



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
11/102,127	7376798	2187	9200



Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 45590 on 08/27/2008

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

The address of record for Customer Number 45590 is:

45590
TRANSMETA C/O MURABITO, HAO & BARNES LLP
TWO NORTH MARKET STREET
THIRD FLOOR
SAN JOSE, CA 95113

PART 1 - ATTORNEY/APPLICANT COPY

page 1 of 1



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.
11/102,127	05/20/2008	7376798	TRAN-P084	6398

7590 04/30/2008
WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

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 237 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 Customer Service Center of the Office of Patent Publication at (571)-272-4200.

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

Guillermo J. Rozas, Los Gatos, CA;

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **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)

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.

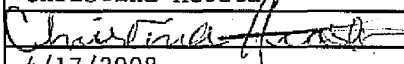
7590

01/17/2008

WAGNER, MURABITO & HAO LLP
 Third Floor
 Two North Market Street
 San Jose, CA 95113

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.

Christina Acosta	(Depositor's name)
	(Signature)
4/17/2008	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/102,127	04/07/2005	Guillermo J. Rozas	TRAN-P084	6398

TITLE OF INVENTION: MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1440	\$0	\$0	\$1440	04/17/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
OTTO, ALAN	2187	711-144000

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)

TRANSMETA CORPORATION

Santa Clara, California

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 10

4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)

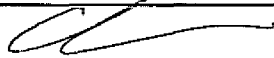
- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 50-4160 (enclose an extra copy of this form).

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

April 17, 2008

Typed or printed name Anthony C. Murabito

Registration No. 35,295

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.

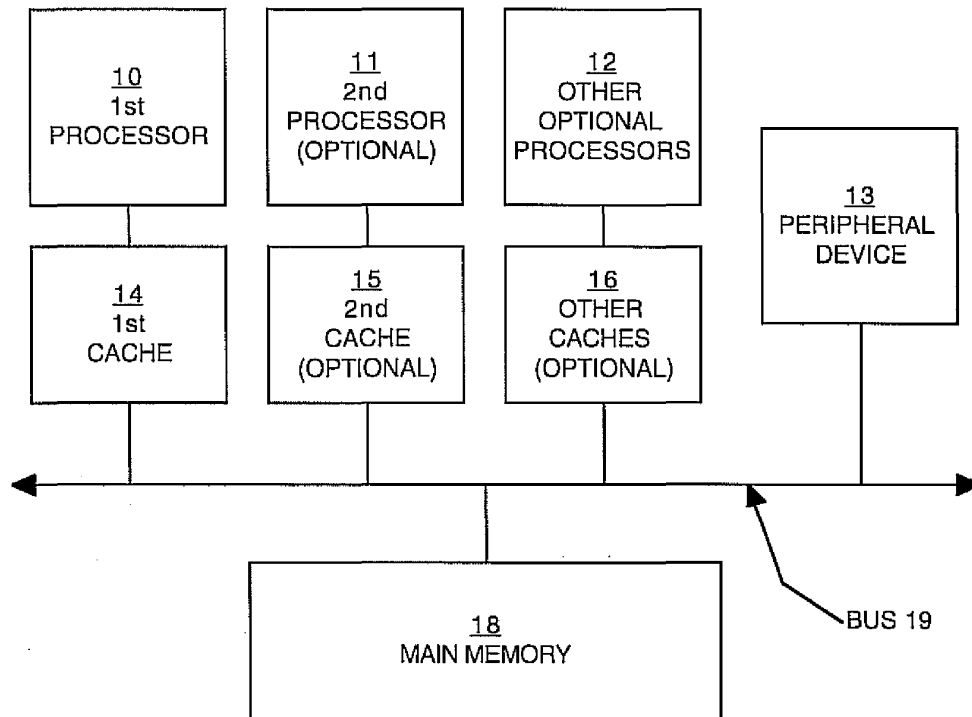


Figure 1

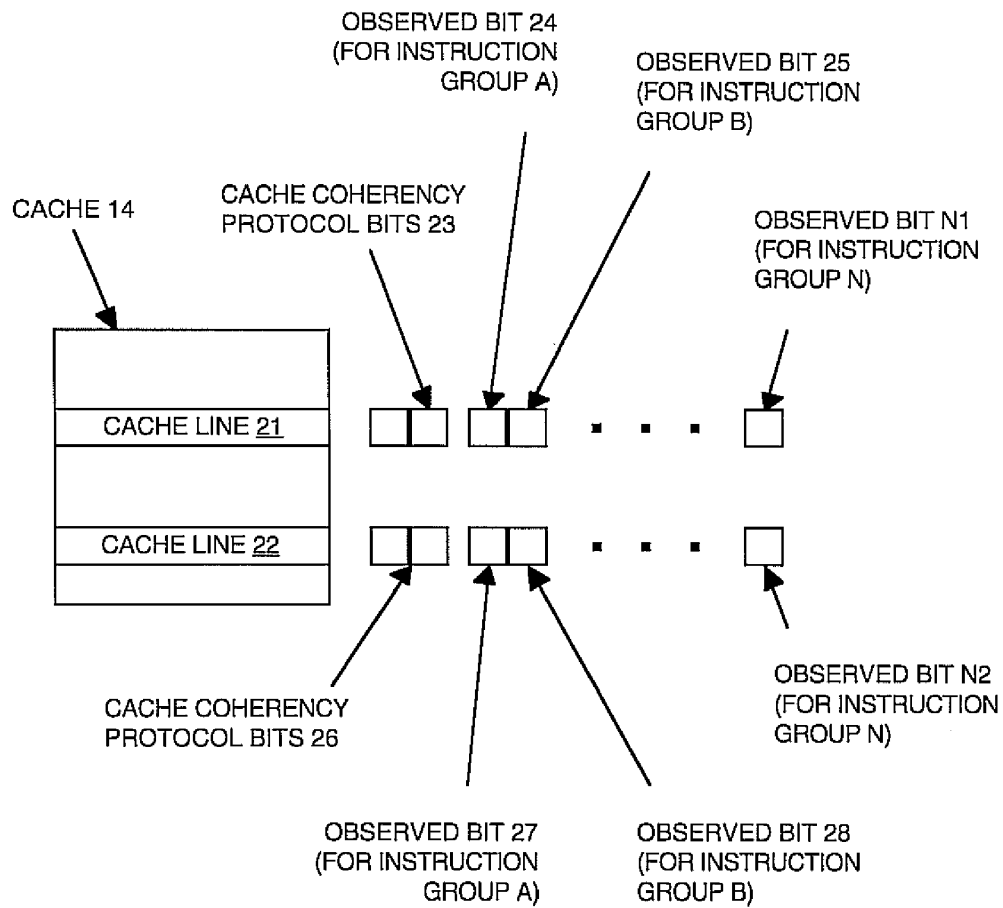


Figure 2

30

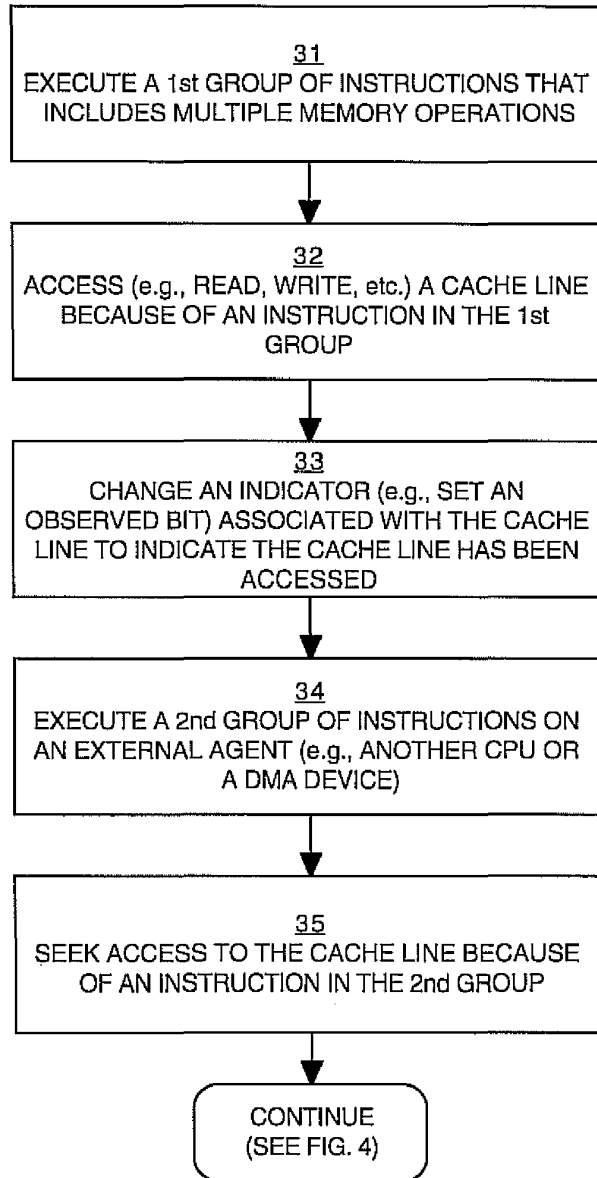


Figure 3

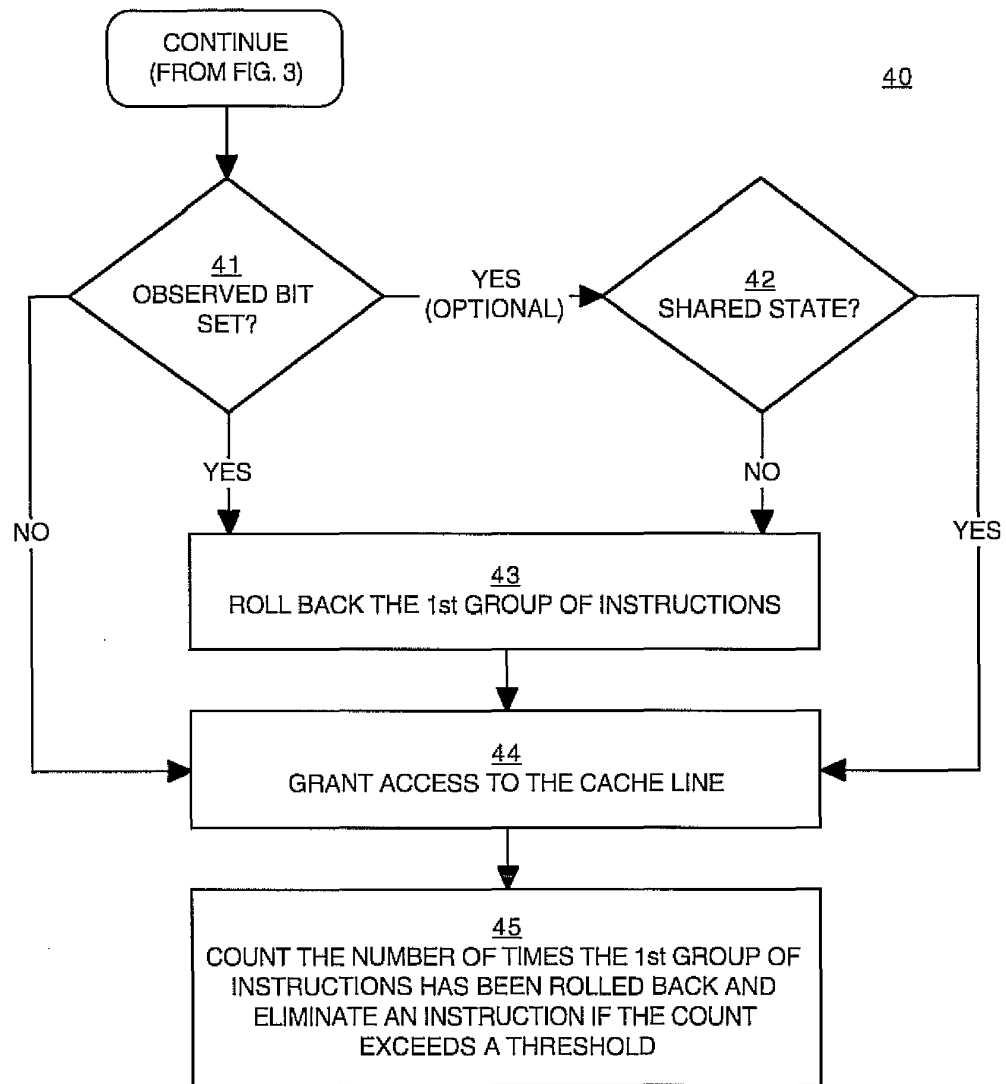


Figure 4

50

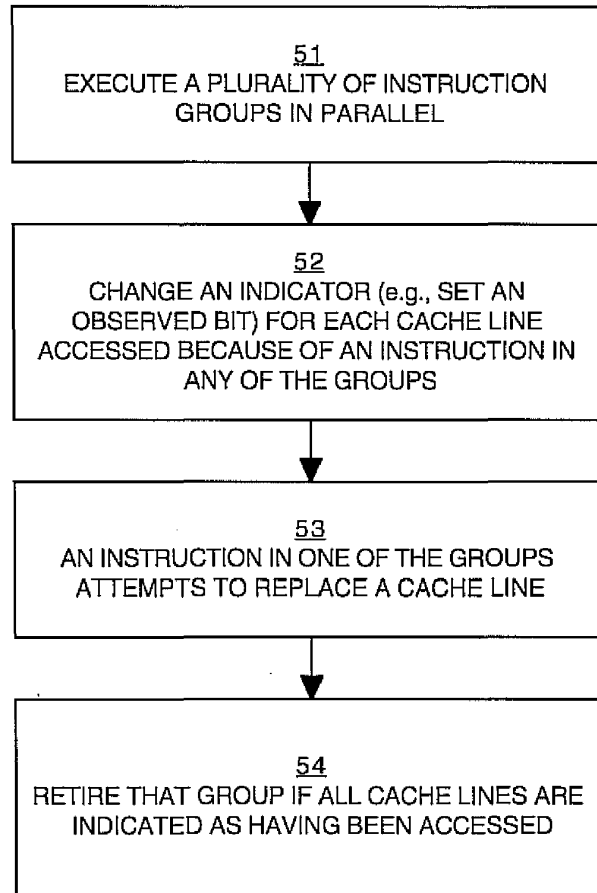


Figure 5

Electronic Patent Application Fee Transmittal				
Application Number:		11102127		
Filing Date:		07-Apr-2005		
Title of Invention:		MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY		
First Named Inventor/Applicant Name:		Guillermo J. Rozas		
Filer:		Anthony C. Murabito/Christina Acosta		
Attorney Docket Number:		TRAN-P084		
Filed as Large Entity				
Utility Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	1501	1	1440	1440
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Printed copy of patent - no color	8001	10	3	30
Total in USD (\$)				1470

Electronic Acknowledgement Receipt	
EFS ID:	3171079
Application Number:	11102127
International Application Number:	
Confirmation Number:	6398
Title of Invention:	MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY
First Named Inventor/Applicant Name:	Guillermo J. Rozas
Correspondence Address:	WAGNER, MURABITO & HAO LLP - Third Floor Two North Market Street San Jose CA 95113 US 4089389060 -
Filer:	Anthony C. Murabito/Christina Acosta
Filer Authorized By:	Anthony C. Murabito
Attorney Docket Number:	TRAN-P084
Receipt Date:	17-APR-2008
Filing Date:	07-APR-2005
Time Stamp:	17:53:06
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Electronic Funds Transfer
Payment was successfully received in RAM	\$ 1470

RAM confirmation Number	2920				
Deposit Account					
Authorized User					
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	TRAN-P084_Issue_Fee.pdf	67321	no	1
			2f389ea74615710d624d04ff827a7a071167630		
Warnings:					
Information:					
2	Drawings-only black and white line drawings	TRAN-P084_FIGURES.pdf	43577	no	5
			a83ad9421266a6e93adf3aa102c331592040a7d0		
Warnings:					
Information:					
3	Fee Worksheet (PTO-06)	fee-info.pdf	8360	no	2
			dda53a0d56cf68c00c9c0dc5b118583346929b5f		
Warnings:					
Information:					
Total Files Size (in bytes):			119258		
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. <u>New Applications Under 35 U.S.C. 111</u> 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. <u>National Stage of an International Application under 35 U.S.C. 371</u> 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. <u>New International Application Filed with the USPTO as a Receiving Office</u> 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

NOTICE OF ALLOWANCE AND FEE(S) DUE

7590 01/17/2008
WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

EXAMINER	
OTTO, ALAN	
ART UNIT	PAPER NUMBER
2187	

DATE MAILED: 01/17/2008

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/102,127	04/07/2005	Guillermo J. Rozas	TRAN-P084	6398
TITLE OF INVENTION: MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY				

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1440	\$0	\$0	\$1440	04/17/2008

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. **PROSECUTION ON THE MERITS IS CLOSED.** THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. **THIS STATUTORY PERIOD CANNOT BE EXTENDED.** SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

- A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.
- B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

- A. Pay TOTAL FEE(S) DUE shown above, or
- B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

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)

7590 01/17/2008

WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

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.
11/102,127	04/07/2005	Guillermo J. Rozas	TRAN-P084	6398

TITLE OF INVENTION: MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1440	\$0	\$0	\$1440	04/17/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
OTTO, ALAN	2187	711-144000

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, 1 _____
- (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. 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.
11/102,127	04/07/2005	Guillermo J. Rozas	TRAN-P084	6398

EXAMINER	
OTTO, ALAN	

ART UNIT	PAPER NUMBER
2187	

DATE MAILED: 01/17/2008

7590 01/17/2008
WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

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 271 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 271 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.

Notice of Allowability	Application No.	Applicant(s)	
	11/102,127	ROZAS, GUILLERMO J.	
	Examiner	Art Unit	
	Alan M. Otto	2187	

-- 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 amendment filed 10/29/2007.
2. ☒ The allowed claim(s) is/are 1-7, 10-18, 21-32, renumbered claims 1-28 respectively.
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)

- | | |
|---|--|
| <ol style="list-style-type: none"> 1. ☐ Notice of References Cited (PTO-892) 2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) 3. ☐ Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date _____ 4. ☐ Examiner's Comment Regarding Requirement for Deposit
of Biological Material | <ol style="list-style-type: none"> 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 _____ |
|---|--|

Detailed Action

The instant application having Application No. 11/102,127 has a total of 28 claims pending in the application, there are 4 independent claim and 24 dependent claims, all of which are ready for examination by the examiner. Claims 8-9 and 19-20 were cancelled by the amendment dated 6/11/2007.

Reasons for Allowance

1. The reasons for allowance can be found below:

The following is an examiner's statement of reasons for allowance:

2. **Claims 1 and 12** recite the limitation of "in response to said second instruction causing said cache line to be read, setting a second bit of said indicator while said first bit remains set to indicate said cache line has been accessed by both said first group and said second group of instructions." This limitation is taught by page 12, lines 5-9 of the specification and shown in fig. 2 in the drawings and is argued by applicant on page 14 of applicant's arguments filed 11/20/2007. This limitation, in combination with other recited limitations of claims 1 and 12 is not taught or suggested by the prior art of record.
3. **Claim 23** recites the limitation of "wherein said second state and said third state indicate said cache line has been accessed concurrently by both said first group and said second group of instructions." This limitation is taught by page 12, lines 5-9 of the specification and shown in fig. 2 in the drawings and is argued by applicant on page 14

of applicant's arguments filed 11/20/2007. This limitation, in combination with other recited limitations of claim 23 is not taught or suggested by the prior art of record.

4. **Claim 31** recites the limitation of "wherein said first and second indicator bits are both set to indicate said cache line has been accessed because of both first instruction and second instruction." This limitation is taught by page 12, lines 5-9 of the specification and shown in fig. 2 in the drawings and is argued by applicant on page 14 and pages 17-18 of applicant's arguments filed 11/20/2007. This limitation, in combination with other recited limitations of claim 31 is not taught or suggested by the prior art of record.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

CLOSING COMMENTS

Conclusion

a. STATUS OF CLAIMS IN THE APPLICATION

5. The following is a summary of the treatment and status of all claims in the application as recommended by **M.P.E.P. 707.07(i)**:

a(1) SUBJECT MATTER CONSIDERED ALLOWABLE

6. Per the instant office action, claims 1-7, 10-18, and 21-32 are allowed, and have been renumbered claims 1-28 respectively.

a(2) CLAIMS NO LONGER IN THE APPLICATION

7. Claims 8-9 and 19-20 were cancelled by the amendment dated 6/11/2007.

b. DIRECTION OF FUTURE CORRESPONDENCES

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Alan Otto whose telephone number is 571-270-1626.

