appear to the operating system as a single logical processor. Execution cores **120**(*a*) and **120**(*b*) process the same code sequence, and the results generated are compared by FRC checker **130**. If the results agree, a machine state corresponding to the code sequence is updated.

[**0027**] In FRC mode, one of execution cores **120** may be designated as the master. The master refers to the execution core that is responsible for updating the resources shared by execution cores **120**. The other execution core **120** may be designated as the slave. The slave is responsible for generating results from the same code sequence to be checked against those of the master. Since an error may occur in either the master or slave, embodiments of the present invention allow the master/slave designation to be changed dynamically. As discussed below, this allows the slave to take over the master designation to implement recovery if a recoverable error is detected in the execution core that is currently designated as master.

[**0028**] In multi-core mode, execution cores **120**(*a*) and **120**(*b*) may appear to the operating system as two different logical processors on a single processor die. In this mode, execution cores **120**(*a*) and **120**(*b*) process different code sequences, and each updates a machine state associated with the code sequence it processes. Portions of the machine state of a logical processor may be stored in a cache and/or register associated with the corresponding execution core. At some point(s) on the processor die, results from execution cores **120**(*a*) and **120**(*b*) are routed to shared resource(s) **170** for, e.g., storage (cache) or transmission off of the processor die (bus). For this embodiment, additional logic is provided to mediate access to shared resources **170** by execution cores **120**(*a*) and **120**(*b*). In general, multi-core mode allows the execution cores of the processor to be controlled separately.

[**0029**] **FIG. 2** is a block diagram representing an embodiment of processor **110** that is capable of operating in multiple modes, e.g. FRC mode and multi-core mode. For the disclosed embodiment, an arbitration unit **180** is provided to manage transactions to shared resources **170** by execution cores **120**(*a*) and **120**(*b*), when processor **110**

operates in multi-core mode. Arbitration unit **180** is associated with FRC unit **130**, placing the arbitration point for multi-core mode operation proximate to the FRC boundary for FRC mode operation. In multi-core mode, signals from execution cores **120**, e.g. transaction request signals, may be processed by arbitration unit **180**, which manages access to shared resource(s) **170**. In FRC mode, signals from execution cores **120** may be processed by FRC checker **130**, which compares them to detect soft errors in either execution core. Locating FRC checker **130** and arbitration unit **180** in close proximity expands the FRC boundary to encompass most, if not all, of the logic for which signals from the two execution cores remain distinct. It also reduces the wiring necessary to support processor **110** in FRC and multi-core modes.

[0030] Expansion of the FRC boundary in this manner naturally increases the time necessary to propagate signals to FRC checker **130**. This increased "flight time" provides more time for a parity or ECC error to reach detector **140**, which increases the opportunity for error recovery. As noted above, the recovery routine triggered by error detector **140** provides greater system availability than the reset routine triggered by FRC checker **130**. Expanding the FRC boundary thus increases both the amount of logic that is duplicated for execution cores **120** and the flight time, during which detectable errors may be identified. The former increases FRC protection, albeit through a reset mechanism. The latter increases the likelihood that errors identifiable through parity, ECC or similar core-specific protections are handled through recovery rather than reset.

[0031] **FIG. 3A** is a block diagram representing one embodiment of a computing system **300** in accordance with the present invention. The disclosed embodiment of system **300** includes a processor **310**, chipset **370**, main memory **380**, non-volatile memory **390** and peripheral device(s) **398**. For the disclosed embodiment of system **300**, processor **310** may be operated in FRC mode or in multi-core mode. The mode may be selected, e.g., when computing system **300** is booted or reset. Chipset **370** manages communications among processor **310**, main memory **380**, non-volatile memory **390** and peripheral devices **398**.

[0032] Processor **310** includes first and second execution cores **320**(*a*) and **320**(*b*), respectively (generically, execution cores **320**). Each execution core includes execution resources **324** and a bus cluster **328**. Execution resources **324** may include, for example, one or more integer, floating point, load/store, and branch execution units, as well as register files and cache(s) to supply them with data (e.g. instructions, operands, addresses). Bus cluster **328** represents logic for managing transactions to a cache **340** that is shared by execution cores **320**(*a*) and **320**(*b*), as well as to a front side bus **360**, for those transactions that may miss in shared cache **340**. Resources corresponding to the error pipeline of **FIGS. 1 and 2** may be associated with execution resources **324** and/or bus cluster **328**.

[0033] Interface units (IFU) **330**(*a*), **330**(*b*) (generically, IFU **330**) represent a boundary between execution cores **320** and shared resources, cache **340** and FSB **360**. The disclosed embodiment of IFU **330** includes an FRC unit **332** and an arbitration unit **334**. As noted above, FRC unit **332** and arbitration unit **334** receive signals from execution cores **320**, and locating them proximate to each other results in significant savings of wiring on the processor die. Also shown in **FIG. 3A** are error units **336**(*a*) and **336**(*b*), which include components to monitor for detectable errors in execution cores **320**(*a*) and **320**(*b*).