The examiner can normally be reached on 8:00-5:30 M-F.

IMPORTANT NOTE

9. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald Sparks can be reached on 571-272-4201. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

10. 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). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AO

Ad



DONALD SPARKS
SUPERVISORY PATENT EXAMINER

Index of Claims 	Application/Control No. 11102127	Applicant(s)/Patent Under Reexamination ROZAS, GUILLERMO J.
	Examiner Otto, Alan M	Art Unit 2187

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

☐ Claims renumbered in the same order as presented by applicant
 ☐ CPA
 ☐ T.D.
 ☐ R.1.47

CLAIM		DATE							
Final	Original	02/26/2007	08/31/2007	12/11/2007					
1	1	✓	✓	=					
2	2	✓	✓	=					
3	3	✓	✓	=					
4	4	✓	✓	=					
5	5	✓	✓	=					
6	6	✓	✓	=					
7	7	✓	✓	=					
	8	✓	-	-					
	9	✓	-	-					
8	10	✓	✓	=					
9	11	✓	✓	=					
10	12	✓	✓	=					
11	13	✓	✓	=					
12	14	✓	✓	=					
13	15	✓	✓	=					
14	16	✓	✓	=					
15	17	✓	✓	=					
16	18	✓	✓	=					
	19	✓	-	-					
	20	✓	-	-					
17	21	✓	✓	=					
18	22	✓	✓	=					
19	23	✓	✓	=					
20	24	✓	✓	=					
21	25	✓	✓	=					
22	26	✓	✓	=					
23	27	✓	✓	=					
24	28	✓	✓	=					
25	29	✓	✓	=					
26	30	✓	✓	=					
27	31	✓	✓	=					
28	32	✓	✓	=					

Search Notes 	Application/Control No. 11102127	Applicant(s)/Patent Under Reexamination ROZAS, GUILLERMO J.
	Examiner Otto, Alan M	Art Unit 2187

SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES		
Search Notes	Date	Examiner
Updated East text search	12/11/07	AO
Updated NPL search - Dialog	12/11/07	AO
Updated Inventor name search (East and Palm)	12/11/07	AO
Consulted with examiners Brian Peugh and Kevin Verbrugge	12/11/07	AO

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
	Interference Search History Printout	12/11/07	AO

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	11487164	break\$4 or reduc\$4 or eliminat\$4 or segment\$4 or divid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L2	4916227	thread or instruction or code or program	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L3	224036	L2 near5 L1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L4	1300	fail\$3 near10 L3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L5	671	L4 and memory	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L6	140951	subdivid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L7	1972	L6 near5 L2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L8	431	L5 and processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L9	459	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L10	11803	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L11	176	L10 and L9	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L12	8514240	task or process or program or thread or instruction or command	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L13	106318	roll-back or rollback or undo or retry or reissue or redo or abort	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L14	14973	L13 near5 L12	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L15	11803	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L16	5120	L15 near20 L12	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L17	1956	shared adj cache	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L18	11803	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L19	80327	roll adj back or reissue or redo or retry	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L20	3756	L19 near3 (instruction or thread or process)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L21	318	L20 and L18	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L22	79	L17 and L21	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L23	1438	711/141.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L24	1208	711/144.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L25	907	711/145.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L26	870	711/146.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L27	1425	710/22.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L28	3130	L23 or L24 or L25 or L26	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L29	459	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L30	152	L29 and L28	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L31	907	711/145.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L32	870	711/146.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L33	1208	711/144.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L34	1438	711/141.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L35	3130	L34 or L33 or L31 or L32	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L36	137	711/133.ccls. and 711/134.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L37	3072	cache adj4 full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L38	3059	L10 near5 read\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L39	61	L38 and L9	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L40	8	dma and L39	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L41	231	thread near15 L10	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L42	80	L41 and L38	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L43	2311	711/118.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L44	9	L42 and L43	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L45	375	cache adj line adj tag	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L46	42	L45 near10 (read or access)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L47	55	L36 and cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L48	246	(thread\$4 or transaction or process or (group near5 instruction)) near5 (L10) near10 (bit or tag or indicator)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L49	30	(L10) same (atomic near3 transaction\$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L50	40639	"711".clas.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L51	61	L42 and L50	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L52	54777	roll adj back or reissue or redo	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L53	1438	711/141.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L54	1208	711/144.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L55	907	711/145.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L56	870	711/146.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L57	459	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L58	3130	L53 or L54 or L55 or L56	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L59	152	L57 and L58	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L60	11803	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L61	81	L60 and L59	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L62	459	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L63	1158965	replace or replacement	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L64	11803	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L65	1092	L63 near5 L64	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L66	106318	roll-back or rollback or undo or retry or reissue or redo or abort	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L67	4338	cache near4 full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L68	4024	count\$3 near8 L66	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L69	13873	count\$3 near8 fail\$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L70	54777	roll adj back or reissue or redo	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L71	1174	L70 near3 (instruction or thread or process)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L72	117	L71 and L18	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L73	459	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L74	2311	711/118.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L75	1808	L52 near7 (instruction or thread or process)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L76	164	L75 and cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L77	41	L27 and L28	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L78	188354	(memory or cache) adj3 (access or accesses or request)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L79	152	L62 and L35	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L80	0	L79 and cache near3 full near6 L66	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L81	140951	subdivid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L82	9	rozás-guillermo.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L83	3	L10 near50 L75	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L84	2	L45 near10 (thread)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L85	7	L20 near20 L18	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L86	45	L45 near10 (thread or instruction)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L87	13	L14 and L30	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L88	32	L21 and (L73 or L74)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L89	2	L72 and (L73 or L74)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L90	3	L20 near10 L18	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L91	2	"5701432".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L92	47	L16 and L30	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L93	10	10/637168	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L94	3	L66 near40 L65	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L95	2	"6006299".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L96	36	L67 near40 L66	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L97	2	L69 and L79	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L98	4	L68 and L79	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L99	2	"20050268049"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L100	9	L78 near10 subdivide	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L101	10	L36 and L66	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L102	112	(thread\$4 or process or (group near5 instruction)) near5 (L10) near10 (bit or tag or indicator)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 13:21
L103	23	L79 and cache near3 full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L104	12	cache near3 full near6 retry	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L105	9	cache near3 full near6 delay	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L106	13	L37 near20 L66	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L107	2	"6625694".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L108	23	L47 and full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L109	27	L79 and cache near3 wait	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L110	796	((transaction\$3 near3 system) or (transaction\$3 near3 memory)) same (cache adj line)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L111	58	L48 and L110 not L102	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12
L112	157	thread' near5 (cache adj (line or tag))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 10:12

EAST Search History

L113	124	L112 not (L49 or L111 or L102)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 13:01
L115	11	(indicator and bit and cache adj line and read and processor and memory).clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 13:03
L116	4	(indicator and bit and cache adj line and read and processor and memory and execut\$3).clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 13:05
L117	0	(indicator and bit and cache adj line and read and processor and memory and execut\$3 and second adj bit).clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 13:16
L119	15295	710/22.ccls. or 711/141.ccls. or 711/144.ccls. or 711/145.ccls. or 711/146.ccls. or 711/130.ccls. or 711/133.ccls. or 711/134.ccls. or 711/118.ccls. or g06f12/00.ipc.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 13:20
L120	51	l102 and l119	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 13:21
L121	14	l120 and (multi-thread\$3 or multithread\$3 or second adj thread\$3 or multi adj thread\$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/12/10 13:22

Dial · g DataStar.

options

logoff

feedback

help



databases

easy
search

Advanced Search:

Inspec - 1898 to date (INZZ)

limit

Search history:

No.	Database	Search term	Info added since	Results	
CP		[Clipboard]		0	-
1	INZZ	cache ADJ line	unrestricted	294	show titles
2	INZZ	indicator AND bit AND 1	unrestricted	0	-
3	INZZ	(thread\$4 OR process OR group NEAR instruction) NEAR 1	unrestricted	4	show titles
4	INZZ	multithreaded OR multi-threaded OR second ADJ thread OR multi ADJ threaded	unrestricted	2382	show titles
5	INZZ	(indicator OR bit) NEAR (read OR access\$3) AND 1	unrestricted	0	-
6	INZZ	unavailable OR full OR busy	unrestricted	226425	show titles
7	INZZ	6 NEAR 1	unrestricted	1	show titles
8	INZZ	roll-back OR rollback OR undo OR retry OR reissue OR redo OR abort	unrestricted	2874	show titles
9	INZZ	8 AND 1	unrestricted	3	show titles

[hide](#) | [delete all search steps...](#) | [delete individual search steps...](#)
Enter your search term(s): [Search tips](#) ☐ Thesaurus mapping
 whole document

 Information added since: or: (YYYYMMDD)

☐ Images

Select special search terms from the following list(s):

- ☒ Publication year 1950-
- ☒ Publication year 1898-1949



10-31-07

RCE/

Docket Number: TRAN-P084

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
Patent Application

I hereby certify that this transmittal of the below described documents is being deposited with the United States Postal Service in an envelope bearing Express Mail Postage and an Express Mail label, with the below serial number, addressed to the Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450, on the below date of deposit.			
Express Mail Label No.:	EM118684921US	Name of Person Making the Deposit:	Anthony Chou
Date of Deposit:	10/29/07	Signature of the Person Making the Deposit:	<i>Anthony Chou</i>

In re Application of: Guillermo J. Rozas

Application No.: 11/102,127

Examiner: Otto, A.

Filed: 04/07/05

Art Unit: 2187

Confirmation No.: 6398

For: MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY

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

REQUEST FOR CONTINUED EXAMINATION (RCE) TRANSMITTAL
(SUBSECTION(B) OF 35 U.S.C. § 132)

This is a Request for Continued Examination (RCE) under 37 C.F.R. § 1.114 for the above-identified application.

1. Submission required under a filing under 37 C.F.R. § 1.114

a. ☐ Previously submitted

i. ☐ Consider the amendment(s)/ reply under 37 C.F.R. § 1.116 previously filed on _____
(Any unentered amendment(s) referred to above will be entered)

ii. ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____

iii. ☐ Other _____

b. ☒ Enclosed

11/01/2007 ATRINH 00000059 11102127

i. ☒ Amendment/Reply

01 FC:1801

810.00 0P

ii. ☐ Affidavit(s)/Declaration(s)

iii. ☐ Information Disclosure Statement (IDS)

iv. ☐ Other _____

2. Miscellaneous

- a. ☐ Suspension of action on the above-identified application is requested under 37 C.F.R. § 1.103(c) for a period of _____ months.
(period of suspension shall not exceed 3 months: Fee under 37 C.F.R. § 1.17(l) required)
- b. Other _____

Extension of Term

3. The proceedings herein are for a patent application and the provisions of 37 C.F.R. 1.136 apply.

- (a) ☐ Applicant petitions for an extension of time under 37 C.F.R. 1.136
(fees: 37 C.F.R. 1.17(a)-(d) for the total number of months checked below:)

Extension	Fee
☐ one month	\$120.00
☐ two months	\$460.00
☐ three months	\$1,050.00
☐ four months	\$1,640.00

Fee \$ _____

If an additional extension of time is required, please consider this a petition therefor.

- (b) ☒ Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition for extension of time.

FEES DUE

The RCE fee under 37 C.F.R. § 1.17(e) is required by 37 C.F.R. § 1.114 when the RCE is filed.

CLAIMS					
	NO. OF CLAIMS		EXTRA CLAIMS	RATE	FEES
Basic Application Fee					\$810.00
Total Claims	28	Minus 32=	0	X \$50 =	\$0.00
Independent Claims	4	Minus 4=	0	X \$210 =	\$0.00
If multiple dependent claims are presented, add \$360.00					\$0.00
TOTAL APPLICATION FEE DUE					\$810.00

PAYMENT OF FEES

1. The full fee due in connection with this communication is provided as follows:
- ☒ The Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No.: 50-4160.

A duplicate copy of this authorization is enclosed.

- [X] A check in the amount of \$810.00
- [] A check in the amount of \$ _____
- [] Charge any fees required or credit any overpayments associated with this filing to Deposit Account No.: 50-4160.

Please direct all correspondence concerning the above-identified application to the following address:

MURABITO HAO & BARNES LLP
Two North Market Street, Third Floor
San Jose, California 95113
(408) 938-9060
Customer No: 45590

Respectfully submitted,

Date: 10/29/07

By: 
William A. Zarbis
Reg. No. 46,120



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Rozas

Serial No.: 11/102,127

Filed: April 7, 2005

For: MEMORY MANAGEMENT
METHODS AND SYSTEMS
THAT SUPPORT CACHE
CONSISTENCY

Examiner: OTTO, A.

Art Unit: 2187

Confirmation No.: 6398

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

RESPONSE TO AN OFFICE ACTION

Dear Sir:

In response to the Office Action mailed September 18, 2007, and per the Request for Continued Examination filed herewith, please consider the following remarks and enter the following amendments.

The claims are presented beginning on page 2 of this response. Remarks begin on page 13 of this response.

Serial No. 11/102,127
Examiner: OTTO, A.

- 1 -

Art Unit 2187
TRAN-P084

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of managing memory in a computer system, said method comprising:

executing a first group of instructions on a first processor, wherein said first group of instructions comprises multiple memory operations and a first instruction that when executed causes a cache line of a cache memory to be read; and

in response to said first instruction causing said cache line to be read, ~~changing~~ setting a first bit of an indicator associated with said cache line ~~first group of instructions~~ to indicate that said cache line has been read, wherein said cache line is so indicated as having been read until execution of said first group of instructions is ended, wherein multiple groups of instructions are executable in parallel by said first processor and wherein said indicator comprises a bit per ~~cache line~~ for each group in said groups of instructions;

executing a second group of instructions on said first processor, wherein said second group of instructions comprises multiple memory operations and a second instruction that when executed causes said cache line to be read; and

in response to said second instruction causing said cache line to be read, setting a second bit of said indicator while said first bit remains set to indicate said cache line has been accessed by both said first group and said second group of instructions.

2. (Currently Amended) The method of Claim 1 further comprising:

executing a third ~~second~~ group of instructions, wherein said third ~~second~~ group of instructions includes a third ~~second~~ instruction that when executed causes said cache line to be accessed, and wherein said ~~first group of~~

~~instructions is associated with a thread executing on said first processor and said~~
~~second~~ third group of instructions is associated with a thread executing on an
agent external to said first processor;

in response to said third ~~second~~ instruction seeking access to said cache
line, using a value of said indicator to determine whether said cache line has
been read and generating a result thereof; and

handling said first group and said second group of instructions according
to said result.

3. (Currently Amended) The method of Claim 2 wherein said handling
comprises rolling back said first group and said second group of instructions
before granting said access provided said indicator indicates said cache line has
been read by said first instruction and said second instruction, and otherwise
granting said access without rolling back said first group and said second group
of instructions.

4. (Currently Amended) The method of Claim 3 wherein said handling
further comprises:

determining a state of said cache line, wherein said state is specified
according to a cache coherency protocol comprising at least a modified state, a
shared state and an invalid state; and

rolling back said first group and said second group of instructions and
then granting said access provided said cache line is in other than said shared
state and said indicator indicates said cache line has been read by said first
instruction and said second instruction, and otherwise granting said access

without rolling back said first group and said second group of instructions only
when said cache line is in said shared state.

5. (Currently Amended) The method of Claim 3 further comprising
maintaining a first count of how many times said first group of instructions has
been rolled back and a second count of how many times said second group of
instructions has been rolled back.

6. (Currently Amended) The method of Claim 5 further comprising
eliminating one or more instructions from said first group of instructions if said
first count exceeds a threshold and eliminating one or more instructions from
said second group of instructions if said second count exceeds said threshold.

7. (Currently Amended) The method of Claim 2 wherein said agent is
selected from the group consisting of: a second processor, and or an
input/output agent capable of direct memory access (DMA).

8-9. (Canceled).

10. (Currently Amended) The method of Claim 1 further comprising ~~[[:]]~~
~~setting a bit associated with said cache line in response to said first instruction~~
~~causing said cache line to be read;~~ and clearing said first bit when said execution
of said first group of instructions is ended, wherein said second bit remains set
until execution of said second group of instructions is ended.

11. (Original) The method of Claim 1 further comprising:
executing a plurality of groups of instructions in parallel;
changing an indicator associated with each cache line in said cache that
has been accessed because of an instruction in any of said plurality of groups of
instructions; and

provided all cache lines in said cache memory are indicated as having been
accessed, rolling back a group of instructions that includes an instruction that when
executed attempts to replace information in a cache line of said cache memory, and
otherwise continuing said executing of said plurality of groups of instructions.

12. (Currently Amended) A computer system comprising:
a first processor;
a cache memory used by said first processor; and
a memory unit coupled to said first processor, said memory unit
containing instructions that when executed implement a memory management
method, said method comprising:

executing a first group of instructions using said first processor,
wherein said first group of instructions includes multiple memory
operations and a first instruction that when executed causes a cache line
of said cache memory to be read; and

in response to said first instruction causing said cache line to be
read, ~~changing~~ setting a first bit of an indicator associated with said cache
line ~~first group of instructions~~ to indicate that said cache line has been read,
wherein said cache line is so indicated as having been read until execution
of said first group of instructions is ended, wherein multiple groups of
instructions are executable in parallel by said first processor and wherein

said indicator comprises a bit ~~per cache line~~ for each group of said groups of instructions being executed at a same time by said first processor;

executing a second group of instructions using said first processor,
wherein said second group of instructions comprises multiple memory
operations and a second instruction that when executed causes said
cache line to be read; and

in response to said second instruction causing said cache line to be
read, setting a second bit of said indicator while said first bit remains set
to indicate said cache line has been accessed by both said first group and
said second group of instructions.

13. (Currently Amended) The computer system of Claim 12 wherein said method further comprises:

executing a third ~~second~~ group of instructions using an agent coupled to said cache memory, wherein said third ~~second~~ group of instructions includes a third ~~second~~ instruction that when executed causes said cache line to be accessed, and wherein said agent is external to said first processor;

in response to said s third ~~second~~ instruction seeking access to said cache line, using a value of said indicator to determine whether said cache line has been read and generating a result thereof; and

handling said first group and said second group of instructions according to said result.

14. (Currently Amended) The computer system of Claim 13 wherein said agent is selected from the group consisting of: a second processor, and ~~or~~ an input/output agent capable of direct memory access (DMA).

15. (Currently Amended) The computer system of Claim 13 wherein said method further comprises rolling back said first group and said second group of instructions before granting said access provided said indicator indicates said cache line has been read by said first instruction and said second instruction, and otherwise granting said access without rolling back said first group and said second group of instructions.

16. (Currently Amended) The computer system of Claim 15 wherein said method further comprises:

determining a state of said cache line, wherein said state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state; and

rolling back said first group and said second group of instructions and then granting said access provided said cache line is in other than said shared state and said indicator indicates said cache line has been read by said first instruction and said second instruction, and otherwise granting said access without rolling back said first group and said second group of instructions only when said cache line is in said shared state.

17. (Currently Amended) The computer system of Claim 15 wherein said method further comprises maintaining a first count of how many times said first group of instructions has been rolled back and a second count of how many times said second group of instructions has been rolled back.

18. (Currently Amended) The computer system of Claim 17 wherein said method further comprises eliminating one or more instructions from said first group of instructions if said first count exceeds a threshold and eliminating one or more instructions from said second group of instructions if said second count exceeds said threshold.

19-20. (Canceled).

21. (Currently Amended) The computer system of Claim 12 wherein said method further comprises ~~[[:]] setting a bit associated with said cache line in response to said first instruction causing said cache line to be read; and~~ clearing said first bit when said execution of said first group of instructions is ended, wherein said second bit remains set until execution of said second group of instructions is ended.

22. (Original) The computer system of Claim 12 wherein said method further comprises:

executing a plurality of groups of instructions in parallel on said first processor;

changing an indicator associated with each cache line in said cache memory that has been accessed because of an instruction in any of said plurality of groups of instructions; and

provided all cache lines in said cache memory are indicated as having been accessed, rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache

memory, and otherwise continuing said executing of said plurality of groups of instructions.

23. (Currently Amended) A method of managing shared memory in a computer system, said method comprising:

executing a first group of instructions on a first processor;

associating a first state with a cache line in a cache comprising a plurality of cache lines, wherein said first state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state; and

associating a second state with said cache line during said executing of said first group of instructions, wherein said second state indicates that said cache line has been accessed because of a first instruction of said first group of instructions and wherein said second state is cleared when execution of said first group of instructions is ended, wherein multiple groups of instructions are executable in parallel by said first processor and wherein said second state comprises a bit for each cache line in said plurality of cache lines for each group in said groups of instructions;

executing a second group of instructions on said first processor;

associating a third state with said cache line during said executing of said second group of instructions, wherein said third state indicates that said cache line has been accessed because of a second instruction of said second group of instructions and wherein said third state is cleared when execution of said second group of instructions is ended, wherein said second state and said third state indicate said cache line has been accessed concurrently by both said first group and said second group of instructions.

24. (Currently Amended) The method of Claim 23 wherein said first group and said second group of instructions each includes multiple memory operations.

25. (Currently Amended) The method of Claim 23 further comprising:
executing a third ~~second~~ group of instructions, wherein said third ~~second~~ group of instructions includes a third ~~second~~ instruction that when executed causes said cache line to be accessed and wherein said first group and said second group of instructions are is associated with respective threads ~~a thread~~ executing on said first processor and wherein said third ~~second~~ group of instructions is associated with a thread executing on an agent external to said first processor;

in response to said third ~~second~~ instruction seeking access to said cache line, using said second state and said second state to indicate said cache line has been accessed because of said first instruction and said second instruction; and

rolling back said first group and said second group of instructions before granting said access.

26. (Original) The method of Claim 25 wherein said rolling back is only performed provided said cache line is in other than said shared state.

27. (Currently Amended) The method of Claim 25 further comprising maintaining a first count of how many times said first group of instructions is rolled back and a second count of how many times said second group of instructions is rolled back.

28. (Currently Amended) The method of Claim 27 further comprising eliminating one or more instructions from said first group of instructions if said first count exceeds a threshold and eliminating one or more instructions from said second group of instructions if said second count exceeds said threshold.

29. (Currently Amended) The method of Claim 25 wherein said agent is selected from the group consisting of: a second processor, and or an input/output agent capable of direct memory access (DMA).

30. (Currently Amended) The method of Claim 23 wherein said second state is indicated by [[:]] setting a first bit associated with said cache line in response to said first instruction causing said cache line to be read, wherein said first bit is cleared [[:]] ~~and clearing said bit~~ when said execution of said first group of instructions is ended, and wherein said third state is indicated by setting a second bit associated with said cache line in response to said second instruction causing said cache line to be read, wherein said second bit is cleared when said execution of said second group of instructions is ended.

31. (Currently Amended) A computer system comprising:
a processor;
a cache memory coupled to said processor and having a plurality of cache lines; and
a memory unit coupled to said processor, wherein said memory unit has a plurality of entries and a plurality of indicator bits per entry, wherein a first ~~an~~ indicator bit of said plurality of indicator bits is set to indicate that a cache line of said cache memory has been accessed because of a first ~~an~~ instruction of a first

group of instructions and wherein said first indicator bit is cleared when execution of said first group of instructions is ended, wherein a second indicator bit of said plurality of indicator bits is set to indicate that said cache line has been accessed because of a second instruction of a second group of instructions and wherein said second indicator bit is cleared when execution of said second group of instructions is ended, wherein said first and second indicator bits are both set to indicate said cache line has been accessed because of both said first instruction and said second instruction, and wherein each of said entries covers the address space of said cache memory such that a cache line maps uniquely to an entry but an entry maps to multiple cache lines, wherein multiple groups of instructions are executable in parallel by said processor and wherein said plurality of indicator bits comprises an indicator bit for each group in said groups of instructions for each cache line in said plurality of cache lines.

32. (Original) The computer system of Claim 31 wherein a subset of address bits of said cache line is used to identify said entry, wherein in response to said cache line being accessed, an indicator bit of said entry corresponding to said subset of address bits is set.

REMARKS

Claims 1-7, 10-18 and 21-32 are pending. Claims 1-7, 10, 12-18, 21, 23-25 and 27-31 are amended. No new matter is added as a result of the claim amendments. Support for the claim amendments can be found at least on page 12 (lines 5-9), page 15 (lines 19-24), and page 17 (line 7) of the instant application.

35 U.S.C. § 102 Rejections

According to the instant Office Action, Claims 1-7, 10, 12-18, 21 and 23-30 are rejected under 35 U.S.C. § 102(e) as being anticipated by Jacobson et al. ("Jacobson;" U.S. Patent No. 6,938,130). The Applicants have reviewed the cited reference and respectfully submit that the present invention as recited in Claims 1-7, 10, 12-18, 21 and 23-30 is not shown or suggested by Jacobson.

According to the present claimed invention, there are multiple indicator bits associated with each cache line in a cache. Each indicator bit is associated with a respective group of instructions. If a first group of instructions is executed, and an instruction in the first group results in a particular cache line being accessed, then an indicator bit associated with both the cache line and the first group is set. Similarly, if a second group of instructions is executed, and an instruction in the second group results in the particular cache line being accessed, then an indicator bit associated with both the cache line and the second group is set.

Applicants respectfully submit that Jacobson does not show or suggest these claimed features. Jacobson appears to utilize only one "store-marking" bit

per cache line, in contrast to the present claimed invention. Jacobson also teaches a "thread ID" per cache line that, according to the Office Action, includes multiple bits. However, Jacobson's thread ID only identifies a single thread (the thread that has the lock on the cache line).

Therefore, Applicants respectfully submit that Jacobson does not show or suggest "executing a first group of instructions on a first processor, wherein said first group of instructions comprises multiple memory operations and a first instruction that when executed causes a cache line of a cache memory to be read; in response to said first instruction causing said cache line to be read, setting a first bit of an indicator associated with said cache line to indicate that said cache line has been read, wherein said cache line is so indicated as having been read until execution of said first group of instructions is ended, ... executing a second group of instructions on said first processor, wherein said second group of instructions comprises multiple memory operations and a second instruction that when executed causes said cache line to be read; and in response to said second instruction causing said cache line to be read, setting a second bit of said indicator while said first bit remains set to indicate said cache line has been accessed by both said first group and said second group of instructions" as recited in independent Claim 1 (emphases added) and as similarly recited in independent Claims 12 and 23.

In summary, Applicants respectfully submit that Jacobson does not show or suggest the limitations of independent Claims 1, 12 and 23 and that these claims are allowable over the cited reference.

Each of the Claims 2-7 and 10 includes all of the limitations of independent Claim 1 plus additional limitations. Each of the Claims 13-18 and 21 includes all of the limitations of independent Claim 12 plus additional limitations. Each of the Claims 24-30 includes all of the limitations of independent Claim 23 plus additional limitations. Applicants respectfully submit that Jacobson does not show or suggest the limitations of Claims 2-7, 10, 13-18, 21 and 24-30 in combination with the limitations of their respective base claims. In other words, Applicants respectfully submit that Claims 2-7, 10, 13-18, 21 and 24-30 are also in condition for allowance as depending from allowable claims.

Consequently, Applicants respectfully submit that the basis for rejecting Claims 1-7, 10, 12-18, 21 and 23-30 under 35 U.S.C. § 102(e) is traversed.

35 U.S.C. § 103 Rejections

Claims 11 and 22

According to the instant Office Action, Claims 11 and 22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Jacobson in view of Arimilli et al. ("Arimilli;" U.S. Patent No. 6,263,407). The Applicants have reviewed the cited references and respectfully submit that the present invention as recited in Claims 11 and 22 is not shown or suggested by Jacobson and Arimilli, alone or in combination.

Claim 11 is dependent on Claim 1 and recites additional limitations, and Claim 22 is dependent on Claim 12 and recites additional limitations. Hence, by demonstrating that the cited references do not show or suggest the limitations of

Claims 1 and 12, it is also demonstrated that the cited references do not show or suggest the limitations of Claims 11 and 22.

As presented above, Applicants respectfully submit that Jacobson does not show or suggest “executing a first group of instructions on a first processor, wherein said first group of instructions comprises multiple memory operations and a first instruction that when executed causes a cache line of a cache memory to be read; in response to said first instruction causing said cache line to be read, setting a first bit of an indicator associated with said cache line to indicate that said cache line has been read, wherein said cache line is so indicated as having been read until execution of said first group of instructions is ended, ... executing a second group of instructions on said first processor, wherein said second group of instructions comprises multiple memory operations and a second instruction that when executed causes said cache line to be read; and in response to said second instruction causing said cache line to be read, setting a second bit of said indicator while said first bit remains set to indicate said cache line has been accessed by both said first group and said second group of instructions” as recited in Claim 1 (emphases added) and as similarly recited in Claim 12.

Applicants also submit that Arimilli does not overcome the shortcomings of Jacobson. More specifically, Applicants respectfully submit that Jacobson and Arimilli, alone or in combination, do not show or suggest an indicator bit per instruction group per cache line as recited in the claims.

In summary, Applicants respectfully submit that Jacobson and Arimilli, alone or in combination, do not show or suggest the limitations of independent Claims 1 and 12 and that these claims are allowable over the cited references. As such, Applicants respectfully submit that Claims 11 and 22 are also in condition for allowance as depending from allowable claims.

Consequently, Applicants respectfully submit that the basis for rejecting Claims 11 and 22 under 35 U.S.C. § 103(a) is traversed.

Claims 31-32

According to the instant Office Action, Claims 31-32 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Jacobson in view of Masri et al. ("Masri," U.S. Patent No. 6,625,694). The Applicants have reviewed the cited references and respectfully submit that the present invention as recited in Claims 31-32 is not shown or suggested by Jacobson and Masri, alone or in combination.

By rationale similar to that presented above, Applicants respectfully submit that Jacobson does not show or suggest "a memory unit ..., wherein said memory unit has a plurality of entries and a plurality of indicator bits per entry, wherein a first indicator bit of said plurality of indicator bits is set to indicate that a cache line of said cache memory has been accessed because of a first instruction of a first group of instructions and wherein said first indicator bit is cleared when execution of said first group of instructions is ended, wherein a second indicator bit of said plurality of indicator bits is set to indicate that said cache line has been accessed because of a second instruction of a second

group of instructions and wherein said second indicator bit is cleared,when execution of said second group of instructions is ended, wherein said first and second indicator bits are both set to indicate said cache line has been accessed because of both said first instruction and said second instruction" as recited in independent Claim 31 (emphases added).

Applicants also submit that Masri does not overcome the shortcomings of Jacobson. More specifically, Applicants respectfully submit that Jacobson and Masri, alone or in combination, do not show or suggest an indicator bit per instruction group per cache line as recited in the claims.

In summary, Applicants respectfully submit that Jacobson and Masri, alone or in combination, do not show or suggest the limitations of independent Claim 31 and that Claim 31 is allowable over the cited references.

Claim 32 includes all of the limitations of independent Claim 31 plus additional limitations. Applicants respectfully submit that Jacobson does not show or suggest the limitations of Claim 32 in combination with the limitations of Claim 31. In other words, Applicants respectfully submit that Claim 32 is also in condition for allowance as depending from allowable claims.

Consequently, Applicants respectfully submit that the basis for rejecting Claims 31-32 under 35 U.S.C. § 103(a) is traversed.

Conclusions

In light of the above remarks, reconsideration of the rejected claims is respectfully requested. Based on the arguments presented above, it is respectfully asserted that Claims 1-7, 10-18 and 21-32 overcome the rejections of record and, therefore, allowance of these claims is solicited.

The Examiner is invited to contact Applicants' undersigned representative if the Examiner believes such action would expedite resolution of the present Application.

Dated: 10/29/07

Respectfully submitted,
MURABITO HAO & BARNES LLP



William A. Zarbis
Registration No. 46,120

Address: MURABITO HAO & BARNES LLP
Two North Market Street
Third Floor
San Jose, California 95113

Telephone: (408) 938-9060



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.
11/102,127	04/07/2005	Guillermo J. Rozas	TRAN-P084	6398

7590 09/18/2007
WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

EXAMINER

OTTO, ALAN

ART UNIT	PAPER NUMBER
----------	--------------

2187

MAIL DATE	DELIVERY MODE
-----------	---------------

09/18/2007

PAPER

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

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

Office Action Summary	Application No.	Applicant(s)	
	11/102,127	ROZAS, GUILLERMO J.	
	Examiner	Art Unit	
	Alan M. Otto	2187	

-- 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 11 June 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) 1-7, 10-18 and 21-32 is/are pending in the application.

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

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) 1-7, 10-18 and 21-32 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 _____ 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

The instant application having Application No. 11/102127 has a total of 28 claims pending in the application, there are 4 independent claim and 24 dependent claims, all of which are ready for examination by the examiner. Claims 8-9 and 19-20 were cancelled by the amendment dated 6/11/2007.

REJECTIONS BASED ON PRIOR ART

Claim Rejections – 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-7, 10, 12-18, 21, and 23-30 are rejected under 35 U.S.C. 102(e) as being anticipated by Jacobson et al.'130 (U.S. Patent No. 6,938,130), herein referred to as Jacobson et al.'130.

3. Referring to claims 1 and 12, Jacobson et al.'130 discloses as claimed, a method of managing memory in a computer system (**see Abstract**), said method comprising: executing a first group of instructions (**see col. 6, lines 42-46, regarding a critical section of code being executed**) on a first processor (**see 1, showing processor 101**), wherein said first group of instructions comprises multiple memory operations (**see col. 7, lines 31-34, regarding loading and store marking cache lines in**

response to loads and stores, which constitute memory operations) and a first instruction that when executed causes a cache line of a cache memory to be read (**see col. 7, lines 37-39, regarding a load operation, causing a cache line to be read**); and in response to said first instruction causing said cache line to be read, changing an indicator associated with said first group of instructions to indicate that said cache line has been read (**see col. 5, lines 27-33, regarding a load-marking bit which is an indicator that is set when data from the cache line has been read**), wherein said cache line is so indicated as having been read until execution of said first group of instructions is ended (**see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache**) wherein multiple groups of instructions are executable in parallel by said first processor (**see col. 2, lines 51-56, regarding multiple threads executing**) and wherein said indicator comprises a bit per cache line for each group in said groups of instructions (**see fig. 12, showing thread id 1211, where the thread id represents bits in a cache line as shown in fig. 12, and as the bits are associated with a group of instruction. The thread id is part of an indicator, as explained in col. 11, lines 26-29. The thread id could show 01 representing cache line 1, or 10 representing cache line 2 for example. Each thread would have a bit associated with a cache line in a thread id**).

4. As to claims 2 and 13, Jacobson et al.'130 also discloses the method of Claim 1 further comprising: executing a second group of instructions (**see col. 6, lines 47-50, regarding monitoring other threads for interfering data accesses**), wherein said

second group of instructions includes a second instruction that when executed causes said cache line to be accessed (**see col. 7, lines 1-5, regarding a store by another thread on a load marked line**), and wherein said first group of instructions is associated with a thread executing on said first processor (**see fig. 1, processor 101, and col. 5, lines 19-30, regarding the first processor sending load and store operations which could be load marked or store marked**) and said second group of instructions is associated with a thread executing on an agent external to said first processor (**see fig. 1, processor 102, and col. 7, lines 8-13, regarding the interfering access thread coming from another processor accessing a cache line**); in response to said second instruction seeking access to said cache line (**see col. 7, lines 8-13, regarding determining if a cache line is being accessed by another processor**), using a value of said indicator to determine whether said cache line has been read and generating a result thereof (**see col. 7, lines 1-13, regarding the use of the load marked or store marked indicator to see if an interfering data access has taken place**); and handling said first group of instructions according to said result (**see col. 6, lines 55-58, regarding re-executing the first group of instructions if an interfering data access has taken place**).

5. As to claims 3 and 15, Jacobson et al.'130 also discloses the method of Claim 2 wherein said handling comprises rolling back said first group of instructions before granting said access (**see col. 6, lines 55-58, regarding discarding the execution of the first group of instructions, and re-executing them if an interfering access is detected**) provided said indicator indicates said cache line has been read by said first

instruction (**see col. 11, lines 54-59 regarding an interfering access, which means that the indicator indicated the cache line was read. Also see col. 7, lines 1-5**), and otherwise granting said access without rolling back said first group of instructions (**see col. 6, lines 47-54, regarding continually monitoring for interfering accesses. If none are detected, the first group of instructions is not rolled back, but is committed, and the second group of instructions are granted access, as they are not interfering**).

6. As to claims 4 and 16, Jacobson et al.'130 also discloses the method of Claim 4 wherein said handling further comprises: determining a state of said cache line, (**see col. 11, lines 8-16, and fig. 12, regarding state information for a cache line, and see col. 11, lines 44-50, regarding making decisions based on the state, which was determined**) wherein said state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state (**see col. 11, lines 8-16, and fig. 12, regarding state information for a cache line. State bits valid 1204, dirty 1206, and owned 1208 can show those states**); and rolling back said first group of instructions (**see col. 11, lines 54-59, regarding rolling back and repeating the first instructions**) and then granting said access provided said cache line is in other than said shared state (**see col. 11, lines 48-50, regarding shared ownership being ok for reading**) and said indicator indicates said cache line has been read by said first instruction (**see col. 11, lines 54-59 regarding an interfering access, which means that the indicator indicated the cache line was read. Also see col. 7, lines 1-5**), and otherwise granting said access without rolling

back said first group of instructions (**see col. 6, lines 47-54, regarding continually monitoring for interfering accesses. If none are detected, the first group of instructions is not rolled back, but is committed, and the second group of instructions are granted access, as they are not interfering**).

7. As to claims 5 and 17, Jacobson et al.'130 also discloses the method of Claim 3 further comprising maintaining a count of how many times said first group of instructions has been rolled back (**see col. 6, lines 59-61, regarding counting how many times the system tries to re-execute the first group of instructions**).

8. As to claims 6 and 18, Jacobson et al.'130 also discloses the method of Claim 5 further comprising eliminating one or more instructions from said first group of instructions if said count exceeds a threshold (**see col. 6, lines 58-57, regarding executing alternative code if the count has exceeded a threshold. This alternative code as stated could revert back to the conventional technique of trying to acquire a lock on a critical section. Thus, this would eliminate possible instructions from the new technique taught by Jacobson, such as monitoring if an interfering data access takes place, or committing the changes after a transactional execution**).

9. As to claims 7 and 14, Jacobson et al.'130 also discloses the method of Claim 2 wherein said agent is a second processor or an input/output agent capable of direct memory access (DMA) (**see fig. 1, processor 102, and col. 7, lines 8-13, regarding the interfering access thread coming from another processor accessing a cache line**).

10. As to claims 10 and 21, Jacobson et al.'130 also discloses the method of Claim 10 further comprising: setting a bit associated with said cache line in response to said first instruction causing said cache line to be read (**see col. 5, lines 27-33, regarding setting a load marking bit when a cache line is loaded**); and clearing said bit when said execution of said first group of instructions is ended (**see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache**).

11. Referring to claim 23, Jacobson et al.'130 discloses as claimed, a method of managing shared memory in a computer system (**see Abstract**), said method comprising: executing a first group of instructions (**see col. 6, lines 42-46, regarding a critical section of code being executed**) on a first processor (**see 1, showing processor 101**); associating a first state with a cache line in a cache comprising a plurality of cache lines (**see col. 11, lines 8-16, and fig. 12, regarding state information associated with a cache line. Note that the caches as shown in fig. 1 have a plurality of cache lines**), wherein said first state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state (**see col. 11, lines 8-16, and fig. 12, regarding state information for a cache line. State bits valid 1204, dirty 1206, and owned 1208 can show those states**); and associating a second state with said cache line during said executing (**see fig. 12, store mark 1210, regarding showing a state where cache line has been written to. Also see col. 5, lines 34-37, regarding a load marking bit not shown in fig. 12**), wherein said second state indicates that said cache line has been accessed

because of a first instruction of said first group of instructions (**see col. 5, lines 27-33, regarding a load-marking bit which is an indicator that is set when data from the cache line has been read**) and wherein said second state is cleared when execution of said first group of instructions is ended (**see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache**) wherein multiple groups of instructions are executable in parallel by said first processor (**see col. 2, lines 51-56, regarding multiple threads executing**) and wherein said second state comprises a bit for each cache line in said plurality of cache lines for each group in said groups of instructions (**see fig. 12, showing thread id 1211, where the thread id represents bits in a cache line as shown in fig. 12, and as the bits are associated with a group of instruction. The thread id is part of an indicator, as explained in col. 11, lines 26-29. The thread id could show 01 representing cache line 1, or 10 representing cache line 2 for example. Each thread would have a bit associated with a cache line in a thread id. The thread id is showing which cache line has been accessed by which thread in conjunction with load marking bits**).