[0034] For FRC mode, FRC unit **332** compares signals from execution cores **320** for transactions to shared resources like cache **340** and FSB **360**. FRC unit **332** thus forms part of the FRC boundary of processor **310**. For multi-core mode, arbitration unit **334** monitors signals from execution cores **320** and grants access to its associated shared resource according to an arbitration algorithm. The arbitration algorithm implemented by arbitration unit **334** may be, for example, a round robin scheme, a priority-based scheme or similar arbitration algorithms. For both FRC and multi-core mode, error unit **336** may monitor signals from execution cores **320** for recoverable errors.

[0035] Portions of recovery module **150** and reset module **160** (**FIG. 2**) may be located on processor **310** or elsewhere in system **300**. For one embodiment, a recovery routine **392** and a reset routine **394** may be stored in non-volatile memory **390** and images of these routines may be loaded in main memory **380** for execution. For this embodiment, recovery module **150** and reset module **160** may include pointers to recovery routine **392** and reset-routine **394**, respectively (or their images in main memory **380**).

[0036] The disclosed embodiment of system **300** also includes interrupt controllers **370**(*a*) and **370**(*b*) (generically, interrupt controller(s) **370**) to process interrupts for execution cores **320**(*a*) and **320**(*b*), respectively. Each interrupt controller **370** is shown having first and second components **374** and **378**, respectively, to accommodate the different clock domains in which interrupt controller **370** may operate. For example, FSB **360** typically operates at a different frequency than processor **310**. Consequently, components of processor **310** that interact directly with FSB **360** typically operate in its clock domain, which is indicated as area **364** on processor **310**.

[0037] The disclosed embodiment of interrupt controller **370** also includes an FRC-boundary-like component in the form of XOR **372**. XOR **372** signals an FRC error if it detects a mismatch between outgoing signals, e.g. interrupt responses, of components **374**(*a*) and **374**(*b*) from execution cores **320**(*a*) and **320**(*b*). Errors attributable to interrupt controllers **370** may still arise, however, from soft errors in components **378**(*a*), **378**(*b*) in FSB clock domain **364**. These error may be detected by the discrepancies they introduce between the subsequent operations of execution cores **320**(*a*) and **320**(*b*).

[0038] For the disclosed embodiment of system **300**, a common snoop block **362** processes snoop transactions to and from execution cores **320**(*a*) and **320**(*b*). XOR **366** provides FRC-checking on snoop responses from execution cores **320**(*a*), **320**(*b*) and signals an error if a mismatch is detected. XORs **372** and **366** may be disabled if processor **310** is operating in multi-core mode.

[0039] **FIG. 3B** is a block diagram representing one embodiment of an apparatus **344** for broadcasting-recoverable error conditions to components of computing system **300**. For example, error units **336**(*a*) and **336**(*b*) may represent ECC or parity error detection logic for various arrays (register, caches, buffers, etc) of execution cores **320**(*a*) and **320**(*b*), respectively, and/or exception logic to handle these

errors. An OR gate **338** monitors error signals from execution cores **320** and asserts a signal to disable FRC unit **332** if either error signal is asserted. The error signal may be a high level interrupt, such as the machine check abort (MCA) defined for Itanium® processors. The output of OR gate **338** is also fed back to execution cores **320** to indicate to the error-free execution core that a recovery mechanism is to be initiated. A second OR gate **339** is provided to transfer error signals to execution cores **320** from the shared resources.

[0040] If the error signal does not disable FRC unit **332** the corrupted data triggers an FRC error, and an otherwise recoverable error is treated as a non-recoverable, e.g. FRC, error. That is, the system goes through a reset operation rather than a shorter recovery operation. Depending on the particular implementation of the system, there may be a number of cases in which the race to FRC unit **332** between the error signal and the mismatched data signals from the execution cores (created by the recoverable error) is close. For this reason, apparatus **344** may include a mechanism to accelerate error signal propagation, at least in FRC mode.

[0041] For one embodiment, apparatus **334** supports a high level interrupt such as an MCA that operates in both FRC and high performance modes. In high performance mode, the error signal is subject to pipeline stalls, e.g., in the front end of the execution core or in the L2 cache. This ensures that no unnecessary MCAs are taken, since the event that triggered the stall may make the error signal moot. In FRC mode, the error signal bypasses these stalls. Bypassing stalls in FRC may result in the processing of some unnecessary error signals, but it also reduces the probability that an FRC error is triggered before the (non-FRC) error signal disables FRC unit **332**. As discussed in conjunction with **FIG. 7**, embodiments of processor **110** may also include a hardware mechanism to mitigate the race between the error signal and the core signals that reflect the corrupted data.

[0042] **FIG. 4** is a block diagram representing data paths for one embodiment of computing system **310**, including FRC components to support processor **310** in FRC mode. For the disclosed embodiment, cache **340**, FSB **360** and execution cores **320** are coupled through a series of buffers. For example, a write-out buffer (WOB) **410** stages data evicted from cache **340** to main memory **380**, and a snoop data buffer (SDB) **420** provides snoop data from execution cores **320** or cache **340** to FSB **360**, responsive to a snoop hit in these structures (Execution cores **320** may each have one or more levels of cache in addition to shared cache **340**).

[0043] A pair of write-line buffers (WLB) **430**(*a*), **430**(*b*) stage data from execution cores **320**(*a*), **320**(*b*), respectively, to cache **340** or FSB **360**, and a pair of read-line buffers **440**(*a*), **440**(*b*) stage data from FSB **360** to cache **340** or execution cores **320**. Coalescing buffers (CB) **450**(*a*), **450**(*b*) collect data to be written to memory **380** and forward it to FSB **360**, periodically. For example, multiple data writes to the same line of memory may be collected in CBs **450** before triggering a write transaction on FSB **360**.

[0044] For the disclosed embodiment, logic associated with these buffers provides the FRC-checking and data routing functions when processor **310** is operated in FRC mode. For example, logic block **454** represents MUX and XOR functions for data in CBs **450**(*a*), **450**(*b*). The XOR function provides FRC checking if processor **310** is operating in FRC mode. The MUX function provides data routing if processor is operating in multi-core mode. Logic blocks **434** and **444** provide similar functions for data in WLBs **430**(*a*), **430**(*b*) and RLBs **440**(*a*), **440**(*b*), respectively. MUXs **460**, **470** and **480** route data from different sources to cache **340**, FSB **360** and execution cores **320**.

[0045] As noted above, the recovery mechanism for errors detected within the FRC boundary may be handled by various combinations of hardware, software and firmware modules. One 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 the processor abstraction layer (PAL), which provides an abstraction of the processor to the rest of the computing system. Implementing recovery in the PAL hides the recovery process from system level code, such as the system abstraction layer (SAL), e.g. BIOS, and the operating system. PAL-based implementations of the recovery mechanism should be able to complete quickly enough to avoid triggering a time-out period enforced by the operating system. Recovery mechanisms may also be implemented using system level code, e.g. SAL/BIOS, or operating system code. The latter implementations may not be subject to the same time constraints as the PAL-based implementation. Unless otherwise noted, the recovery mechanisms discussed below may be implemented using code associated with any of the foregoing sources.

[0046] **FIG. 5** is a flowchart representing a mechanism for recovering from an error detected in one of the execution cores before it triggers an FRC reset. Responsive to a parity, ECC or other error detected in one of the execution cores, a signal is broadcast **510** to indicate the start of a recovery routine. As long as the error is detected before it triggers an FRC reset, the corrupted data can be localized to one of the execution cores, leaving the machine state data of the other execution core available for recovery. Accordingly, the machine state of the good core is saved **520**. To prepare the processor for recovery, both cores are initialized **530** to a specified condition, and the saved machine state is restored **540** to the initialized cores. FRC mode is then restored **550** and the processor returns **560** to the interrupted code.

[0047] For one embodiment of the invention, one of execution cores **120** may be designated as the master core and the other as the slave core when processor **10** is operating in FRC mode. For this embodiment, signals generated by the master and slave cores are compared at the FRC boundary to determine if reset is necessary. If no FRC reset is warranted, signals generated by the master core are transferred to shared resource(s) **170**, and signals generated by the slave core are dropped. For this embodiment, a bit in a status register of each execution core **120** may be used to indicate its status as master or slave. The bit may be set, for example, when the system is booted or reset. As discussed below in greater detail, the master/slave status of an execution may also be changed dynamically to allow recovery for an error in either core. For errors detected within the FRC boundary, e.g. recoverable errors, the actions of the master and slave core may differ, depending on which core generated the error.

[0048] **FIG. 6** is a flowchart representing one embodiment of a mechanism **600** for recovering from an error detected in an execution core designated as the slave execution core.

Operations of the slave execution core are shown on the left, and operations of the master execution core are shown on the right.

[0049] Routine **600** is initiated if the slave execution core detects **610** an error (parity, ECC, etc.). The slave core triggers **620** an interrupt to signal the error condition to other components of the computing system. For embodiments of routine **600** implemented by PAL or comparable processor-level code, broadcast of the interrupt signal may be limited to components within the processor chip, such as the master execution core. In addition to signaling the error, the slave execution core disables **630** the FRC unit and suspends its activities. Disabling the FRC unit prevents the error from triggering an FRC reset when it reaches the FRC boundary, and suspending activities in the slave core prevents it from disrupting the recovery process.

[0050] In response **624** to the interrupt, the master execution core determines **640** whether its state data contains any errors. For example, each 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. Except for the very rare case in which soft errors occur in both execution cores almost simultaneously, the master core is likely to be clean. If it is not clean **640**, there is no uncorrupted processor state with which to implement a recovery. In this case, the master core signals **642** a reset condition to the slave core and the computing system executes **644** a full, e.g. FRC-level, reset.

[0051] If the state data for the master core is not corrupted, the master core saves **660** its machine state and flushes **664** queues and buffers in its pipeline. For example, the master core may save the contents of its data and control registers and low-level cache(s) to a protected area of memory. The master core also signals **668** a limited reset to the slave core and sets **676** it resources to a specified state, e.g. initializes its pipeline. The slave core detects **670** the limited reset and initializes **674** its pipeline, synchronizing the states of the cores.