12. As to claim 24, Jacobson et al.'130 also discloses the method of Claim 23 wherein said first group of instructions includes multiple memory operations. (**see col. 7, lines 31-34, regarding loading and store marking cache lines in response to loads and stores, which constitute memory operations**)

13. As to claim 25, Jacobson et al.'130 also discloses the method of Claim 23 further comprising: executing a second group of instructions (**see col. 6, lines 47-50,**

regarding monitoring other threads for interfering data accesses), wherein said second group of instructions includes a second instruction that when executed causes said cache line to be accessed (**see col. 7, lines 1-5, regarding a store by another thread on a load marked line**) and wherein said first group of instructions is associated with a thread executing on said first processor (**see fig. 1, processor 101, and col. 5, lines 19-30, regarding the first processor sending load and store operations which could be load marked or store marked**) and wherein said second group of instructions is associated with a thread executing on an agent external to said first processor (**see fig. 1, processor 102, and col. 7, lines 8-13, regarding the interfering access thread coming from another processor accessing a cache line**); in response to said second instruction seeking access to said cache line (**see col. 7, lines 8-13, regarding determining if a cache line is being accessed by another processor**), using said second state to indicate said cache line has been accessed because of said first instruction (**see col. 7, lines 1-13, regarding the use of the load marked or store marked indicator to see if an interfering data access has taken place**); and rolling back said first group of instructions before granting said access (**see col. 6, lines 55-58, regarding re-executing the first group of instructions if an interfering data access has taken place**).

14. As to claim 26, Jacobson et al.'130 also discloses the method of Claim 25 wherein said rolling back is only performed provided said cache line is in other than said shared state (**see col. 11, lines 48-50, regarding shared ownership being ok for**

reading, and therefore the rolling back is not necessary if the cache line is in the shared state).

15. As to claim 27, Jacobson et al.'130 also discloses the method of Claim 25 further comprising maintaining a count of how many times said first group of instructions is rolled back **(see col. 6, lines 59-61, regarding counting how many times the system tries to re-execute the first group of instructions).**

16. As to claim 28, Jacobson et al.'130 also discloses the method of Claim 27 further comprising eliminating one or more instructions from said first group of instructions if said count exceeds a threshold **(see col. 6, lines 58-57, regarding executing alternative code if the count has exceeded a threshold. This alternative code as stated could revert back to the conventional technique of trying to acquire a lock on a critical section. Thus, this would eliminate possible instructions from the new technique taught by Jacobson, such as monitoring if an interfering data access takes place, or committing the changes after a transactional execution).**

17. As to claim 29, Jacobson et al.'130 also discloses the method of Claim 25 wherein said agent is a second processor or an input/output agent capable of direct memory access (DMA) **(see fig. 1, processor 102, and col. 7, lines 8-13, regarding the interfering access thread coming from another processor accessing a cache line).**

18. As to claim 30, Jacobson et al.'130 also discloses the method of Claim 23 wherein said second state is indicated by: setting a bit associated with said cache line in response to said first instruction causing said cache line to be read **(see col. 5, lines**

27-33, regarding setting a load marking bit when a cache line is loaded); and clearing said bit when said execution of said first group of instructions is ended (see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache).

Claim Rejections - 35 USC § 103

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

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

20. Claims 11 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jacobson et al.'130 in view of Arimilli et al.'407 (U.S. Patent No. 6,263,407), herein referred to as Arimilli et al.'407.

21. As to claims 11 and 22, Jacobson et al.'130 discloses as claimed, the method of Claim 1 further comprising: executing a plurality of groups of instructions in parallel (**see Abstract**); changing an indicator associated with each cache line in said cache that has been accessed because of an instruction in any of said plurality of groups of instructions (**see col. 3, lines 39-46, regarding changing a load mark indicator associated with each cache line when a cache line has been accessed from a plurality of groups of instructions**);

Jacobson et al.'130 discloses the claimed invention except for where all cache lines in said cache memory are indicated as having been accessed, rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory, and otherwise continuing said executing of said plurality of groups of instructions.

However, Arimilli et al.'407 discloses where all cache lines in said cache memory are indicated as having been accessed (**see col. 9, lines 2-6, regarding the write queue of the cache being full, hence all cache lines have been accessed**), rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory (**see col. 9, lines 2-6, regarding a transaction being retried. Because the transaction was headed for the write queue, it is a write transaction that was attempting to write or replace information in a cache line**), and otherwise continuing said executing of said plurality of groups of instructions (**see col. 9, lines 2-6, where the transaction would take place if the write queue were not busy, which would take place if the cache lines had not been accessed, and were available**).

Jacobson et al.'130 and Arimilli et al.'407 are analogous art because they are from the same field of endeavor of writing to and maintaining cache systems in multiprocessor systems (**see Jacobson et al.'130, col. 2, lines 50-64, regarding transactional execution with caches involving multiple threads and processors, and see Arimilli et al.'407, col. 1, lines 8-17, regarding maintaining cache coherency in a multiprocessor system**).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Jacobson et al.'130 to comprise where all cache lines in said cache memory are indicated as having been accessed, rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory, and otherwise continuing said executing of said plurality of groups of instructions, as taught by Arimilli et al.'407, in order to provide for more predictable software behavior and consistent software timing (see col. 8, lines 56-64, regarding advantages of operating in precise mode, where retries can happen).

22. Claims 31 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jacobson et al.'130 in view of Masri et al.'694 (U.S. Patent No. 6,625,694), herein referred to as Masri et al.'694.

23. Referring to claim 31, Jacobson et al.'130 discloses as claimed, a computer system comprising: a processor (see fig. 1, processor 101); a cache memory coupled to said processor and having a plurality of cache lines (see fig. 1, cache unit 115); and a memory unit coupled to said processor (see fig. 1, cache memory 115, where the cache memory can constitute a memory unit), wherein said memory unit has a plurality of entries and a plurality of indicator bits per entry (see fig. 1, load marking bits 116 which are indicator bits. Also see fig. 12 showing a thread id where the thread id is part of an indicator, as explained in col. 11, lines 26-29), wherein an indicator bit of said plurality of indicator bits is set to indicate that a cache line of said cache memory has been accessed because of an instruction of a group of instructions

(see col. 5, lines 27-33, regarding setting a load marking bit when a cache line is loaded. Store marking bits can be set to signify a store. Also see col. 5, lines 56-58) and wherein said indicator bit is cleared when execution of said group of instructions is ended (see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache) wherein multiple groups of instructions are executable in parallel by said first processor (see col. 2, lines 51-56, regarding multiple threads executing) and wherein said plurality of indicator bits comprises an indicator bit for each group in said groups of instructions for each cache line in said plurality of cache lines (see fig. 12, showing thread id 1211, where the thread id represents bits in a cache line as shown in fig. 12, and as the bits are associated with a group of instruction. The thread id is part of an indicator, as explained in col. 11, lines 26-29. The thread id could show 01 representing cache line 1, or 10 representing cache line 2 for example. Each thread would have a bit associated with a cache line in a thread id).

Jacobson et al.'130 discloses the claimed invention except for where each of said entries covers the address space of said cache memory such that a cache line maps uniquely to an entry but an entry maps to multiple cache lines.

However, Masri et al.'694 discloses where each of said entries covers the address space of said cache memory such that a cache line maps uniquely to an entry but an entry maps to multiple cache lines (see col. 4, lines 49-58, regarding a set associative cache, where one directory entry corresponds to multiple cache lines).

Jacobson et al.'130 and Masri et al.'694 are analogous art because they are from the same field of endeavor of writing to and maintaining cache systems in multiprocessor systems (**see Jacobson et al.'130, col. 2, lines 50-64, regarding transactional execution with caches involving multiple threads and processors, and see Masri et al.'694, abstract, regarding a replacement mechanism for caches in a multiprocessor system**).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Jacobson et al.'130 to comprise where each of said entries covers the address space of said cache memory such that a cache line maps uniquely to an entry but an entry maps to multiple cache lines, as taught by Masri et al.'694, in order to provide a replacement algorithm that can improve the cache-hit rate (**see col. 2, lines 5-19, discussing advantages of a set associative cache. In addition, using a set associative cache is well known in the art**).

24. As to claim 32, Jacobson et al.'130 discloses the claimed invention except for the computer system of Claim 31 wherein a subset of address bits of said cache line is used to identify said entry, wherein in response to said cache line being accessed, an indicator bit of said entry corresponding to said subset of address bits is set.

However, Masri et al.'694 discloses where a subset of address bits of said cache line is used to identify said entry (**see col. 5, lines 1-7, regarding each memory line being identified by a location offset**), wherein in response to said cache line being accessed, an indicator bit of said entry corresponding to said subset of address bits is set (**see col. 5, lines 15-20, where state field 326 is an indicator bit corresponding**).

Jacobson et al.'130 and Masri et al.'694 are analogous art because they are from the same field of endeavor of writing to and maintaining cache systems in multiprocessor systems (see Jacobson et al.'130, col. 2, lines 50-64, regarding **transactional execution with caches involving multiple threads and processors, and see Masri et al.'694, abstract, regarding a replacement mechanism for caches in a multiprocessor system**).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Jacobson et al.'130 to comprise where a subset of address bits of said cache line is used to identify said entry, wherein in response to said cache line being accessed, an indicator bit of said entry corresponding to said subset of address bits is set, as taught by Masri et al.'694, in order provide a replacement algorithm that can improve the cache-hit rate (see col. 2, lines 5-19, discussing **advantages of a set associative cache**. In addition, a well-known advantage of using a set associative cache is that index bits don't have to include the part of main memory which is specified by the cache, thus allowing for fewer bits).

ACKNOWLEDGEMENT OF ISSUES RAISED BY THE APPLICANT

Response to Amendment

25. Applicant's arguments filed June 11th, 2007 were received and fully considered, however they were not deemed persuasive.

ARGUMENTS CONCERNING PRIOR ART REJECTIONS

26. Regarding the applicant's traversal of the rejection of claims 1-7, 10, 12-18, 21, and 23-30, applicant asserts that Jacobson et al.'130 does not show specifically "wherein said indicator comprises a bit per cache line for each group in said groups of instructions" as recited in independent claim 1, and similarly in claims 12 and 23. Applicant points to a store-marking bit used by Jacobson. However, in fig. 12, and in col. 11, lines 26-29, cache lines also contain a thread id which indicates which thread has been store marked by a cache line. See fig. 12, showing thread id 1211, where the thread id represents bits in a cache line as shown in fig. 12, and as the bits are associated with a group of instruction. The thread id is part of an indicator, as explained in col. 11, lines 26-29. The thread id could show 01 representing cache line 1, or 10 representing cache line 2 for example. Each thread would have a bit associated with a cache line in a thread id. Therefore, examiner asserts that Jacobson et al.'130 does teach the elements recited in claim 1, and similarly in claims 12 and 23.

27. Regarding the applicant's traversal of the rejection of claims 11-22, the applicant argues that neither Arimilli or Jacobson do not show specifically "wherein said indicator comprises a bit per cache line for each group in said groups of instructions" from claim 1. The examiner's explanation in regards to claim 1 applies.

28. Regarding the applicant's traversal of the rejection of claims 31-32, applicant argues that neither Masri or Jacobson teach "a plurality of indicator bits per entry" and "wherein said plurality of indicator bits comprises an indicator bit for each group in said groups of instructions for each cache line in said plurality of cache lines." However, Jacobson utilizes store and load marking bits, in addition to a thread id as shown in fig.

12. In col. 11, lines 26-29, cache lines also contain a thread id which indicates which thread has been store marked by a cache line. Therefore the thread id is part of an indicator bit. See fig. 12, showing thread id 1211, where the thread id represents bits in a cache line as shown in fig. 12, and as the bits are associated with a group of instruction. The thread id could show 01 representing cache line 1, or 10 representing cache line 2 for example. Each thread would have a bit associated with a cache line in a thread id. Therefore, examiner asserts that Jacobson et al.'130 does teach the elements recited in claims 31-32 along with Masri.

CLOSING COMMENTS

Conclusion

a. STATUS OF CLAIMS IN THE APPLICATION

29. The following is a summary of the treatment and status of all claims in the application as recommended by M.P.E.P. 707.07(i):

a(1) CLAIMS NO LONGER IN THE APPLICATION

30. Claims 8-9 and 19-20 were cancelled by the amendment dated 6/11/07.

a(2) CLAIMS REJECTED IN THE APPLICATION

31. Per the instant office action, claims 1-7, 10-18, and 21-32 have received a second action on the merits and are subject of a final rejection.

b. DIRECTION OF FUTURE CORRESPONDENCES

32. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Alan Otto whose telephone number is 571-270-1626. The examiner can normally be reached on 8:00-5:30 M-F.

IMPORTANT NOTE

33. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald Sparks can be reached on 571-272-4201. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

34. 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). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AO
As


DONALD SPARKS
SUPERVISORY EXAMINER

Index of Claims 	Application/Control No. 11102127	Applicant(s)/Patent Under Reexamination ROZAS, GUILLERMO J.
	Examiner Otto, Alan M	Art Unit 2109

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

☐ Claims renumbered in the same order as presented by applicant
 ☐ CPA
 ☐ T.D.
 ☐ R.1.47

CLAIM		DATE							
Final	Original	02/26/2007	08/31/2007						
	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	✓	✓						
	31	✓	✓						
	32	✓	✓						

Search Notes 	Application/Control No. 11102127	Applicant(s)/Patent Under Reexamination ROZAS, GUILLERMO J.
	Examiner Otto, Alan M	Art Unit 2109

SEARCHED			
Class	Subclass	Date	Examiner
711	118, 130, 133, 134, 141, 144, 145, 146	2/26/07	AO

SEARCH NOTES		
Search Notes	Date	Examiner
East Search	2/26/07	AO
NPL Search	2/26/07	AO
Updated East text search	8/31/07	AO
Updated NPL search - Dialog	8/31/07	AO
Inventor name search (East and Palm)	8/31/07	AO
Consulted with examiner Brian Peugh	8/31/07	AO
711/118,130,141,144,145,146, 710/22 (text search - see attached search results)	8/31/07	AO

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S103	4604223	thread or instruction or code or program	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:20
S104	10919066	break\$4 or reduc\$4 or eliminat\$4 or segment\$4 or divid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:20
S105	204801	S103 near5 S104	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:22
S106	1178	fail\$3 near10 S105	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:24
S107	581	S106 and memory	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:24
S109	133508	subdivid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:35
S110	1850	S109 near5 S103	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:37
S112	363	S107 and processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 17:01

EAST Search History

S11 3	9	roz-as-guillermo.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S11 4	35900	"711".clas.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S11 5	2249	711/118.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S11 6	11477	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S11 7	447	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S11 8	173	S116 and S117	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S11 9	2973	S116 near5 read\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S12 0	60	S119 and S117	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S12 1	8	dma and S120	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S12 2	213	thread near15 S116	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S12 3	73	S122 and S119	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S12 4	56	S123 and S114	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S12 5	9	S123 and S115	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S12 6	53888	roll adj back or reissue or redo	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S12 7	1751	S126 near7 (instruction or thread or process)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S12 8	159	S127 and cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S12 9	3	S116 near50 S127	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S13 0	53888	roll adj back or reissue or redo	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S13 1	11477	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S13 2	1136	S130 near3 (instruction or thread or process)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S13 3	114	S132 and S131	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S13 4	447	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S13 5	2249	711/118.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S13 6	2	S133 and (S134 or S135)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S13 7	78577	roll adj back or reissue or redo or retry	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S13 8	3638	S137 near3 (instruction or thread or process)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S13 9	312	S138 and S131	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S14 0	32	S139 and (S134 or S135)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S14 1	370	cache adj line adj tag	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S14 2	3	S138 near10 S131	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S14 3	42	S141 near10 (read or access)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S14 4	2	S141 near10 (thread)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S14 5	1899	shared adj cache	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S14 6	77	S145 and S139	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S14 7	7	S138 near20 S131	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S14 8	43	S141 near10 (thread or instruction)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S14 9	1412	711/141.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S15 0	1191	711/144.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S15 1	894	711/145.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S15 2	854	711/146.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S15 3	1392	710/22.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S15 4	3076	S149 or S150 or S151 or S152	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S15 5	447	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S15 6	152	S155 and S154	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S15 7	41	S153 and S154	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S15 8	8321859	task or process or program or thread or instruction or command	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S15 9	103604	roll-back or rollback or undo or retry or reissue or redo or abort	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S16 0	14485	S159 near5 S158	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S16 1	13	S160 and S156	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S16 2	11477	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S16 3	4964	S162 near20 S158	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S16 4	2	"5701432".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S16 5	47	S163 and S156	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S16 6	8321859	task or process or program or thread or instruction or command	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S16 7	11477	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S16 8	4964	S167 near20 S166	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S16 9	447	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S17 0	854	711/146.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S17 1	894	711/145.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S17 2	1191	711/144.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S17 3	1412	711/141.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S17 4	3076	S173 or S172 or S171 or S170	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S17 5	152	S169 and S174	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S17 6	47	S168 and S175	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S17 7	81	S167 and S175	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S17 8	34	S177 not S176	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S17 9	8	10/637168	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S18 0	447	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S18 1	1133021	replace or replacement	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S18 2	11477	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S18 3	1055	S181 near5 S182	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S18 4	103604	roll-back or rollback or undo or retry or reissue or redo or abort	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S18 5	3	S184 near40 S183	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S18 6	2	"6006299".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S18 7	4183	cache near4 full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S18 8	1412	711/141.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S18 9	34	S187 near40 S184	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S19 0	3899	count\$3 near8 S184	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S19 1	13491	count\$3 near8 fail\$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S19 2	1191	711/144.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S19 3	894	711/145.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S19 4	854	711/146.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S19 5	3076	S188 or S192 or S193 or S194	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S19 6	152	S180 and S195	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S19 7	2	S191 and S196	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S19 8	4	S190 and S196	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S19 9	0	S196 and cache near3 full near6 S184	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S20 0	12	cache near3 full near6 retry	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S20 1	23	S196 and cache near3 full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S20 2	8	cache near3 full near6 delay	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 15:53
S20 3	134	711/133.ccls. and 711/134.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S20 4	53	S203 and cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S20 5	23	S204 and full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S20 6	2955	cache adj4 full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S20 7	12	S206 near20 S184	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S20 8	2	"6625694".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43

EAST Search History

S209	11278389	break\$4 or reduc\$4 or eliminat\$4 or segment\$4 or divid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S210	27	S196 and cache near3 wait	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S211	181009	(memory or cache) adj3 (access or accesses or request)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S212	138259	subdivid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S213	2	"20050268049"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:43
S214	9	S211 near10 subdivide	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 14:18
S215	10	S203 and S184	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 13:51
S216	109	(thread\$4 or process or (group near5 instruction)) near5 (S116) near10 (bit or tag or indicator)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 15:54

EAST Search History

S21 7	769	((transaction\$3 near3 system) or (transaction\$3 near3 memory)) same (cache adj line)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 15:54
S21 8	237	(thread\$4 or transaction or process or (group near5 instruction)) near5 (S116) near10 (bit or tag or indicator)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 15:55
S21 9	55	S218 and S217 not S216	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 16:00
S22 0	30	(S116) same (atomic near3 transaction\$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 16:01
S22 1	142	thread near5 (cache adj (line or tag))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 16:11
S22 2	111	S221 not (S220 or S219 or S216)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/28 16:11

Dial · g DataStar

options

logoff

feedback

help

databases

easy
search

Advanced Search:

Inspec - 1898 to date (INZZ)

limit

Search history:

No.	Database	Search term	Info added since	Results	
CP		[Clipboard]		0	-
1	INZZ	cache ADJ line	unrestricted	286	show titles
2	INZZ	unavailable OR full OR busy	unrestricted	220858	show titles
3	INZZ	1 NEAR 2	unrestricted	1	show titles
4	INZZ	roll-back OR rollback OR undo OR retry OR reissue OR redo OR abort	unrestricted	2826	show titles
5	INZZ	4 AND 1	unrestricted	3	show titles
6	INZZ	1 AND 2	unrestricted	14	show titles
7	INZZ	use ADJ bit	unrestricted	160	show titles
10	INZZ	7 AND 1	unrestricted	0	-
11	INZZ	cache	unrestricted	14069	show titles
12	INZZ	11 AND 7	unrestricted	1	show titles

[hide](#) | [delete all search steps...](#) | [delete individual search steps...](#)Enter your search term(s): [Search tips](#) ☐ Thesaurus mapping Information added since: or:
(YYYYMMDD)

search

☐ Images

Select special search terms from the following list(s):

- ☒ Publication year 1950-
- ☒ Publication year 1898-1949
- ☒ Inspec thesaurus - browse headings
- ☒ Inspec thesaurus - enter a term



Attorney Docket No.: TRAN-P084

ITW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

I hereby certify that this transmittal of the below described document is being deposited with the United States Postal Service in an envelope bearing First Class Postage and addressed to the Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450, on the below date of deposit.