[0052] With the cores thus synchronized, FRC mode is reactivated **680**. This may be accomplished, for example, by having each core execute a handler routine, which sets appropriate state bits in its status/control registers. The saved state is restored **684** to both execution cores, and control is returned **690** to the interrupted code sequence.

[0053] Method **600** represents an embodiment of the recovery mechanism for the case in which the error is detected in the execution core that is currently designated as the slave core. For one embodiment, the slave core is the execution core that does not "control" the shared resource(s). For example, in FRC mode, signals from the slave execution core are dropped following comparison with those from the master execution core at the FRC boundary. If no FRC error is detected, signals from the master core are used to control the shared resources outside the FRC boundary.

[0054] If the 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. For example, master/slave designation may be indicated by the state of a bit in a status register associated with each execution core. The execution core for which this status bit is in the master state controls the shared resources, which are used to implement the state-saving operations of recovery routine **600**, e.g. operation **660**.

[0055] For one embodiment of the recovery routine, the execution core in which the error originates may check its master/slave status bit. If the status bit indicates it is the slave, method **600** may be implemented as described. If the status bit indicates it is the master, it may signal the slave to change its status to master, change its own status to slave, and suspend activities.

[0056] **FIG. 7** is a block diagram illustrating an embodiment of an FRC-checker **730** that mitigates race conditions between recoverable and non-recoverable error handling. The disclosed embodiment of FRC checker **730** includes a compare unit **734**, queue **736**, and timer unit **738**. Queue **736** receives data from execution core (a), and compare unit **734** compares the data from cores A and B, and sets a status flag to indicate if the comparison yields a match. If the data matches, the status flag is set to indicate the match.

[0057] If the data does not match, the status flag is set to indicate the mismatch and timer unit **738** is triggered to begin a countdown interval. If error detector **140** receives an error flag before the timeout interval expires, it disables FRC checker **730** and triggers recovery unit **150** to implement the recovery routine.

[0058] There has thus been disclosed a mechanism for handling recoverable and non-recoverable errors in multi-core processor. The multiple cores may be operated in FRC mode, in which case, one or more checker units compare signals from the cores to detect non-recoverable errors. In addition, each core includes an error unit to detect recoverable errors. If a recoverable error is detected, the checker units are disabled and a recovery routine is implemented. A multi-core mode embodiment of the multi-core processor may include an arbitration unit, proximate to the checker, to control access to the shared resource(s). Proximity of the FRC boundary to the shared resources increases the logic protected by the FRC boundary and reduces wiring necessary for the multi-core mode implementation.

[0059] Embodiments of the present invention detect virtually all errors that are undetected in non-FRC-enabled systems, and they support recovery of virtually all detectable errors, including those that are typically handled through reset in other FRC-enabled processors.

[0060] The disclosed embodiments have been provided to illustrate various features of the present invention. Persons skilled in the art of processor design, having the benefit of this disclosure, will recognize variations and modifications of the disclosed embodiments, which none the less fall within the spirit and scope of the appended claims.

We claim:

1. A processor comprising:

first and second execution cores to operate in an FRC mode;

a resource to process a transaction from at least one of the first and second execution cores; and

an interface control unit to regulate access to the resource by the first and second execution cores, the interface control unit including an FRC check unit to compare

transaction signals from the first and second execution cores and to signal an error if the comparison indicates a mismatch.

2. The processor of claim 1, further comprising an error detector to detect errors in the first and second execution cores and to disable the FRC checker responsive to detecting an error.

3. The processor of claim 2, wherein the error detector comprises first and second error detectors to detect errors in the first and second execution cores, respectively.

4. The processor of claim 3, wherein the first error detector triggers an error signal, responsive to an error in the first execution core, to disable the FRC check unit and to initiate a recovery procedure using the second execution core.

5. The processor of claim 4, wherein the second execution core is designated as an FRC slave and is re-designated as an FRC master, responsive to the error signal.

6. The processor of claim 5, wherein the second execution core saves its machine state data to a memory location, and executes a reset sequence.

7. The processor of claim 2, wherein the first and second execution cores may also operate in a multi-core mode and the interface control unit further comprises an arbitration unit to regulate access to the shared resource by the execution cores if they operate in multi-core mode.

8. The processor of claim 7, wherein the shared resource comprises a cache that may process transactions from both the first and second cores in multi-core mode and that may process transactions from only one of the first and second core in FRC mode.

9. The processor of claim 7, wherein, responsive to detecting an error the error detector triggers an interrupt if the processor is in multi-core mode and an accelerated interrupt if the processor is in FRC mode.

10. The processor of claim 9, wherein the accelerated interrupt bypasses portions of an execution core traversed by the interrupt in multi-core mode.

11. A system comprising:

a first memory location to store a recovery routine;

a second memory location to store a reset routine;

first and second execution cores capable of operating in an FRC mode;

an error unit to initiate the recovery routine responsive to detecting an error in one of the first and second execution cores; and

an FRC checker to initiate the reset routine responsive to detecting a mismatch between signals from the first and second execution cores.

12. The system of claim 11, wherein the error unit disables the FRC checker, responsive to detecting the error in one of the first and second execution cores.

13. The system of claim 12, wherein the reset routine includes instructions executable by the first and second execution cores to initialize the first and second execution cores in either a multi-core mode or the FRC mode.

14. The system of claim 13, further comprising a cache to be shared by the first and second execution cores if the first and second execution cores are initialized in multi-core mode.

15. The system of claim 14, further comprising an arbitration unit to manage access to the cache by the first and second execution cores in multi-core mode.

16. The system of claim 15, wherein the FRC checker monitors transaction signals to the arbitration unit from the first and second execution cores in FRC mode and initiates the reset routine responsive to a mismatch in the transaction signals.

17. The system of claim 11, wherein the first and second execution cores operate as master and slave, respectively, in an FRC mode.

18. The system of claim 17, wherein the first execution core is disabled and the second execution core operates as the master, responsive to an error in the first execution core.

19. The system of claim 11, wherein the first and second execution cores may be initialized in a multi-core mode or an FRC mode.

20. The system of claim 19, wherein the error unit triggers an interrupt to the first and second execution cores, responsive to an error in one of the execution cores.

21. The system of claim 20, wherein the interrupt is an accelerated interrupt if the execution cores are in FRC mode.

22. The system of claim 21, wherein the accelerated interrupt bypasses a portion of the execution core.

23. A method comprising:

operating first and second execution cores in an FRC mode;

monitoring data of the first and second execution cores for errors;

comparing signals generated by the first and second execution cores;

executing a recovery routine, responsive to an error in the first or second execution core; and

executing a reset routine, responsive to a mismatch between signals generated by the first and second execution cores.

24. The method of claim 23, further comprising suspending signal comparison responsive to the error in the first or second execution core.

25. The method of claim 24, wherein executing a reset routine responsive to the mismatch further comprises:

triggering a delay interval responsive to the mismatch;

executing the reset routine if no error is monitored in the execution cores before the delay interval expires.

26. The method of claim 25, further comprising executing the recovery routine if an error is detected in one of the execution cores before the delay interval expires.

27. The method of claim 23, wherein operating the first and second execution cores in an FRC mode comprises operating the first and second cores in the FRC mode or in a multi-core mode, responsive to a reset signal.

28. The method of claim 27, wherein executing the recovery routine comprises:

executing the recovery routine responsive to an error signaled by an interrupt if the cores are operating in multi-core mode; and

executing the recovery routine responsive to an error signaled by an accelerated interrupt if the cores are operating in FRC mode.

29. The method of claim 23, wherein operating the first and second execution cores in an FRC mode further comprises designating the first and second execution cores as master and slave execution cores, respectively.

30. The method of claim 23, wherein executing the recovery routine further comprises disabling the first execution core and designating the second execution core as master, responsive to an error in the first execution core.

* * * * *

PTO/SB/57 (01-18)
Approved for use through 11/30/2024. OMB 0651-0064
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.

(Also referred to as FORM PTO-1465)

REQUEST FOR *EX PARTE* REEXAMINATION TRANSMITTAL FORM

Address to:
Mail Stop *Ex Parte* Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attorney Docket No.: 185945.012400

Date: September 1, 2023

1. ☒ This is a request for *ex parte* reexamination pursuant to 37 CFR 1.510 of patent number 7,627,781 issued December 1, 2009. The request is made by:
   ☐ patent owner. ☒ third party requester.

2. ☒ The name and address of the person requesting reexamination is:
   Unified Patents, LLC
   4445 Willard Ave., Suite 600
   Chevy Chase, MD 20815

3. Requester ☐ asserts small entity status (37 CFR 1.27) or ☐ certifies micro entity status (37 CFR 1.29). Only a patent owner requester can certify micro entity status. Form PTO/SB/15A or B must be attached to certify micro entity status.

4. This request is accompanied by payment of the reexamination fee as set forth in:
   ☒ 37 CFR 1.20(c)(2); **or**
   ☐ 37 CFR 1.20(c)(1). **In checking this box for payment of the fee set forth in 37 CFR 1.20(c)(1), requester asserts that this request has forty (40) or fewer pages and complies with all other requirements of 37 CFR 1.20(c)(1).**

   Payment of the reexamination fee is made by the method set forth below.
   a. ☐ A check in the amount of $_____ is enclosed to cover the reexamination fee;
   b. ☒ The Director is hereby authorized to charge the reexamination fee to Deposit Account No. 50-6990;
   c. ☐ Payment by credit card. Form PTO-2038 is attached; **or**
   d. ☐ Payment made via EFS-Web.

   ☒ In addition, the Director is hereby authorized to charge any fee deficiencies to Deposit Account No. 50-6990.

5. ☒ Any refund should be made by ☐ check or ☐ credit to Deposit Account No. 50-6990. 37 CFR 1.26(c). If payment is made by credit card, refund must be to credit card account.