Date of Deposit:	06/08/07	Name of Person Making the Deposit:	Julie Giaramita	Signature of the Person Making the Deposit:	
------------------	----------	------------------------------------	-----------------	---	--

In re Application of: Guillermo J. Rozas

Application No.: 11/102,127

Examiner: Otto, A.

Filed: 04/07/05

Art Unit: 2109

Confirmation No.: 6398

For: MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY

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

AMENDMENT TRANSMITTAL

1. Transmitted herewith is an amendment for this application

☒ Transmitted herewith is a response to an office action for the above identified patent application.

(15 sheets)

Transmitted herewith are _____ sheets of substitute formal drawings.

Other: _____

2. Applicant is other than a small entity

Extension of Term

3. The proceedings herein are for a patent application and the provisions of 37 C.F.R. 1.136 apply.

(a) ☐ Applicant petitions for an extension of time under 37 C.F.R. 1.136
(fees: 37 C.F.R. 1.17(a)-(d) for the total number of months checked below:)

<u>Extension</u>	<u>Fee</u>
☐ one month	\$120.00
☐ two months	\$450.00
☐ three months	\$1,020.00
☐ four months	\$1,590.00
☐ five months	\$2,160.00
	<u>Fee \$</u>

If an additional extension of time is required, please consider this a petition therefor.

(b) ☒ Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition for extension of time.

Fee Calculation

4. The fee for claims (37 C.F.R. 1.16(b)-(d)) has been calculated as shown below:

(for other than a small entity)					
Fee Items	Claims Remaining After Amendment	Highest Number of Claims Previously Paid For	Present Extra Claims	Fee Rate	Total
Total Claims	28	- 32 =	0	x \$50.00	\$0.00
Independent Claims	4	- 4 =	0	x \$200.00	\$0.00
Multiple Dependent Claim Fee (one or more, first added by this amendment)				\$360.00	\$0.00
Total Fees					\$0.00

PAYMENT OF FEES

5. The full fee due in connection with this communication is provided as follows:
- ☒ [X] The Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No.: 50-4160. A duplicate copy of this authorization is enclosed.
- ☐ [] A check in the amount of \$
- ☐ [] Charge any fees required or credit any overpayments associated with this filing to Deposit Account No.: 50-4160.

Please direct all correspondence concerning the above-identified application to the following address:

MURABITO HAO & BARNES LLP
 Two North Market Street, Third Floor
 San Jose, California 95113
 (408) 938-9060
 Customer No: 45590

Respectfully submitted,

Date: 6/8/07

By: William A. Zarbis
 William A. Zarbis
 Reg. No. 48,120



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Rozas

Serial No.: 11/102,127

Filed: April 7, 2005

For: MEMORY MANAGEMENT
METHODS AND SYSTEMS
THAT SUPPORT CACHE
CONSISTENCY

Examiner: OTTO, A.

Art Unit: 2109

Confirmation No.: 6398

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

RESPONSE TO AN OFFICE ACTION

Dear Sir:

In response to the Office Action mailed March 8, 2007, please consider the following remarks and enter the following amendments.

The claims are presented beginning on page 2 of this response. Remarks begin on page 11 of this response.

Serial No. 11/102,127
Examiner: OTTO, A.

- 1 -

Art Unit 2109
TRAN-P084

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of managing memory in a computer system, said method comprising:

executing a first group of instructions on a first processor, wherein said first group of instructions comprises multiple memory operations and a first instruction that when executed causes a cache line of a cache memory to be read; and

in response to said first instruction causing said cache line to be read, changing an indicator associated with said first group of instructions to indicate that said cache line has been read, wherein said cache line is so indicated as having been read until execution of said first group of instructions is ended, wherein multiple groups of instructions are executable in parallel by said first processor and wherein said indicator comprises a bit per cache line for each group in said groups of instructions.

2. (Currently Amended) The method of Claim 1 further comprising:

executing a second group of instructions, wherein said second group of instructions includes a second instruction that when executed causes said cache line to be accessed, and wherein said first group of instructions is associated with a thread executing on said [[a]] first processor and said second group of instructions is associated with a thread executing on an agent external to said first processor;

in response to said second instruction seeking access to said cache line, using a value of said indicator to determine whether said cache line has been read and generating a result thereof; and

handling said first group of instructions according to said result.

3. (Original) The method of Claim 2 wherein said handling comprises rolling back said first group of instructions before granting said access provided said indicator indicates said cache line has been read by said first instruction, and otherwise granting said access without rolling back said first group of instructions.

4. (Original) The method of Claim 3 wherein said handling further comprises:

determining a state of said cache line, wherein said state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state; and

rolling back said first group of instructions and then granting said access provided said cache line is in other than said shared state and said indicator indicates said cache line has been read by said first instruction, and otherwise granting said access without rolling back said first group of instructions.

5. (Original) The method of Claim 3 further comprising maintaining a count of how many times said first group of instructions has been rolled back.

6. (Original) The method of Claim 5 further comprising eliminating one or more instructions from said first group of instructions if said count exceeds a threshold.

7. (Original) The method of Claim 2 wherein said agent is a second processor or an input/output agent capable of direct memory access (DMA).

8-9. (Canceled).

10. (Original) The method of Claim 1 further comprising:

setting a bit associated with said cache line in response to said first instruction causing said cache line to be read; and
clearing said bit when said execution of said first group of instructions is ended.

11. (Original) The method of Claim 1 further comprising:

executing a plurality of groups of instructions in parallel;
changing an indicator associated with each cache line in said cache that has been accessed because of an instruction in any of said plurality of groups of instructions; and
provided all cache lines in said cache memory are indicated as having been accessed, rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory, and otherwise continuing said executing of said plurality of groups of instructions.

12. (Currently Amended) A computer system comprising:

a first processor;
a cache memory used by said first processor; and
a memory unit coupled to said first processor, said memory unit containing instructions that when executed implement a memory management method, said method comprising:

executing a first group of instructions using said first processor,
wherein said first group of instructions includes multiple memory
operations and a first instruction that when executed causes a cache line
of said cache memory to be read; and

in response to said first instruction causing said cache line to be
read, changing an indicator associated with said first group of instructions
to indicate that said cache line has been read, wherein said cache line is
so indicated as having been read until execution of said first group of
instructions is ended, wherein multiple groups of instructions are
executable in parallel by said first processor and wherein said indicator
comprises a bit per cache line for each group of said groups of
instructions being executed at a same time by said first processor.

13. (Original) The computer system of Claim 12 wherein said method
further comprises:

executing a second group of instructions using an agent coupled to said
cache memory, wherein said second group of instructions includes a second
instruction that when executed causes said cache line to be accessed, and
wherein said agent is external to said first processor;

in response to said second instruction seeking access to said cache line,
using a value of said indicator to determine whether said cache line has been
read and generating a result thereof; and

handling said first group of instructions according to said result.

14. (Original) The computer system of Claim 13 wherein said agent is a second processor or an input/output agent capable of direct memory access (DMA).

15. (Original) The computer system of Claim 13 wherein said method further comprises rolling back said first group of instructions before granting said access provided said indicator indicates said cache line has been read by said first instruction, and otherwise granting said access without rolling back said first group of instructions.

16. (Original) The computer system of Claim 15 wherein said method further comprises:

determining a state of said cache line, wherein said state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state; and

rolling back said first group of instructions and then granting said access provided said cache line is in other than said shared state and said indicator indicates said cache line has been read by said first instruction, and otherwise granting said access without rolling back said first group of instructions.

17. (Original) The computer system of Claim 15 wherein said method further comprises maintaining a count of how many times said first group of instructions is rolled back.

18. (Original) The computer system of Claim 17 wherein said method further comprises eliminating one or more instructions from said first group of instructions if said count exceeds a threshold.

19-20. (Canceled).

21. (Original) The computer system of Claim 12 wherein said method further comprises:

 setting a bit associated with said cache line in response to said first instruction causing said cache line to be read; and
 clearing said bit when said execution of said first group of instructions is ended.

22. (Original) The computer system of Claim 12 wherein said method further comprises:

 executing a plurality of groups of instructions in parallel on said first processor;
 changing an indicator associated with each cache line in said cache memory that has been accessed because of an instruction in any of said plurality of groups of instructions; and
 provided all cache lines in said cache memory are indicated as having been accessed, rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory, and otherwise continuing said executing of said plurality of groups of instructions.

23. (Currently Amended) A method of managing shared memory in a computer system, said method comprising:

executing a first group of instructions on a first processor;

associating a first state with a cache line in a cache comprising a plurality of cache lines, wherein said first state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state; and

associating a second state with said cache line during said executing, wherein said second state indicates that said cache line has been accessed because of a first instruction of said first group of instructions and wherein said second state is cleared when execution of said first group of instructions is ended, wherein multiple groups of instructions are executable in parallel by said first processor and wherein said second state comprises a bit for each cache line in said plurality of cache lines for each group in said groups of instructions.

24. (Original) The method of Claim 23 wherein said first group of instructions includes multiple memory operations.

25. (Currently Amended) The method of Claim 23 further comprising:

executing a second group of instructions, wherein said second group of instructions includes a second instruction that when executed causes said cache line to be accessed and wherein said first group of instructions is associated with a thread executing on said [[a]] first processor and wherein said second group of instructions is associated with a thread executing on an agent external to said first processor;

in response to said second instruction seeking access to said cache line, using said second state to indicate said cache line has been accessed because of said first instruction; and

rolling back said first group of instructions before granting said access.

26. (Original) The method of Claim 25 wherein said rolling back is only performed provided said cache line is in other than said shared state.

27. (Original) The method of Claim 25 further comprising maintaining a count of how many times said first group of instructions is rolled back.

28. (Original) The method of Claim 27 further comprising eliminating one or more instructions from said first group of instructions if said count exceeds a threshold.

29. (Original) The method of Claim 25 wherein said agent is a second processor or an input/output agent capable of direct memory access (DMA).

30. (Original) The method of Claim 23 wherein said second state is indicated by:

setting a bit associated with said cache line in response to said first instruction causing said cache line to be read; and

clearing said bit when said execution of said first group of instructions is ended.

31. (Currently Amended) A computer system comprising:

a processor;

a cache memory coupled to said processor and having a plurality of cache lines; and

a memory unit coupled to said processor, wherein said memory unit has a plurality of entries and a plurality of an indicator bits bit per entry, wherein an said indicator bit of said plurality of indicator bits is set to indicate that a cache line of said cache memory has been accessed because of an instruction of a group of instructions and wherein said indicator bit is cleared when execution of said group of instructions is ended, and wherein each of said entries covers the address space of said cache memory such that a cache line maps uniquely to an entry but an entry maps to multiple cache lines, wherein multiple groups of instructions are executable in parallel by said processor and wherein said plurality of indicator bits comprises an indicator bit for each group in said groups of instructions for each cache line in said plurality of cache lines.

32. (Original) The computer system of Claim 31 wherein a subset of address bits of said cache line is used to identify said entry, wherein in response to said cache line being accessed, an indicator bit of said entry corresponding to said subset of address bits is set.

REMARKS

Please cancel Claims 8-9 and 19-20 without prejudice. Claims 1-7, 10-18 and 21-32 are pending. Claims 1-2, 12, 23, 25 and 31 are amended. No new matter is added as a result of the claim amendments. Support for the claim amendments can be found at least on page 9, lines 17-23, of the instant application.

35 U.S.C. § 102 Rejections

According to the instant Office Action, Claims 1-7, 10, 12-18, 21 and 23-30 are rejected under 35 U.S.C. § 102(e) as being anticipated by Jacobson et al. ("Jacobson;" U.S. Patent No. 6,938,130). The Applicants have reviewed the cited reference and respectfully submit that the present invention as recited in Claims 1-7, 10, 12-18, 21 and 23-30 is not shown or suggested by Jacobson.

Applicants respectfully submit that Jacobson does not show or suggest "changing an indicator associated with said first group of instructions to indicate that said cache line has been read, wherein said cache line is so indicated as having been read until execution of said first group of instructions is ended, wherein multiple groups of instructions are executable in parallel by said first processor and wherein said indicator comprises a bit per cache line for each group in said groups of instructions" as recited in independent Claim 1 (emphasis added) and as similarly recited in independent Claims 12 and 23. Jacobson appears to utilize only one "store-marking" bit per cache line, in contrast to the present claimed invention.

In summary, Applicants respectfully submit that Jacobson does not show or suggest the limitations of independent Claims 1, 12 and 23 and that these claims are allowable over the cited reference. As such, Applicants respectfully submit that Claims 2-7, 10, 13-18, 21 and 24-30 are also in condition for allowance as depending from allowable claims.

Consequently, Applicants respectfully submit that the basis for rejecting Claims 1-7, 10, 12-18, 21 and 23-30 under 35 U.S.C. § 102(e) is traversed.

35 U.S.C. § 103 Rejections

Claims 11 and 22

According to the instant Office Action, Claims 11 and 22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Jacobson in view of Arimilli et al. ("Arimilli;" U.S. Patent No. 6,263,407). The Applicants have reviewed the cited references and respectfully submit that the present invention as recited in Claims 11 and 22 is not shown or suggested by Jacobson and Arimilli, alone or in combination.

Claim 11 is dependent on Claim 1 and recites additional limitations, and Claim 22 is dependent on Claim 12 and recites additional limitations. Hence, by demonstrating that the cited references do not show or suggest the limitations of Claims 1 and 12, it is also demonstrated that the cited references do not show or suggest the limitations of Claims 11 and 22.

As presented above, Applicants respectfully submit that Jacobson does not show or suggest "changing an indicator associated with said first group of

instructions to indicate that said cache line has been read, wherein said cache line is so indicated as having been read until execution of said first group of instructions is ended, wherein multiple groups of instructions are executable in parallel by said first processor and wherein said indicator comprises a bit per cache line for each group in said groups of instructions" as recited in Claim 1 (emphasis added) and as similarly recited in Claim 12.

Applicants also submit that Arimilli does not overcome the shortcomings of Jacobson. More specifically, Applicants respectfully submit that Jacobson and Arimilli, alone or in combination, do not show or suggest an indicator bit per cache line per instruction group that can be executed in parallel by a processor.

In summary, Applicants respectfully submit that Jacobson and Arimilli, alone or in combination, do not show or suggest the limitations of independent Claims 1 and 12 and that these claims are allowable over the cited references. As such, Applicants respectfully submit that Claims 11 and 22 are also in condition for allowance as depending from allowable claims.

Consequently, Applicants respectfully submit that the basis for rejecting Claims 11 and 22 under 35 U.S.C. § 103(a) is traversed.

Claims 31-32

According to the instant Office Action, Claims 31-32 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Jacobson in view of Masri et al. ("Masri;" U.S. Patent No. 6,625,694). The Applicants have reviewed the cited references and respectfully submit that the present invention as recited in Claims

31-32 is not shown or suggested by Jacobson and Masri, alone or in combination.

Applicants respectfully submit that Jacobson does not show or suggest “a memory unit ..., wherein said memory unit has a plurality of entries and a plurality of indicator bits per entry, wherein an indicator bit of said plurality of indicator bits is set to indicate that a cache line of said cache memory has been accessed because of an instruction of a group of instructions ..., wherein multiple groups of instructions are executable in parallel by said processor and wherein said plurality of indicator bits comprises an indicator bit for each group in said groups of instructions for each cache line in said plurality of cache lines” as recited in independent Claim 31 (emphases added). Jacobson appears to utilize only one “store-marking” bit per cache line, in contrast to the present claimed invention.

Applicants also submit that Masri does not overcome the shortcomings of Jacobson. More specifically, Applicants respectfully submit that Jacobson and Masri, alone or in combination, do not show or suggest an indicator bit per entry (that maps to multiple cache lines) per instruction group that can be executed in parallel by a processor.

In summary, Applicants respectfully submit that Jacobson and Masri, alone or in combination, do not show or suggest the limitations of independent Claim 31 and that Claim 31 is allowable over the cited references. As such, Applicants respectfully submit that Claim 32 is also in condition for allowance as depending from an allowable claim.

Consequently, Applicants respectfully submit that the basis for rejecting Claims 31-32 under 35 U.S.C. § 103(a) is traversed.

Conclusions

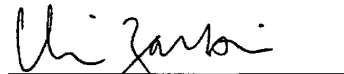
In light of the above remarks, reconsideration of the rejected claims is respectfully requested. Based on the arguments presented above, it is respectfully asserted that Claims 1-7, 10-18 and 21-32 overcome the rejections of record and, therefore, allowance of these claims is solicited.

Applicants have reviewed the references cited but not relied upon and did not find these references to show or suggest the present claimed invention: U.S. Patent Nos. 5,428,761, 5,701,432, 6,658,536, 5,974,438 and 6,006,299.

The Examiner is invited to contact Applicants' undersigned representative if the Examiner believes such action would expedite resolution of the present Application.

Respectfully submitted,
MURABITO HAO & BARNES LLP

Dated: 6/8/07



William A. Zarbis
Registration No. 46,120

Address: MURABITO HAO & BARNES LLP
Two North Market Street
Third Floor
San Jose, California 95113

Telephone: (408) 938-9060

PATENT APPLICATION FEE DETERMINATION RECORD
Effective December 8, 2004

11/102,127

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS		
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	minus 20=	*
INDEPENDENT CLAIMS	minus 3=	*
MULTIPLE DEPENDENT CLAIM PRESENT ☐		

SMALL ENTITY
TYPE ☐

OR
OTHER THAN
SMALL ENTITY

RATE	FEE
BASIC FEE	
X\$ 25=	
X100=	
+180=	
TOTAL	

RATE	FEE
BASIC FEE	
X\$50=	
X200=	
+360=	
TOTAL	

If the difference in column 1 is less than zero, enter "0" in column 2.

CLAIMS AS FILED - PART II

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	Minus **	=
	Independent	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

SMALL ENTITY

OR
OTHER THAN
SMALL ENTITY

RATE	ADDITIONAL FEE
X\$ 25=	
X100=	
+180=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$50=	
X200=	
+360=	
TOTAL ADDIT. FEE	

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	Minus **	=
	Independent	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

RATE	ADDITIONAL FEE
X\$ 25=	
X100=	
+180=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$50=	
X200=	
+360=	
TOTAL ADDIT. FEE	

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT C	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	Minus **	=
	Independent	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

RATE	ADDITIONAL FEE
X\$ 25=	
X100=	
+180=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$50=	
X200=	
+360=	
TOTAL ADDIT. 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.



UNITED STATES PATENT AND TRADEMARK OFFICE

un
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.
11/102,127	04/07/2005	Guillermo J. Rozas	TRAN-P084	6398

7590 03/08/2007
WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

EXAMINER

OTTO, ALAN

ART UNIT	PAPER NUMBER
----------	--------------

2109

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/08/2007	PAPER

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

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.		Applicant(s)		
	11/102,127		ROZAS, GUILLERMO J.		
	Examiner		Art Unit		
	Alan M. Otto		2109		

-- 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 07 April 2005.

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) 1-32 is/are pending in the application.

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

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) 1-32 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 07 April 2005 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

This Office Action is responsive to the Application filed 04/07/2005. Claims 1-32 are presented for examination and are currently pending.

Claim Rejections – 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-10, 12-21, and 23-30 are rejected under 35 U.S.C. 102(e) as being anticipated by Jacobson et al.'130 (U.S. Patent No. 6,938,130), herein referred to as Jacobson et al.'130.

3. Referring to claims 1 and 12, Jacobson et al.'130 discloses as claimed, a method of managing memory in a computer system (see Abstract), said method comprising: executing a first group of instructions (see col. 6, lines 42-46, regarding a critical section of code being executed), wherein said first group of instructions comprises multiple memory operations (see col. 7, lines 31-34, regarding loading and store marking cache lines in response to loads and stores, which constitute memory operations) and a first instruction that when executed causes a cache line of a cache memory to be read (see col. 7, lines 37-39, regarding a load operation, causing a cache line to be read); and in response to said first instruction causing said cache line to be read, changing an

indicator associated with said first group of instructions to indicate that said cache line has been read (see col. 5, lines 27-33, regarding a load-marking bit which is an indicator that is set when data from the cache line has been read), wherein said cache line is so indicated as having been read until execution of said first group of instructions is ended (see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache).