6. ☒ A copy of the patent to be reexamined having a double column format on one side of a separate paper is enclosed. 37 CFR 1.510(b)(4).

7. ☐ CD-ROM or CD-R in duplicate, Computer Program (Appendix) or large table
   ☐ Landscape Table on CD

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 0651-0064. Public burden for this form is estimated to average 18 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. **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.

8. ☐ Nucleotide and/or Amino Acid Sequence Submission
If applicable, items a. – c. are required.
   a. ☐ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. ☐ CD-ROM (2 copies) or CD-R (2 copies) **or**
      ii. ☐ paper
   c. ☐ Statements verifying identity of above copies.

9. ☐ A copy of any disclaimer, certificate of correction or reexamination certificate issued in the patent is included.

10. ■ Reexamination of claim(s) 1-30 is requested.

11. ■ A copy of every patent or printed publication relied upon is submitted herewith including a listing thereof on Form PTO/SB/08, PTO-1449, or equivalent.

12. ■ An English language translation of all necessary and pertinent non-English language patents and/or printed publications is attached.

13. ■ The attached detailed request includes at least the following items:
   a. A statement identifying each substantial new question of patentability based on prior patents and printed publications. 37 CFR 1.510(b)(1).
   b. An identification of every claim for which reexamination is requested, and a detailed explanation of the pertinency and manner of applying the cited art to every claim for which reexamination is requested. 37 CFR1.510(b)(2).

14. ☐ A proposed amendment is included (only where the patent owner is the requester). 37 CFR 1.510(e).

15. ■ It is certified that the statutory estoppel provisions of 35 U.S.C. 315(e)(1) or 35 U.S.C. 325(e)(1) do not prohibit requester from filing this *ex parte* reexamination request. 37 CFR 1.510(b)(6).

16. Service
   a. ■ It is certified that a copy of this request (if filed by other than the patent owner) has been served in its entirety on the patent owner as provided in 37 CFR 1.33(c).

      The name and address of the party served are:

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

      Date of Service: 9/1/2023

   OR

   b. ☐ A duplicate copy is enclosed since service on patent owner was not possible. An explanation of the efforts made to serve patent owner **is attached**. See MPEP 2220.

17. Correspondence Address: Direct all communication about the reexamination to:

☒ The address associated with Customer Number: 67395

OR

☐ Firm or Individual Name _____
(at the address identified below)

Address

City	State	Zip

Country

Telephone	Email

18. ☒ The patent is currently the subject of the following concurrent proceeding(s):

a. ☐ Copending reissue Application No. _____

b. ☐ Copending reexamination Control No. _____

c. ☐ Copending Interference No. _____

d. ☒ Copending litigation styled:

Case No. 1:23-cv-00640 (W.D. Tex., June 6, 2023);

Case No. 2:23-cv-00321 (E.D. Tex., July 12, 2023); Case No. 2:23-cv-00150 (E.D. Tex., April 5, 2023).

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.

/Jon Bills/ — Authorized Signature

09/01/2023 — Date

Jon Bills — Typed/Printed Name

71895 — Registration No.

☐ For Patent Owner Requester ☒ For Third Party Requester

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 9) 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.

Additional Uses

Additional USPTO uses of the information in this record may include disclosure to: 1) the International Bureau of the World Intellectual Property Organization, if the record is related to an international application filed under the Patent Cooperation Treaty; 2) the public i) after publication of the application pursuant to 35 U.S.C. 122(b), ii) after issuance of a patent pursuant to 35 U.S.C. 151, iii) 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, or iv) without publication of the application or patent under the specific circumstances provided for by 37 CFR 1.14(a)(1)(v)-(vii); and/or 3) the National Archives and Records Administration, for inspection of records.

- determining step is performed by system firmware [col. 5, lines 17-18].

As per claims 11-12:

Quach further teaches:

- 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 [col. 4, lines 12-24].

- 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 [fig. 5, col. 11, lines 1-18].

As per claims 13-15:

Quach further teaches:

- changing a system identifier of the spare processor to that of the boot processor [col. 13, lines 24-26];

changing a system identifier 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 [col. 13, lines 16-26; col. 14, lines 50-57].

As per claims 16-23:

Claims 16-23 are similar to claims 1-9 except using system firmware for determining LOL {Quach, col.5, lines 17-18} and copying the state of the determined one of said lockstep pair of processors that is not the cause of LOL to a spare processor {Quach, col.4, lines 12-28, fig.5, col. 11, lines 1-18}. Therefore, these claims are also rejected under the same rationale applied against claims 1-9. In addition, all of the limitations have been noted in the rejection as per claims 1-9.

As per claim 29:

Due to the similarity of claim 29 to claim 1 except for ***a system*** comprising MEANS for detecting loss of lockstep, MEANS for the determining whether the processor for which LOL is detected, MEANS for switching the role of boot processor, etc… instead of ***a method*** comprising detecting loss of lockstep, determining whether the processor for which LOL is detected, switching the role of boot processor, etc…; therefore, this claim is also rejected under the same rationale applied against claim 1. In addition, all of the limitations have been noted in the rejection as per claim 1.