4. As to claims 2 and 13, Jacobson et al.'130 also discloses the method of Claim 1 further comprising: executing a second group of instructions (see col. 6, lines 47-50, regarding monitoring other threads for interfering data accesses), wherein said second group of instructions includes a second instruction that when executed causes said cache line to be accessed (see col. 7, lines 1-5, regarding a store by another thread on a load marked line), and wherein said first group of instructions is associated with a thread executing on a first processor (see fig. 1, processor 101, and col. 5, lines 19-30, regarding the first processor sending load and store operations which could be load marked or store marked) and said second group of instructions is associated with a thread executing on an agent external to said first processor (see fig. 1, processor 102, and col. 7, lines 8-13, regarding the interfering access thread coming from another processor accessing a cache line); in response to said second instruction seeking access to said cache line (see col. 7, lines 8-13, regarding determining if a cache line is being accessed by another processor), using a value of said indicator to determine whether said cache line has been read and generating a result thereof (see col. 7, lines 1-13, regarding the use of the load marked or store marked indicator to see if an

interfering data access has taken place); and handling said first group of instructions according to said result (see col. 6, lines 55-58, regarding re-executing the first group of instructions if an interfering data access has taken place).

5. As to claims 3 and 15, Jacobson et al.'130 also discloses the method of Claim 2 wherein said handling comprises rolling back said first group of instructions before granting said access (see col. 6, lines 55-58, regarding discarding the execution of the first group of instructions, and re-executing them if an interfering access is detected) provided said indicator indicates said cache line has been read by said first instruction (see col. 11, lines 54-59 regarding an interfering access, which means that the indicator indicated the cache line was read. Also see col. 7, lines 1-5), and otherwise granting said access without rolling back said first group of instructions (see col. 6, lines 47-54, regarding continually monitoring for interfering accesses. If none are detected, the first group of instructions is not rolled back, but is committed, and the second group of instructions are granted access, as they are not interfering).

6. As to claims 4 and 16, Jacobson et al.'130 also discloses the method of Claim 4 wherein said handling further comprises: determining a state of said cache line, (see col. 11, lines 8-16, and fig. 12, regarding state information for a cache line, and see col. 11, lines 44-50, regarding making decisions based on the state, which was determined) wherein said state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state (see col. 11, lines 8-16, and fig. 12, regarding state information for a cache line. State bits valid 1204, dirty 1206, and owned 1208 can show those states); and rolling back said first group of instructions

(see col. 11, lines 54-59, regarding rolling back and repeating the first instructions) and then granting said access provided said cache line is in other than said shared state (see col. 11, lines 48-50, regarding shared ownership being ok for reading) and said indicator indicates said cache line has been read by said first instruction (see col. 11, lines 54-59 regarding an interfering access, which means that the indicator indicated the cache line was read. Also see col. 7, lines 1-5), and otherwise granting said access without rolling back said first group of instructions (see col. 6, lines 47-54, regarding continually monitoring for interfering accesses. If none are detected, the first group of instructions is not rolled back, but is committed, and the second group of instructions are granted access, as they are not interfering).

7. As to claims 5 and 17, Jacobson et al.'130 also discloses the method of Claim 3 further comprising maintaining a count of how many times said first group of instructions has been rolled back (see col. 6, lines 59-61, regarding counting how many times the system tries to re-execute the first group of instructions).

8. As to claims 6 and 18, Jacobson et al.'130 also discloses the method of Claim 5 further comprising eliminating one or more instructions from said first group of instructions if said count exceeds a threshold (see col. 6, lines 58-57, regarding executing alternative code if the count has exceeded a threshold. This alternative code as stated could revert back to the conventional technique of trying to acquire a lock on a critical section. Thus, this would eliminate possible instructions from the new technique taught by Jacobson, such as monitoring if an interfering data access takes place, or committing the changes after a transactional execution).

9. As to claims 7 and 14, Jacobson et al.'130 also discloses the method of Claim 2 wherein said agent is a second processor or an input/output agent capable of direct memory access (DMA) (see fig. 1, processor 102, and col. 7, lines 8-13, regarding the interfering access thread coming from another processor accessing a cache line).

10. As to claims 8 and 19, Jacobson et al.'130 also discloses the method of Claim 2 wherein said indicator comprises one bit per instruction group for each group of instructions being executed at a same time by said first processor (see col. 1, lines 33-42, regarding an alternate mode of sharing resources via locks. An indicator would be a lock for a critical section of code. Also see col. 6, lines 61-67, regarding using this alternate locking mode).

11. As to claims 9 and 20, Jacobson et al.'130 also discloses the method of Claim 2 wherein said indicator comprises one bit per cache line per instruction group for each group of instructions being executed at a same time by said first processor (see fig. 1, load marking bits 116, also see col. 6, lines 43-50, and col. 7, lines 1-5, regarding a bit being used for load marking for one thread being executed at one time by a processor).

12. As to claims 10 and 21, Jacobson et al.'130 also discloses the method of Claim 10 further comprising: setting a bit associated with said cache line in response to said first instruction causing said cache line to be read (see col. 5, lines 27-33, regarding setting a load marking bit when a cache line is loaded); and clearing said bit when said execution of said first group of instructions is ended (see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache).

13. Referring to claim 23, Jacobson et al.'130 discloses as claimed, a method of managing shared memory in a computer system (see Abstract), said method comprising: executing a first group of instructions (see col. 6, lines 42-46, regarding a critical section of code being executed); associating a first state with a cache line (see col. 11, lines 8-16, and fig. 12, regarding state information associated with a cache line), wherein said first state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state (see col. 11, lines 8-16, and fig. 12, regarding state information for a cache line. State bits valid 1204, dirty 1206, and owned 1208 can show those states); and associating a second state with said cache line during said executing (see fig. 12, store mark 1210, regarding showing a state where cache line has been written to. Also see col. 5, lines 34-37, regarding a load marking bit not shown in fig. 12), wherein said second state indicates that said cache line has been accessed because of a first instruction of said first group of instructions (see col. 5, lines 27-33, regarding a load-marking bit which is an indicator that is set when data from the cache line has been read) and wherein said second state is cleared when execution of said first group of instructions is ended (see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache).

14. As to claim 24, Jacobson et al.'130 also discloses the method of Claim 23 wherein said first group of instructions includes multiple memory operations. (see col. 7, lines 31-34, regarding loading and store marking cache lines in response to loads and stores, which constitute memory operations)

15. As to claim 25, Jacobson et al.'130 also discloses the method of Claim 23 further comprising: executing a second group of instructions (see col. 6, lines 47-50, regarding monitoring other threads for interfering data accesses), wherein said second group of instructions includes a second instruction that when executed causes said cache line to be accessed (see col. 7, lines 1-5, regarding a store by another thread on a load marked line) and wherein said first group of instructions is associated with a thread executing on a first processor (see fig. 1, processor 101, and col. 5, lines 19-30, regarding the first processor sending load and store operations which could be load marked or store marked) and wherein said second group of instructions is associated with a thread executing on an agent external to said first processor (see fig. 1, processor 102, and col. 7, lines 8-13, regarding the interfering access thread coming from another processor accessing a cache line); in response to said second instruction seeking access to said cache line (see col. 7, lines 8-13, regarding determining if a cache line is being accessed by another processor), using said second state to indicate said cache line has been accessed because of said first instruction (see col. 7, lines 1-13, regarding the use of the load marked or store marked indicator to see if an interfering data access has taken place); and rolling back said first group of instructions before granting said access (see col. 6, lines 55-58, regarding re-executing the first group of instructions if an interfering data access has taken place).

16. As to claim 26, Jacobson et al.'130 also discloses the method of Claim 25 wherein said rolling back is only performed provided said cache line is in other than said

shared state (see col. 11, lines 48-50, regarding shared ownership being ok for reading, and therefore the rolling back is not necessary if the cache line is in the shared state).

17. As to claim 27, Jacobson et al.'130 also discloses the method of Claim 25 further comprising maintaining a count of how many times said first group of instructions is rolled back (see col. 6, lines 59-61, regarding counting how many times the system tries to re-execute the first group of instructions).

18. As to claim 28, Jacobson et al.'130 also discloses the method of Claim 27 further comprising eliminating one or more instructions from said first group of instructions if said count exceeds a threshold (see col. 6, lines 58-57, regarding executing alternative code if the count has exceeded a threshold. This alternative code as stated could revert back to the conventional technique of trying to acquire a lock on a critical section. Thus, this would eliminate possible instructions from the new technique taught by Jacobson, such as monitoring if an interfering data access takes place, or committing the changes after a transactional execution).

19. As to claim 29, Jacobson et al.'130 also discloses the method of Claim 25 wherein said agent is a second processor or an input/output agent capable of direct memory access (DMA) (see fig. 1, processor 102, and col. 7, lines 8-13, regarding the interfering access thread coming from another processor accessing a cache line).

20. As to claim 30, Jacobson et al.'130 also discloses the method of Claim 23 wherein said second state is indicated by: setting a bit associated with said cache line in response to said first instruction causing said cache line to be read (see col. 5, lines 27-33, regarding setting a load marking bit when a cache line is loaded); and clearing said

bit when said execution of said first group of instructions is ended (see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache).

Claim Rejections - 35 USC § 103

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

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

22. The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

23. Claims 11 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jacobson et al.'130 in view of Arimilli et al.'407 (U.S. Patent No. 6,263,407), herein referred to as Arimilli et al.'407.

24. As to claims 11 and 22, Jacobson et al.'130 discloses as claimed, the method of Claim 1 further comprising: executing a plurality of groups of instructions in parallel (see Abstract); changing an indicator associated with each cache line in said cache that has been accessed because of an instruction in any of said plurality of groups of instructions

(see col. 3, lines 39-46, regarding changing a load mark indicator associated with each cache line when a cache line has been accessed from a plurality of groups of instructions);

Jacobson et al.'130 discloses the claimed invention except for where all cache lines in said cache memory are indicated as having been accessed, rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory, and otherwise continuing said executing of said plurality of groups of instructions.

However, Arimilli et al.'407 discloses where all cache lines in said cache memory are indicated as having been accessed (see col. 9, lines 2-6, regarding the write queue of the cache being full, hence all cache lines have been accessed), rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory (see col. 9, lines 2-6, regarding a transaction being retried. Because the transaction was headed for the write queue, it is a write transaction that was attempting to write or replace information in a cache line), and otherwise continuing said executing of said plurality of groups of instructions (see col. 9, lines 2-6, where the transaction would take place if the write queue were not busy, which would take place if the cache lines had not been accessed, and were available).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Jacobson et al.'130 to comprise where all cache lines in said cache memory are indicated as having been accessed, rolling back a group of

instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory, and otherwise continuing said executing of said plurality of groups of instructions, as taught by Arimilli et al.'407, in order to provide for more predictable software behavior and consistent software timing (see col. 8, lines 56-64, regarding advantages of operating in precise mode, where retries can happen).

25. Claims 31 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jacobson et al.'130 in view of Masri et al.'694 (U.S. Patent No. 6,625,694), herein referred to as Masri et al.'694.

26. Referring to claim 31, Jacobson et al.'130 discloses as claimed, a computer system comprising: a processor (see fig. 1, processor 101); a cache memory coupled to said processor and having a plurality of cache lines (see fig. 1, cache unit 115); and a memory unit coupled to said processor (see fig. 1, cache memory 115, where the cache memory can constitute a memory unit), wherein said memory unit has a plurality of entries and an indicator bit per entry (see fig. 1, load marking bits 116 which are indicator bits), wherein said indicator bit is set to indicate that a cache line of said cache memory has been accessed because of an instruction of a group of instructions (see col. 5, lines 27-33, regarding setting a load marking bit when a cache line is loaded. Store marking bits can be set to signify a store. Also see col. 5, lines 56-58) and wherein said indicator bit is cleared when execution of said group of instructions is ended (see fig. 7, step 704, regarding once execution has been finished, committing changes, and during step 704, clearing load marks from the cache).

Jacobson et al.'130 discloses the claimed invention except for where each of said entries covers the address space of said cache memory such that a cache line maps uniquely to an entry but an entry maps to multiple cache lines.

However, Masri et al.'694 discloses where each of said entries covers the address space of said cache memory such that a cache line maps uniquely to an entry but an entry maps to multiple cache lines (see col. 4, lines 49-58, regarding a set associative cache, where one directory entry corresponds to multiple cache lines).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Jacobson et al.'130 to comprise where each of said entries covers the address space of said cache memory such that a cache line maps uniquely to an entry but an entry maps to multiple cache lines, as taught by Masri et al.'694, in order to provide a replacement algorithm that can improve the cache-hit rate (see col. 2, lines 5-19, discussing advantages of a set associative cache. In addition, using a set associative cache is well known in the art.)

27. As to claim 32, Jacobson et al.'130 discloses the claimed invention except for the computer system of Claim 31 wherein a subset of address bits of said cache line is used to identify said entry, wherein in response to said cache line being accessed, an indicator bit of said entry corresponding to said subset of address bits is set.

However, Masri et al.'694 discloses where a subset of address bits of said cache line is used to identify said entry (see col. 5, lines 1-7, regarding each memory line being identified by a location offset), wherein in response to said cache line being

accessed, an indicator bit of said entry corresponding to said subset of address bits is set (see col. 5, lines 15-20, where state field 326 is an indicator bit corresponding).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Jacobson et al.'130 to comprise where a subset of address bits of said cache line is used to identify said entry, wherein in response to said cache line being accessed, an indicator bit of said entry corresponding to said subset of address bits is set, as taught by Masri et al.'694, in order provide a replacement algorithm that can improve the cache-hit rate (see col. 2, lines 5-19, discussing advantages of a set associative cache. In addition, a well-known advantage of using a set associative cache is that index bits don't have to include the part of main memory which is specified by the cache, thus allowing for fewer bits.)

Conclusion

28. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Wang et al. discloses an apparatus for handling lock conditions, where a first instruction executed by a first processor processes data that is common to a second processor while the second processor is locked from simultaneously executing a second instruction that also processes that same data. A lock bit is set during the execution.

Neufeld discloses a computer system where each process thread is assigned to a register for each level of cache, and the number is associated with the number of cache lines in the memory that are being used.

Arimilli et al.'536 discloses a method of extending a cache, by expanding on the MESI cache coherency protocol, by adding a cache entry corresponding to a most recently accessed state.

Wong et al. discloses a multi threaded processing system that has a cache commonly accessible to each thread. The cache has a plurality of entries, each entry identified by a number. The lookup entry number can obtain a lock on the entry.

Herlihy et al. discloses a computer system which provides transactional memory operations, in which after an operation has been completed by the CPU, it attempts to commit the transaction, and this includes checking the status bit; if the bit has been toggled, the transaction aborts, the data item invalidated in local memory. Otherwise the transaction is committed.

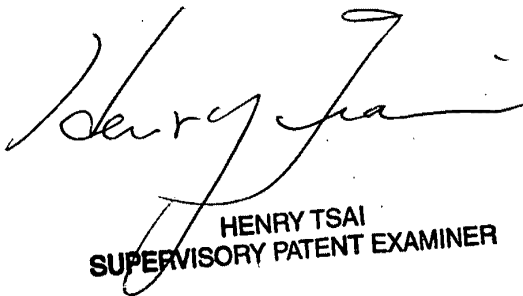
Contact Information

29. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Alan Otto whose telephone number is 571-270-1626. The examiner can normally be reached on 7:30-5:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Henry Tsai can be reached on 571-272-4176. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

30. 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). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AOTTO


HENRY TSAI
SUPERVISORY PATENT EXAMINER
3/3/07

Notice of References Cited	Application/Control No. 11/102,127	Applicant(s)/Patent Under Reexamination ROZAS, GUILLERMO J.	
	Examiner Alan M. Otto	Art Unit 2109	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,938,130	08-2005	Jacobson et al.	711/144
*	B	US-6,263,407	07-2001	Arimilli et al.	711/144
*	C	US-6,625,694	09-2003	Masri et al.	711/133
*	D	US-5,428,761	06-1995	Herlihy et al.	711/130
*	E	US-5,701,432	12-1997	Wong et al.	711/130
*	F	US-6,658,536	12-2003	Arimilli et al.	711/141
*	G	US-5,974,438	10-1999	Neufeld, E. David	718/104
*	H	US-6,006,299	12-1999	Wang et al.	710/108
	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. 11102127	Applicant(s)/Patent Under Reexamination ROZAS, GUILLERMO J.
	Examiner Otto, Alan M	Art Unit 2109

✓	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	02/26/2007										
	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	✓										
	31	✓										
	32	✓										

Search Notes 	Application/Control No. 11102127	Applicant(s)/Patent Under Reexamination ROZAS, GUILLERMO J.
	Examiner Otto, Alan M	Art Unit 2109

SEARCHED			
Class	Subclass	Date	Examiner
711	118, 130, 133, 134, 141, 144, 145, 146	2/26/07	AO

SEARCH NOTES		
Search Notes	Date	Examiner
East Search	2/26/07	AO
NPL Search	2/26/07	AO

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

2/26/2007

18:25:03 PM



UNITED STATES PATENT AND TRADEMARK OFFICE

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



Bib Data Sheet

CONFIRMATION NO. 6398

SERIAL NUMBER 11/102,127	FILING OR 371(c) DATE 04/07/2005 RULE	CLASS 711	GROUP ART UNIT 2109	ATTORNEY DOCKET NO. TRAN-P084
APPLICANTS Guillermo J. Rozas, Los Gatos, CA;				
** CONTINUING DATA ***** Ao				
** FOREIGN APPLICATIONS ***** Ao				
IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** 05/09/2005				
Foreign Priority claimed ☐ yes ☒ no 35 USC 119 (a-d) conditions ☐ yes ☒ no ☐ Met after met Verified and Allowance <i>[Signature]</i> 2-26-07 Acknowledged Examiner's Signature Initials		STATE OR COUNTRY CA	SHEETS DRAWING 5	TOTAL CLAIMS 32
INDEPENDENT CLAIMS 4				
ADDRESS WAGNER, MURABITO & HAO LLP Third Floor Two North Market Street San Jose, CA95113				
TITLE Memory management methods and systems that support cache consistency				
FILING FEE RECEIVED 1800	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:		☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit	

Dial · g DataStar

[options](#)[logout](#)[feedback](#)[help](#)[databases](#)[easy
search](#)**Advanced Search:**

Inspec - 1898 to date (INZZ)

[limit](#)

Search history:

No.	Database	Search term	Info added since	Results	
CP		[Clipboard]		0	-
1	INZZ	cache ADJ line	unrestricted	267	show titles
2	INZZ	unavailable OR full OR busy	unrestricted	213993	show titles
4	INZZ	1 NEAR 2	unrestricted	0	-
5	INZZ	roll-back OR rollback OR undo OR retry OR reissue OR redo OR abort	unrestricted	2734	show titles
6	INZZ	5 AND 1	unrestricted	2	show titles
7	INZZ	1 AND 2	unrestricted	13	show titles
8	INZZ	use ADJ bit	unrestricted	158	show titles
9	INZZ	8 AND 1	unrestricted	0	-
10	INZZ	cache	unrestricted	13519	show titles
11	INZZ	8 AND 10	unrestricted	1	show titles

[hide](#) | [delete all search steps...](#) | [delete individual search steps...](#)Enter your search term(s): [Search tips](#) ☐ Thesaurus mapping Information added since: or: 
(YYYYMMDD)[search](#)☐ Documents with images

Select special search terms from the following list(s):

- ☒ Publication year 1950-
- ☒ Publication year 1898-1949
- ☒ Inspec thesaurus - browse headings 
- ☒ Inspec thesaurus - enter a term 

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	4604223	thread or instruction or code or program	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:20
L2	10919066	break\$4 or reduc\$4 or eliminat\$4 or segment\$4 or divid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:20
L3	204801	L1 near5 L2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:22
L5	1178	fail\$3 near10 I3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:24
L6	581	I5 and memory	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:24
L7	137	I5 and cache	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:35
L8	133508	subdivid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:35
L10	1850	I8 near5 I1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:37

EAST Search History

L11	4	l10 near15 fail\$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:50
L12	363	l6 and processor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 17:01
L13	121	l12 and cache	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 17:15
L14	2	"20050268049"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 17:47
L15	167624	(memory or cache) adj3 (access or accesses or request)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 17:48
L16	9	l15 near10 subdivide	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 17:48
S1	8	rozas-guillermo.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 09:02
S2	33514	"711".clas.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 14:13