As per claim 30:

Quach explicitly teaches the invention. Quach teaches:

- detecting loss of lockstep (LOL) for a processor in a multi-processor system [abstract, col. 3, lines 35-36 and col. 5, lines 20-28].
- determining that the processor for which said LOL is detected is assigned a role of boot processor [fig. 1, col. 4, lines 38-48].

Safford does not explicitly address:

- information stored to non-volatile data storage and an Itanium Processor Family (IPF) processor.

However, Quach does disclose capability of:

- a computer system includes processor having dual execution cores [abstract] comprising:
- *information stored in the non-volatile memory* [col. 3, lines 52-54] and redundant mode and split mode *in supporting the recovery from loss of lockstep in a multi-processors system*[col. 3, lines 35-37].

In addition, Fox explicitly teaches:

- system response to processor failure in a multi-processor data processing system [abstract];

- information stored in the memory and active/spare pair used in supporting the recovery from loss of lockstep in a multi-processors system[col.8, lines 60-68].

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of Applicant's invention to realizing Fox's information stored in the memory and active/spare pair used in supporting the recovery from loss of lockstep in a multi-processors system as being information stored to non-volatile data storage and an Itanium Processor Family (IPF) processor as claimed by Applicant. This is because Quach's mutli-data processors /fault tolerant system explicitly performed data failure detection and recovery (i.e., *loss of lockstep* and/or *fail-over*) within plurality of processors via *state information*, *parity information*, and *boot information* processes.

This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to improve the multi-data processors, failover, and/or fault-tolerant system operation availability and network/system performance.

Allowable Subject Matter

Claims 24-28 are allowed over the prior art of record.

However, These claims are still objected under the double patenting rejection stated in the paragraph above.

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dieu-Minh Le whose telephone number is (571) 272-3660. The examiner can normally be reached on Monday - Thursday from 8:30 AM to 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Scott Baderman can be reached on (571)272-3644.
The Tech Center 2100 phone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/DIEU-MINH T. LE/
DIEU-MINH THAI LE
PRIMARY EXAMINER
ART UNIT 2114

DML
02/28/08

Notice of References Cited	Application/Control No. 10/972,588	Applicant(s)/Patent Under Reexamination MICHAELIS ET AL.	
	Examiner Dieu-Minh Le	Art Unit 2114	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,625,749	09-2003	Quach, Nhon	714/10
*	B	US-7,251,746	07-2007	Fox et al.	714/13
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

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								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								
	21	✓								
	22	✓								
	23	✓								
	24	=								
	25	=								
	26	=								
	27	=								
	28	=								
	29	✓								
	30	✓								

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	10-13; 712/31	2/21/08	dml

SEARCH NOTES		
Search Notes	Date	Examiner
EAST search all files updated	2/21/08	dml

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	25072	(loss adj (lockstep lock-step)) or LOL	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:49
L2	1637	(boot$3 initializ $3) adj processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:49
L3	59297	(plural$3 multipl $3 multi-) adj (processor CPU)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:49
L4	29535	(master slave primary second $3 pair spare redundan$2 back-up "back up") adj (processor CPU)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:49
L5	11	L1 same L2 same L3 same L4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:49
L6	96	(712/31).ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:49
L7	2051	(714/10-13). ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:49
L8	2	L5 and (L6 L7)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:49

L9	31	1 and 3 and 4 and (nonvolatile non-volatile RAM NVRAM)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:52
L10	14	9 and firmware	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:52
L11	26	(michaelis-scott$ myer-sylvia$ mchardy-william $).in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:53
L12	26	L11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:53
L13	8	("714"/$).ccls. and 1 and 2 and 4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:55
L14	0	("712"/$).ccls. and 1 and 2 and 4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 17:55
L15	37	1 and 4 and (nonvolatile non-volatile RAM NVRAM flash remapp$3 map $3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 18:00
L16	10	15 and 2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/02/21 18:00

2/21/08 6:02:35 PM
C:\Documents and Settings\MLe\My Documents\EAST\Workspaces\10972588a.wsp

RECEIVED
CENTRAL FAX CENTER
DEC 12 2007

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

RESPONSE TO NON-FINAL OFFICE ACTION MAILED 09/12/2007

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is in response to the Office action mailed September 12, 2007, in the above-captioned application.

Amendments to the specification are reflected in substitute paragraphs beginning on page 2 of this paper.

Remarks and arguments begin on page 8 of this paper.