EAST Search History

S3	2124	711/118.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 14:27
S4	10797	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 16:28
S5	420	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 16:29
S6	159	S4 and S5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 16:48
S7	2799	S4 near5 read\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 16:48
S8	56	S7 and S5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 16:49
S9	7	dma and S8	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 17:31
S10	171	thread near15 S4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 17:31

EAST Search History

S11	57	S10 and S7	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 17:32
S12	42	S11 and S2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 17:32
S13	5	S11 and S3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 18:03
S14	52196	roll adj back or reissue or redo	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 18:05
S15	1649	S14 near7 (instruction or thread or process)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:06
S16	140	S15 and cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 18:19
S17	3	S4 near50 S15	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/20 18:19
S18	52196	roll adj back or reissue or redo	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:54

EAST Search History

S19	1067	S18 near3 (instruction or thread or process)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:54
S20	10797	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:07
S21	99	S19 and S20	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:07
S22	2124	711/118.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:07
S23	420	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:07
S24	2	S21 and (S23 or S22)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:07
S25	75313	roll adj back or reissue or redo or retry	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:54
S26	3409	S25 near3 (instruction or thread or process)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:55

EAST Search History

S27	282	S26 and S20	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:55
S28	32	S27 and (S23 or S22)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 10:32
S29	3	S26 near10 S20	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:57
S30	7	S26 near20 S20	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 09:57
S31	1777	shared adj cache	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 10:32
S32	76	S31 and S27	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 10:40
S33	357	cache adj line adj tag	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 10:40
S35	40	S33 near10 (read or access)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 10:45

EAST Search History

S36	2	S33 near10 (thread)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 10:47
S37	43	S33 near10 (thread or instruction)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:21
S38	1330	711/141.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:21
S39	1159	711/144.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:21
S40	862	711/145.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:21
S41	833	711/146.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:21
S42	1284	710/22.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:22
S43	2941	S38 or S39 or S40 or S41	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:22

EAST Search History

S44	420	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:22
S45	147	S44 and S43	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:10
S46	39	S42 and S43	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:23
S47	7975278	task or process or program or thread or instruction or command	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:47
S48	98542	roll-back or rollback or undo or retry or reissue or redo or abort	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:58
S49	13552	S48 near5 S47	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 15:59
S50	13	S49 and S45	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 16:37
S51	2	"5701432".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 16:37

EAST Search History

S52	10797	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 17:05
S53	4679	S52 near20 S47	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 17:06
S54	45	S53 and S45	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/21 17:06
S55	1335	711/141.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:10
S56	1159	711/144.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:10
S57	863	711/145.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:10
S58	833	711/146.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:10
S59	2947	S55 or S56 or S57 or S58	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:10

EAST Search History

S60	420	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:10
S61	147	S60 and S59	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:10
S62	10816	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:10
S63	76	S62 and S61	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:11
S64	7979343	task or process or program or thread or instruction or command	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:11
S65	4691	S62 near20 S64	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:11
S66	45	S65 and S61	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 09:11
S67	31	S63 not S66	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 11:13

EAST Search History

S68	7	10/637168	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/22 11:13
S69	1085818	replace or replacement	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 09:16
S70	10816	cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 09:16
S71	997	S69 near5 S70	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 09:18
S72	98645	roll-back or rollback or undo or retry or reissue or redo or abort	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 09:17
S73	3	S72 near40 S71	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 09:18
S74	3872	cache near4 full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 09:19
S75	34	S74 near40 S72	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 10:47

EAST Search History

S76	2	"6006299".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:11
S77	3700	count\$3 near8 S72	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:17
S78	1335	711/141.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:14
S79	1159	711/144.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:14
S80	863	711/145.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:14
S81	833	711/146.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:14
S82	2947	S78 or S79 or S80 or S81	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:14
S83	420	711/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:14

EAST Search History

S84	147	S83 and S82	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:14
S85	4	S77 and S84	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:14
S86	12850	count\$3 near8 fail\$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:18
S87	2	S86 and S84	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:18
S88	6	cache near3 full near6 delay	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:42
S89	12	cache near3 full near6 retry	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:42
S90	0	S84 and cache near3 full near6 S72	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:42
S91	22	S84 and cache near3 full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:44

EAST Search History

S92	26	S84 and cache near3 wait	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:47
S93	128	711/133.ccls. and 711/134.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:48
S94	10	S93 and S72	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 11:48
S95	2	"6625694".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 15:01
S96	49	S93 and cache adj line	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 15:01
S97	21	S96 and full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 15:23
S98	2701	cache adj4 full	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 15:25
S99	12	S98 near20 S72	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:13

EAST Search History

S10 0	10919066	break\$4 or reduc\$4 or eliminat\$4 or segment\$4 or divid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:14
S10 1	4604223	thread or instruction or code or program	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:18
S10 2	10919066	break\$4 or reduc\$4 or eliminat\$4 or segment\$4 or divid\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/26 16:16

17236 U.S. PTO
040705

Attorney Docket No.: TRAN-P084

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

I hereby certify that this transmittal of the below described documents is being deposited with the United States Postal Service in an envelope bearing Express Mail Postage and an Express Mail label, with the below serial number, addressed to the Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450, on the below date of deposit.			
Express Mail Label No.:	EV561340528US	Name of Person Making the Deposit:	Anthony Chou
Date of Deposit:	04/07/05	Signature of the Person Making the Deposit:	<i>Anthony Chou</i>

112948 U.S. PTO
11/102127

040705

Inventor(s): Guillermo J. Rozas

Title: MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY

Mail Stop Patent Application
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
Sir:

Transmittal of a Patent Application
(Under 37 CFR §1.53)

Transmitted herewith is the above identified patent application, including:

- ☒ Specification, claims and abstract, totaling 30 pages.
- ☐ Formal drawings, totaling _____ pages.
- ☒ Informal drawings, totaling 5 pages.
- ☒ Declaration and Power of Attorney.
- ☐ Information Disclosure statement.
- ☐ Form 1449
- ☒ Assignment(s)
- ☒ Assignment Recordation Form (duplicate)
- ☒ Request and Certification under 35 U.S.C. 122(b)(2)(B)(i)

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

Application Type	FILING FEES		SEARCH FEES		EXAMINATION FEES		Fees Paid (\$)
	Fee (\$)	Small Entity	Fee (\$)	Small Entity	Fee (\$)	Small Entity	
Utility	300	150	500	250	200	100	\$1,000.00
Design	200	100	100	50	130	65	\$
Plant	200	100	300	150	160	80	\$
Reissue	300	150	500	250	600	300	\$
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description

Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent
Each independent claim over 3 or, for reissues, each independent claim more than in the original patent
Multiple dependent claims

Small Entity	
Fee (\$)	Fee (\$)
50	25
200	100
360	180

Total Claims	Extra Claims	Fee (\$)	Fee Paid (\$)	Multiple Dependent Claims
32 -20 or HP=	12	x \$50.00	\$600.00	Fee (\$)
HP= highest number of total claims paid for, if greater than 20				
Indep. Claims	Extra Claims	Fee (\$)	Fee Paid (\$)	
4 -3 or HP=	1	x \$200	\$200.00	
HP= highest number of independent claims paid for, if greater than 3				

3. APPLICATION SIZE FEE

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 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets						
Total Sheets		Extra Sheets		Number of each additional 50 or fraction thereof	Fee \$	Fee Paid \$
36	-100	0	/50		=	\$0.00

4. OTHER FEE(S)

Non-English specification, \$130 fee (no small entity discount)
Other: Assignment Recordation Fee: \$40.00

Total Fees Due (\$)

\$1,840.00

5. PAYMENT OF FEES

The full fee due in connection with this communication is provided as follows:

1. Not enclosed
 - ☐ No filing fee is to be paid at this time.
2. Enclosed
 - ☒ Filing fee
 - ☒ Recording assignment
 - ☒ The Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No.: 23-0085. A duplicate copy of this authorization is enclosed.
 - ☒ A check in the amount of \$1,840.00
 - ☐ Charge any fees required or credit any overpayments associated with this filing to Deposit Account No.: 23-0085.

This application is filed pursuant to 37 C.F.R. § 1.53 in the name of the above-identified Inventor(s).

[X] Customer No: 45590

Please direct all correspondence concerning the above-identified application to the following address:

WAGNER, MURABITO & HAO LLP
Two North Market Street, Third Floor
San Jose, California 95113
(408) 938-9060

[X] This transmittal ends with this page.

Respectfully submitted,

Date: 4/7/05

By: W.A. Zarpis
William A. Zarpis
Reg. No. 46,120

TRAN-P084

UNITED STATES PATENT APPLICATION

for

MEMORY MANAGEMENT METHODS AND SYSTEMS
THAT SUPPORT CACHE CONSISTENCY

Inventor:

GUILLERMO J. ROZAS

Prepared by:

WAGNER, MURABITO & HAO LLP

TWO NORTH MARKET STREET

THIRD FLOOR

SAN JOSE, CALIFORNIA 95113

(408) 938-9060

TRAN-P084/ACM/WAZ

MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY

BACKGROUND OF THE INVENTION

5 FIELD OF THE INVENTION

Embodiments of the present invention relate to computer system memory, in particular the management of cache memory.

RELATED ART

10 With direct memory access (DMA), an input/output (I/O) system can issue read requests and writes directly to main memory without passing through the central processing unit (CPU). However, if the I/O system uses DMA to write to main memory, and changes data cached previously by the CPU, then the CPU will not receive the new data unless the CPU fetches the data from main
15 memory. Also, for DMA reads, the CPU cache may contain more recent data than main memory, and so the I/O system will not receive the new data unless it reads the cache instead of main memory. Multiprocessor systems, particularly systems referred to as shared-memory simultaneous multiprocessor (SMP) architectures, have to deal with similar types of scenarios. The MESI (Modified,
20 Exclusive, Shared, Invalid) protocol is a popular cache consistency (coherency) protocol that addresses these issues. A modification of the MESI protocol is the MOESI (Modified, Owned, Exclusive, Shared, Invalid) protocol. These protocols are known in the art.

25 Constraints are also applied to memory operations to prevent processors and DMA systems from reordering memory operations at will. If, for example,

each processor could reorder memory operations for optimization, code sequences that work on a single processor would not work on multiprocessor systems. One type of constraint can be referred to as sequential consistency. With sequential consistency, the legal orders of memory operations are those
5 that are indistinguishable from strict interleaving of the operations from each thread of control. For example, for two threads, the operations of one thread can be interleaved with those of the other thread, but the order of operations within each thread is preserved.

10 There are different classes of high-level processor architectures with regard to operation reordering, optimization and speculation. One such class can be referred to as a lumped in-order architecture, and another such class as a lumped out-of-order architecture. With lumped in-order architectures, instructions are lumped into instruction groups that can be committed and rolled back
15 atomically. Different instruction groups are committed sequentially and in order, but within an instruction group, arbitrary reordering and optimization can occur. Full speculation is possible within an instruction group, but speculation, reordering and optimization across instruction groups is limited. With lumped out-of-order architectures, instructions are lumped into instruction groups that can
20 be committed and rolled back atomically. Different groups can execute out of order but are committed in order.

SUMMARY OF THE INVENTION

A method or system that can maintain sequential consistency for lumped (in-order or out-of-order) architectures would be advantageous. Prior attempts to achieve a sequentially consistent, lumped architecture are limited with respect to the number of memory operations that can be included in an instruction group. Specifically, such attempts are limited to a single memory operation per instruction group. Embodiments in accordance with the present invention overcome this disadvantage.

According to one embodiment of the present invention, a group of instructions is executed. The group of instructions can include multiple memory operations, and also includes an instruction that when executed causes a cache line to be accessed (e.g., read, write, store, load, etc.). In response to execution of that instruction, an indicator associated with the group of instructions is updated to indicate that the cache line has been accessed. The cache line is indicated as having been accessed until execution of the group of instructions is ended. If an external agent (e.g., another processor or a DMA system) snoops the cache and the cache line (or any other cache line in the cache, for that matter) is indicated as having been accessed, then the instruction group is rolled back and reissued. If an external agent (e.g., another processor or a DMA system) snoops the cache and no cache line is indicated as having been accessed, then the snoop can be processed using a conventional cache coherency protocol (MESI, for example).

25

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the present invention and, together with the description, serve to explain the principles of the invention. The
5 drawings referred to in this description should not be understood as being drawn to scale except if specifically noted.

Figure 1 is a block diagram of a portion of a computer system upon which
embodiments of the present invention can be implemented.

10

Figure 2 is a block diagram of a cache and showing cache tags that are used according to one embodiment of the present invention.

Figure 3 is a flowchart of a method for managing memory in accordance
15 with one embodiment of the present invention.

Figure 4 is a flowchart that continues from the flowchart of Figure 3.

Figure 5 is a flowchart of a method for replacing a cache line in
20 accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the various embodiments of the invention, examples of which are illustrated in the accompanying drawings.

While the invention will be described in conjunction with these embodiments, it
5 will be understood that they are not intended to limit the invention to these
embodiments. On the contrary, the invention is intended to cover alternatives,
modifications and equivalents, which may be included within the spirit and scope
of the invention as defined by the appended claims. Furthermore, in the following
detailed description of the present invention, numerous specific details are set
10 forth in order to provide a thorough understanding of the present invention.
However, it will be recognized by one of ordinary skill in the art that the present
invention may be practiced without these specific details. In other instances,
well-known methods, procedures, components, and circuits have not been
described in detail as not to unnecessarily obscure aspects of the present
15 invention.

Some portions of the detailed descriptions that follow are presented in
terms of procedures, logic blocks, processing, and other symbolic
representations of operations on data bits within a computer memory. These
20 descriptions and representations are the means used by those skilled in the data
processing arts to most effectively convey the substance of their work to others
skilled in the art. A procedure, logic block, process, etc., is here, and generally,
conceived to be a self-consistent sequence of steps or instructions leading to a
desired result. The steps are those requiring physical manipulations of physical
25 quantities. Usually, though not necessarily, these quantities take the form of
electrical or magnetic signals capable of being stored, transferred, combined,

compared, and otherwise manipulated in a computer system. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, bytes, values, elements, symbols, characters, terms, numbers, or the like.

5

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, discussions utilizing terms such as "executing," "changing," "using," "accessing," "handling," "rolling back," "maintaining," "eliminating," "setting," "clearing," "incrementing," "decrementing," "associating," "granting," "determining" or the like, refer to the action and processes (e.g., flowcharts 30, 40 and 50 of Figures 3, 4 and 5, respectively) of a computer system or similar intelligent electronic computing device (generally, a controller), that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

Figure 1 is a block diagram of a portion of a computer system 5 upon which embodiments of the present invention can be implemented. Computer system 5 can include elements in addition to those illustrated in Figure 1. Generally speaking, embodiments in accordance with the present invention pertain to a computer system (or similar electronic device) that includes a

processor, a main memory, a cache associated with the processor, and one or more external agents that can access the main memory and the processor's cache.

5 In the example of Figure 1, computer system 5 includes a first processor 10, an optional second processor 11, and other optional processors 12 that are coupled to main memory 18 via bus 19. In the present embodiment, computer system 5 also includes a peripheral device 13 coupled to main memory 18 via bus 19. For ease of discussion, the processors 10, 11 and 12 and the peripheral
10 device 13 may be collectively referred to herein as agents 10-13. Relative to processor 10, processors 11 and 12 and peripheral device 13 may be referred to as external agents; relative to processor 11, processors 10 and 12 and peripheral device 13 may be referred to as external agents; and so on.

15 In one embodiment, peripheral device 13 is capable of direct memory access (DMA). In general, DMA refers to a hardware circuit or software agent that can transfer data directly to or from main memory 18.

 Associated with the processors 10-12 are first cache 14, optional second
20 cache 15, and other optional caches 16, respectively. In one embodiment, each of the processors 10-12 has their own cache. The caches may be physically incorporated into the processors 10-12, or they may be physically external to the processors 10-12. In the present embodiment, each of the processors 10-12 can access each of the caches 14, 15 and 16 via bus 19.

25

In general, the agents 10-13 may each be executing a group of instructions according to a lumped in-order architecture or according to a lumped out-of-order architecture. The group of instructions can each include one or more memory operations (e.g., read, load, store, write, lock acquisition, etc.). The
5 group of instructions can be demarcated by commit points (e.g., a commit ending one group of instructions can mark the beginning of the next group of instructions, which in turn ends at another commit). An instruction group is handled atomically (e.g., it is treated as a unit). An instruction (or instructions) in the group of instructions being executed by one of the agents 10-13 may seek
10 access to a cache associated with any of the other agents 10-13.

In one embodiment, computer system 5 implements a cache coherency protocol such as, but not limited to, MESI (Modified, Exclusive, Shared, Invalid). According to the MESI protocol, a cache line is identified as being in one of the
15 modified, exclusive, shared or invalid states. In the modified state, the cache line includes more recent information than main memory 18 (only the information in the cache line is valid because main memory 18 is not up to date), and no other cache holds the information in the cache line. In the exclusive state, the information in the cache line is not in any other cache and the information in the
20 cache line is unmodified, and accordingly main memory 18 is up to date. In the shared state, the information in the cache line may be in one or more other caches, and main memory 18 is up to date. In the invalid state, the cache line does not contain valid information. MESI is consistent with sequential consistency.

25

Figure 2 is a block diagram of cache 14 according to one embodiment of the present invention. Cache 14 includes a number of cache lines exemplified by cache lines 21 and 22.

5 In the embodiment of Figure 2, a cache line tag is associated with each of the cache lines in cache 14. In one such embodiment, each cache line tag includes a number of cache coherency protocol bits. For example, cache coherency protocol bits 23 are associated with cache line 21, and cache coherency protocol bits 26 are associated with cache line 22. The cache
10 coherency protocol bits 23 and 26 can be used to indicate the state of the cache lines 21 and 22, respectively, according to a cache coherency protocol such as the MESI protocol.

 Furthermore, in the present embodiment, each cache line tag includes a
15 number of additional bits referred to herein as observed bits for ease of discussion. For example, observed bits 24, 25, ..., N1 are associated with cache line 21, and observed bits 27, 28, ..., N2 are associated with cache line 22. The observed bits for cache 14 are associated with groups of instructions executed by the agent (e.g., processor 10) that is associated with cache 14. In one
20 embodiment, each cache line tag is extended by one observed bit for each group of instructions that may be executed by processor 10 (Figure 1) at the same time. If, for example, processor 10 can execute five (5) instruction groups in parallel, then the cache line tags are each extended by 5 observed bits.

In the example of Figure 2, observed bits 24 and 27 (associated with cache line 21) are associated with instruction group A, and observed bits 25 and 28 (associated with cache line 22) are associated with instruction group B.

5 In one embodiment, an observed bit is set (e.g., set to a value of one) for a cache line in a cache if execution of an instruction in the group of instructions executed by the agent associated with that cache causes that cache line to be accessed (e.g., load, store, read, or write). For example, in one embodiment, observed bit 24 is set if an instruction in instruction group A, when executed by
10 processor 10, causes cache line 21 to be accessed, because cache line 21 is in cache 14 associated with processor 10 (Figure 1). In such an embodiment, observed bit 27 (also associated with instruction group A) is not set unless cache line 22 is also accessed because of instruction group A.

15 In general, with reference to Figures 1 and 2, a group of instructions executed by processor 10 may result in multiple cache lines in cache 14 being accessed. For each cache line in cache 14 accessed because of a group of instructions executed by processor 10, the observed bit corresponding to that group of instructions is set. However, an observed bit associated with a cache
20 line in cache 14 is not set if an instruction executed on an external agent 11-13 causes access (e.g., a snoop or interrogation, or perhaps a read) to that cache line. For example, in one embodiment, an instruction executed by an agent other than processor 10 that causes cache line 21 to be accessed will not set an observed bit for cache line 21.

25

In one embodiment, when execution of a group of instructions is ended (e.g., the instruction group is committed, rolled back, or aborted), each observed bit set by that group of instructions is cleared (e.g., unset, or set to a value of zero). For example, if instruction group A is committed (or retired), rolled back or
5 aborted, any of the observed bits 24 and 27 that are set are cleared. From another perspective, the observed bits associated with instruction group A, if set, remain set until the execution of instruction group A is ended.

To summarize, in one embodiment, when an agent (e.g., processor 10)
10 begins to execute a group of instructions, it allocates an observed bit for that group of instructions to each cache line in the cache associated with the agent (e.g., cache 14 associated with processor 10). If a cache line (e.g., cache line 21) has been accessed because of an instruction in an instruction group (e.g., instruction group A) being executed by processor 10, then the observed bit for
15 that cache line and corresponding to instruction group A (e.g., observed bit 24) is set. If another cache line (e.g., cache line 22) has also been accessed because of an instruction in instruction group A, then the observed bit for that cache line and corresponding to instruction group A (e.g., observed bit 27) is set. When execution of an instruction group is ended, the observed bits associated with that
20 instruction group are cleared, and thus the observed bits are available for subsequent instruction groups executed by the agent. For example, if instruction group A is committed, rolled back or aborted, then observed bits 24 and 27 are cleared.