RECEIVED
CENTRAL FAX CENTER
DEC 12 2007

AMENDMENTS TO THE SPECIFICATION

[0001] The present application is related to the following concurrently filed and commonly assigned U.S. Patent Applications: ~~Serial No. _____ [Attorney Docket No. 200404944-1]~~ U.S. Patent Application Serial No. 10/973,077 (U. S. Published Patent Application 20060107112) titled "SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR"; ~~Serial No. _____ [Attorney Docket No. 200404943-1]~~ U.S. Patent Application Serial No. 10/973,004 (U. S. Published Patent Application 20060107117, now U.S. Patent Number 7,308,566) titled "SYSTEM AND METHOD FOR CONFIGURING LOCKSTEP MODE OF A PROCESSOR MODULE"; ~~Serial No. _____ [Attorney Docket No. 200404946-1]~~ U.S. Patent Application Serial 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"; ~~Serial No. _____ [Attorney Docket No. 200404973-1~~ U.S. Patent Application Serial 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"; ~~Serial No. _____ [Attorney Docket No. 200404971-1]~~ U.S. Patent Application Serial 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"; ~~Serial No. _____ [Attorney Docket No. 200404970-1]~~ U.S. Patent Application Serial 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"; ~~Serial No. _____ [Attorney Docket No. 200404972-1]~~ U.S. Patent Application Serial 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 ~~Serial No. _____ [Attorney Docket No.~~

PAGE 4/14 * RCVD AT 12/12/2007 5:22:43 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/14 * DNIS:2738300 * CSID:303 297 2266 * DURATION (mm-ss):04-18

~~200404974-1~~] U.S. Patent Application Serial 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.

[0019] As described further herein and in concurrently filed and commonly assigned U.S. Patent Application Serial No. _____ ~~[Attorney Docket No. 200404941-1]~~ 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. Exemplary techniques are described in concurrently filed and commonly assigned U.S. Patent Application Serial No. _____ ~~[Attorney Docket No. 200404944-1]~~ U.S. Patent Application Serial No. 10/973,077 (U. S. Published Patent Application 20060107112) titled "SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR," the disclosure of which is incorporated herein by reference, for establishing a spare processor for the boot processor. Such exemplary techniques include techniques for establishing a hot spare processor that is held in reserve until such time as it is used to assume the role of boot processor responsive to a detected LOL in the boot processor. The exemplary techniques further include techniques for establishing an active spare processor that is not held in reserve, but is instead made available to the OS for executing instructions during normal system operation and is dynamically transformed into a hot spare (i.e., it is idled) such that it can assume the role of the boot processor responsive to detection of LOL in the boot processor.

[0021] As described further in concurrently filed and commonly assigned U.S. Patent Application Serial No. ~~[Attorney Docket No. 200404944-1]~~ U.S. Patent Application Serial No. 10/973,077 (U. S. Published Patent Application 20060107112) titled "SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR," in certain embodiments, information (referred to herein as a "device tree") is stored to a 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).

[0023] Concurrently filed and commonly assigned U.S. Patent Application Serial No. ~~[Attorney Docket No. 200404941-1]~~ 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 recognize the lockstep pair of processors as again being available for servicing instructions.

[0024] 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 switching the role of boot processor to an established spare processor responsive to detection of LOL 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 Serial No. ~~[Attorney Docket No. 200404941-1]~~ 10/973,076 (U. S. Published Patent Application 20060107107) titled "SYSTEM AND METHOD FOR PROVIDING FIRMWARE RECOVERABLE LOCKSTEP PROTECTION," for example.

[0041] Thus, operational block 109 essentially makes the spare processor pair 15 the system's boot processor, and then in certain embodiments firmware 18 resets the lockstep processor pair 12 and reestablishes its lockstep in block 110. 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 FIGURE 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 Serial No. ~~[Attorney Docket No. 200404974-1]~~ U.S. Patent Application Serial 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

PAGE 7/14 * RCVD AT 12/12/2007 5:22:43 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/14 * DNIS:2738300 * CSID:303 297 2266 * DURATION (mm-ss):04-18

FOR ANOTHER PROCESSOR", the disclosure of which is hereby incorporated herein by reference.

[0059] Turning to FIGURE 4, an exemplary operational flow diagram for switching the role of boot processor to a hot spare processor according to one embodiment is shown. In this case, an LOL has been detected for the boot processor. In operational block 401, the system firmware allocates space to save the status of the good processor of the lockstep pair of processors for which LOL was detected. Exemplary techniques for determining which of the lockstep pair of processors had the error are disclosed in concurrently filed and commonly assigned U.S. Patent Application Serial No. ~~[Attorney Docket No. 200404946-1]~~ U.S. Patent Application Serial 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," the disclosure of which is hereby incorporated herein by reference. Accordingly, an appropriate amount of data storage space for storing the cache of the "good" processor in the lockstep pair of processors is allocated in block 401. For instance, an amount of SRAM may be allocated for storing the registers from the good processor.

[0066] In operational block 504, the system firmware allocates space to save the status of the good processor of the lockstep pair of processors for which LOL was detected (i.e., the lockstep pair of processors of the CBP). Exemplary techniques for determining which of the lockstep pair of processors had the error are disclosed in concurrently filed and commonly assigned U.S. Patent Application Serial No. ~~[Attorney Docket No. 200404946-1]~~ U.S. Patent Application Serial 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," the disclosure of which is hereby incorporated herein by reference. Accordingly, an appropriate amount of data storage space for storing the cache of the "good" processor in the lockstep pair of processors is