25 In one embodiment, the observed bits are used in the following manner. Continuing with reference to Figures 1 and 2, if an external agent (e.g., another

processor 11 or 12 or a DMA agent such as peripheral device 13) executes an instruction that causes access to (or a request for access to) cache line 21, for example, and any of the observed bits 24, 25, ..., N1 have been set, then the instruction groups associated with the observed bits that are set are forced to roll
5 back and then reissue. Also, any of the observed bits 24, 25, ..., N1 associated with cache line 21 that have been set are cleared. For example, if observed bits 24 and 25 have been set, then instruction group A (which corresponds to observed bit 24) and instruction group B (which corresponds to observed bit 25) are rolled back, and observed bits 24 and 25 are cleared. The rollback also
10 clears the observed bits associated with other cache lines in cache 14 if those observed bits correspond to the groups of instructions that are rolled back. For example, if the instruction groups A and B are rolled back as mentioned above, then observed bits 27 and 28 are also cleared if those observed bits are set. The cache state transition occurs at this point, before the instruction groups A and B
15 are reissued and are able to again cause cache line 21 (and perhaps cache line 22) to be accessed.

In general, if an external agent 11-13 accesses, or requests access to, a cache line in cache 14, and an observed bit associated with that cache line has
20 been set, then all the observed bits that have been set for that cache line are cleared, all the instruction groups corresponding to the observed bits that are cleared are rolled back, and the observed bits associated with the instruction groups that are rolled back are also cleared for other cache lines in cache 14.

25 In one embodiment, the rollbacks just described occur only if an external agent 11-13 accesses, or requests access to, a cache line in cache 14, an

observed bit associated with that cache line has been set, and the cache line is not in the shared state. Again consider an example in which an external agent (e.g., another processor 11 or 12 or a DMA agent such as peripheral device 13) executes an instruction that seeks access to cache line 21. In one embodiment, 5 in addition to checking the observed bits 24, 25, ..., N1 associated with cache line 21, the cache coherency protocol bits 23 are also checked to determine the state (e.g., the MESI state) of cache line 21. In a MESI embodiment, for example, if cache line 21 is in the shared state, and if the access from the external agent is for sharing (e.g., it is not exclusive), then access to cache line 10 21 can be granted without rolling back the instruction groups associated with the observed bits 24, 25, ..., N1 that are set.

In one embodiment, observed bits that are set are not forgotten (e.g., cleared) except in the case of a rollback, commit or abort. In such an 15 embodiment, a cache line that has at least one of its observed bits set is not replaceable, although the cache line and its set observed bit(s) can be moved to another location. This can lead to a situation in which a cache line is needed, but no cache lines are available (that is, all cache lines have at least one of their respective bits set, so none of the cache lines can be replaced or moved). In one 20 embodiment, if such a situation arises, the group of instructions that needs the cache line, as well as following instruction groups, is rolled back and then reissued.

To prevent instruction groups from being repeatedly rolled back, in one 25 embodiment, the number of times each instruction group is rolled back is counted and compared to a threshold. In such an embodiment, if the number of times an

instruction group is rolled back exceeds the threshold, then the instruction group is rolled back for the last time. If it fails to execute to completion in the next attempt, the instruction group is reformed into a new instruction group that is reduced in size (e.g., one or more instructions is removed from the instruction group), and the counter for the new instruction group is set to zero. In the limiting case, the instruction group is eventually reduced to a single instruction that will be able to complete execution.

Figures 3 and 4 are flowcharts 30 and 40 of a method for managing memory in accordance with one embodiment of the present invention (the flowchart of Figure 4 is a continuation of the flowchart of Figure 3). Although specific steps are disclosed in flowcharts 30 and 40, such steps are exemplary. That is, embodiments of the present invention are well-suited to performing various other steps or variations of the steps recited in flowcharts 30 and 40. It is appreciated that the steps in flowcharts 30 and 40 may be performed in an order different than presented, and that not all of the steps in flowcharts 30 and 40 may be performed. Some or all of the processes and steps of flowcharts 30 and 40 can be realized, in one embodiment, as a series of instructions that reside in hardware or software (e.g., in main memory 18 of Figure 1) and are executed by one or more of the agents 10-13 in a computer system 5 (Figure 1).

In step 31 of Figure 3, a first group of instructions (e.g., instruction group A) is executed (e.g., by processor 10 of Figure 1). In one embodiment, the first group of instructions includes multiple memory operations (e.g., reads, writes, loads, stores, lock acquisition, etc.). At least one of the instructions in the first group of instructions, when executed, causes a cache line (e.g., cache line 21 of

Figure 2) in a cache associated with processor 10 (e.g., cache 14) to be accessed (e.g., read, written, etc.).

In step 32 of Figure 3, and continuing with reference to Figure 2, that
5 instruction is executed, causing cache line 21 to be accessed.

In step 33 of Figure 3, with reference also to Figure 2, an indicator associated with the first group of instructions is changed, in order to indicate that cache line 21 has been accessed because of the first group of instructions. In
10 one embodiment, the indicator is also associated with cache line 21. In one embodiment, a bit (e.g., observed bit 24) associated with cache line 21 and the first group of instructions is set. In one embodiment, cache line 21 is indicated as having been accessed (that is, for example, the observed bit remains set) until
15 execution of the first group of instructions is ended (e.g., because of a roll back, commit or abort).

In step 34 of Figure 3, with reference also to Figure 1, a second group of instructions is executed by an agent external to processor 10 (e.g., processor 11 or 12 or a DMA agent such as peripheral device 13). At least one of the
20 instructions in the second group of instructions, when executed, causes (or will cause) the cache line of interest (e.g., cache line 21 of Figure 2) to be accessed.

In step 35 of Figure 3, access to the cache line of interest (e.g., cache line 21 of Figure 2) is sought because of the second group of instructions. As will be
25 seen, the first group of instructions is then handled depending on whether or not cache line 21 has been accessed because of the first group of instructions and is

identified as having been accessed (e.g., by an observed bit). That is, it is possible that cache line 21 is accessed because of the second group of instructions before it is accessed because of the first group of instructions.

5 In step 41 of Figure 4, with reference also to Figure 2, a determination is made with regard to whether or not the cache line of interest (e.g., cache line 21) has been accessed because of the first group of instructions. In one embodiment, if the observed bit (e.g., observed bit 24) associated with cache line 21 and the first instruction group is set, then flowchart 40 proceeds to step 43,
10 and otherwise flowchart 44 proceeds to step 44. In another embodiment, if the observed bit (e.g., observed bit 24) associated with cache line 21 and the first instruction group is set, then flowchart 40 proceeds to step 42.

 In step 42 of Figure 4, and with reference to Figure 2, a determination is
15 made with regard to whether or not cache line 21 is in the shared state (e.g., according to the MESI protocol). If cache line 21 is in the shared state, then flowchart 40 proceeds to step 44, and otherwise flowchart 40 proceeds to step 43.

20 In step 43 of Figure 4, with reference also to Figure 2, the first group of instructions is atomically rolled back, and the observed bits 24, 25, ..., N1 associated with cache line 21 are cleared. Other observed bits, associated with other cache lines in cache 14 and corresponding to the first group of instructions, are also cleared if they are set. For example, if observed bit 27 (which also
25 corresponds to the first group of instructions) is set, then it is also cleared by the rollback of the first group of instructions. Also, other groups of instructions

associated with any of the observed bits of cache 14 that are cleared are also rolled back. For example, if observed bit 25 is set, then it is cleared and the instruction group associated with observed bit 25 (e.g., instruction group B) is rolled back, which in turn results in observed bit 28 (which also corresponds to
5 instruction group B) of cache line 22 being cleared.

In step 44 of Figure 4, access to cache line 21 (Figure 2) is granted.

In step 45 of Figure 4, in one embodiment, the number of times that the
10 first group of instructions has been rolled back is counted. If the number of times the first group of instructions has been rolled back exceeds a threshold, then one or more instructions are eliminated from the first instruction group. Specifically, in one embodiment, the first group of instructions is reissued, and if the next execution attempt results in another rollback, then the first instruction group is
15 reduced by one or more instructions.

Figure 5 is a flowchart 50 of a method for replacing a cache line in accordance with one embodiment of the present invention. Although specific steps are disclosed in flowchart 50, such steps are exemplary. That is,
20 embodiments of the present invention are well-suited to performing various other steps or variations of the steps recited in flowchart 50. It is appreciated that the steps in flowchart 50 may be performed in an order different than presented, and that not all of the steps in flowchart 50 may be performed. Some or all of the processes and steps of flowchart 50 can be realized, in one embodiment, as a
25 series of instructions that reside in hardware or software (e.g., in main memory

18 of Figure 1) and are executed by one or more of the agents 10-13 in a computer system 5 (Figure 1).

In step 51 of Figure 5, a number of instruction groups are executed in parallel using processor 10 (Figure 1), for example.

In step 52 of Figure 5, with reference also to Figures 1 and 2, each cache line in the cache associated with processor 10 (e.g., cache 14) that is accessed because of the groups of instructions being executed in step 51 is indicated as being accessed. For example, as previously described herein, an observed bit can be set for each cache line accessed because of the groups of instructions of step 51.

In step 53 of Figure 5, one of the instructions in one of the instruction groups attempts to replace a cache line.

In step 54, in one embodiment, if all of the cache lines in the cache of interest (e.g., cache 14) are indicated as having been accessed (e.g., at least one observed bit is set for each of the cache lines in cache 14, then the instruction group that has the instruction that attempted the cache line replacement (the instruction group of step 53) is rolled back.

In summary, methods and systems that can maintain sequential consistency for lumped (in-order or out-of-order) architectures are described. Significantly, embodiments in accordance with the present invention permit

sequentially consistent, lumped architectures in which multiple memory operations can be included in instruction groups.

Embodiments in accordance with the present invention can incorporate variations to the features described above. For systems with large speculation ability (e.g., potentially 20 or more active instruction groups), the cost (e.g., the storage cost) of the associated number of observed bits can be reduced in various ways. For example, a protocol can be established in which any external snoop causes all active instruction groups to roll back and reissue after the cache line state transition. In effect, such a protocol acts as if there is a single observed bit that is always set, except during the period of inactivity after rollback, and so no additional storage is needed for the observed bit. Alternatively, instead of an observed bit per cache line per instruction group, there can be an observed bit per instruction group.

15

Furthermore, new hardware structures can be added to track which cache lines have been accessed, without the storage of data. These hardware structures can be direct-mapped, set-associative, or fully-associative and of varying capacity.

20

Also, instead of having an observed bit per cache line per instruction group, a smaller memory structure (e.g., another cache, or a memory structure within main memory) that covers the whole cache address space and that has an observed bit per entry and per instruction group can be used. Each entry covers some portion of the cache address space, in such a way that every cache line address maps uniquely to an entry, but an entry maps to many cache lines. For

25

example, some subset of the address bits of a cache line can be used to identify an entry. The subset of the address bits are the address summary. When accessing a cache line, the entry corresponding to the address summary for the cache line is used to set the observed bits. Similarly, on a snoop request, the address is “summarized,” and the entry corresponding to the summary is examined for the presence of observed bits. The observed bits are cleared when an instruction group is committed, rolled back, or aborted. To illustrate, consider a cache complex where each cache holds 32 bytes, so that, for example, the bottom five bits (bits 4-0) of an address indicate bytes within the same cache line. Furthermore, consider a 32-entry observed bit structure, where bits 9-5 of the cache line address are used to select an entry, and this entry holds the observed bits. Then, any two addresses that have the same bits 9-5 will use the same entry in the summary structure; that is, they collide. This can mean that when a snoop comes, an instruction group that did not really conflict may be rolled back, because there is a conflict/collision in the summary structure even though there would not have been a conflict/collision in the cache. Thus, there is a tradeoff between the size of the summary structure and the overhead incurred by the “spurious” rollbacks due to collisions.

Embodiments in accordance with the present invention are thus described. While the present invention has been described in particular embodiments, it should be appreciated that the present invention should not be construed as limited by such embodiments, but rather construed according to the below claims.

CLAIMS

What is claimed is:

1. A method of managing memory in a computer system, said method
5 comprising:
 executing a first group of instructions, wherein said first group of
instructions comprises multiple memory operations and a first instruction that
when executed causes a cache line of a cache memory to be read; and
 in response to said first instruction causing said cache line to be read,
10 changing an indicator associated with said first group of instructions to indicate
that said cache line has been read, wherein said cache line is so indicated as
having been read until execution of said first group of instructions is ended.
2. The method of Claim 1 further comprising:
15 executing a second group of instructions, wherein said second group of
instructions includes a second instruction that when executed causes said cache
line to be accessed, and wherein said first group of instructions is associated with
a thread executing on a first processor and said second group of instructions is
associated with a thread executing on an agent external to said first processor;
20 in response to said second instruction seeking access to said cache line,
using a value of said indicator to determine whether said cache line has been
read and generating a result thereof; and
 handling said first group of instructions according to said result.
- 25 3. The method of Claim 2 wherein said handling comprises rolling
back said first group of instructions before granting said access provided said

indicator indicates said cache line has been read by said first instruction, and otherwise granting said access without rolling back said first group of instructions.

- 5 4. The method of Claim 3 wherein said handling further comprises:
 determining a state of said cache line, wherein said state is specified
 according to a cache coherency protocol comprising at least a modified state, a
 shared state and an invalid state; and
 rolling back said first group of instructions and then granting said access
10 provided said cache line is in other than said shared state and said indicator
 indicates said cache line has been read by said first instruction, and otherwise
 granting said access without rolling back said first group of instructions.
5. The method of Claim 3 further comprising maintaining a count of
15 how many times said first group of instructions has been rolled back.
6. The method of Claim 5 further comprising eliminating one or more
 instructions from said first group of instructions if said count exceeds a threshold.
- 20 7. The method of Claim 2 wherein said agent is a second processor or
 an input/output agent capable of direct memory access (DMA).
8. The method of Claim 2 wherein said indicator comprises one bit per
 instruction group for each group of instructions being executed at a same time by
25 said first processor.

9. The method of Claim 2 wherein said indicator comprises one bit per cache line per instruction group for each group of instructions being executed at a same time by said first processor.

5 10. The method of Claim 1 further comprising:
setting a bit associated with said cache line in response to said first instruction causing said cache line to be read; and
clearing said bit when said execution of said first group of instructions is ended.

10

11. The method of Claim 1 further comprising:
executing a plurality of groups of instructions in parallel;
changing an indicator associated with each cache line in said cache that has been accessed because of an instruction in any of said plurality of groups of
15 instructions; and

provided all cache lines in said cache memory are indicated as having been accessed, rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory, and otherwise continuing said executing of said plurality of groups of
20 instructions.

12. A computer system comprising:
a first processor;
a cache memory used by said first processor; and
a memory unit coupled to said first processor, said memory unit containing
5 instructions that when executed implement a memory management method, said
method comprising:
executing a first group of instructions using said first processor,
wherein said first group of instructions includes multiple memory
operations and a first instruction that when executed causes a cache line
10 of said cache memory to be read; and
in response to said first instruction causing said cache line to be
read, changing an indicator associated with said first group of instructions
to indicate that said cache line has been read, wherein said cache line is
so indicated as having been read until execution of said first group of
15 instructions is ended.

13. The computer system of Claim 12 wherein said method further
comprises:
executing a second group of instructions using an agent coupled to said
20 cache memory, wherein said second group of instructions includes a second
instruction that when executed causes said cache line to be accessed, and
wherein said agent is external to said first processor;
in response to said second instruction seeking access to said cache line,
using a value of said indicator to determine whether said cache line has been
25 read and generating a result thereof; and
handling said first group of instructions according to said result.

14. The computer system of Claim 13 wherein said agent is a second processor or an input/output agent capable of direct memory access (DMA).

5 15. The computer system of Claim 13 wherein said method further comprises rolling back said first group of instructions before granting said access provided said indicator indicates said cache line has been read by said first instruction, and otherwise granting said access without rolling back said first group of instructions.

10

16. The computer system of Claim 15 wherein said method further comprises:

 determining a state of said cache line, wherein said state is specified according to a cache coherency protocol comprising at least a modified state, a
15 shared state and an invalid state; and

 rolling back said first group of instructions and then granting said access provided said cache line is in other than said shared state and said indicator indicates said cache line has been read by said first instruction, and otherwise granting said access without rolling back said first group of instructions.

20

17. The computer system of Claim 15 wherein said method further comprises maintaining a count of how many times said first group of instructions is rolled back.

18. The computer system of Claim 17 wherein said method further comprises eliminating one or more instructions from said first group of instructions if said count exceeds a threshold.

5 19. The computer system of Claim 17 wherein said indicator comprises one bit per instruction group for each group of instructions being executed at a same time by said first processor.

20. The computer system of Claim 17 wherein said indicator comprises
10 one bit per cache line per instruction group for each group of instructions being executed at a same time by said first processor.

21. The computer system of Claim 12 wherein said method further comprises:
15 setting a bit associated with said cache line in response to said first instruction causing said cache line to be read; and
 clearing said bit when said execution of said first group of instructions is ended.

20 22. The computer system of Claim 12 wherein said method further comprises:
 executing a plurality of groups of instructions in parallel on said first processor;
 changing an indicator associated with each cache line in said cache
25 memory that has been accessed because of an instruction in any of said plurality of groups of instructions; and

provided all cache lines in said cache memory are indicated as having been accessed, rolling back a group of instructions that includes an instruction that when executed attempts to replace information in a cache line of said cache memory, and otherwise continuing said executing of said plurality of groups of
5 instructions.

23. A method of managing shared memory in a computer system, said method comprising:

executing a first group of instructions;
10 associating a first state with a cache line, wherein said first state is specified according to a cache coherency protocol comprising at least a modified state, a shared state and an invalid state; and
associating a second state with said cache line during said executing, wherein said second state indicates that said cache line has been accessed
15 because of a first instruction of said first group of instructions and wherein said second state is cleared when execution of said first group of instructions is ended.

24. The method of Claim 23 wherein said first group of instructions
20 includes multiple memory operations.

25. The method of Claim 23 further comprising:
executing a second group of instructions, wherein said second group of instructions includes a second instruction that when executed causes said cache
25 line to be accessed and wherein said first group of instructions is associated with a thread executing on a first processor and wherein said second group of

instructions is associated with a thread executing on an agent external to said first processor;

in response to said second instruction seeking access to said cache line, using said second state to indicate said cache line has been accessed because
5 of said first instruction; and
rolling back said first group of instructions before granting said access.

26. The method of Claim 25 wherein said rolling back is only performed provided said cache line is in other than said shared state.
10

27. The method of Claim 25 further comprising maintaining a count of how many times said first group of instructions is rolled back.

28. The method of Claim 27 further comprising eliminating one or more
15 instructions from said first group of instructions if said count exceeds a threshold.

29. The method of Claim 25 wherein said agent is a second processor or an input/output agent capable of direct memory access (DMA).

20 30. The method of Claim 23 wherein said second state is indicated by:
setting a bit associated with said cache line in response to said first instruction causing said cache line to be read; and
clearing said bit when said execution of said first group of instructions is ended.

25

31. A computer system comprising:
a processor;
a cache memory coupled to said processor and having a plurality of cache lines; and
5 a memory unit coupled to said processor, wherein said memory unit has a plurality of entries and an indicator bit per entry, wherein said indicator bit is set to indicate that a cache line of said cache memory has been accessed because of an instruction of a group of instructions and wherein said indicator bit is cleared when execution of said group of instructions is ended, and wherein each
10 of said entries covers the address space of said cache memory such that a cache line maps uniquely to an entry but an entry maps to multiple cache lines.

32. The computer system of Claim 31 wherein a subset of address bits of said cache line is used to identify said entry, wherein in response to said cache
15 line being accessed, an indicator bit of said entry corresponding to said subset of address bits is set.

MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY

ABSTRACT

- 5 Methods and systems for maintaining cache consistency are described. A group of instructions is executed. The group of instructions can include multiple memory operations, and also includes an instruction that when executed causes a cache line to be accessed. In response to execution of that instruction, an indicator associated with the group of instructions is updated to indicate that the
- 10 cache line has been accessed. The cache line is indicated as having been accessed until execution of the group of instructions is ended.

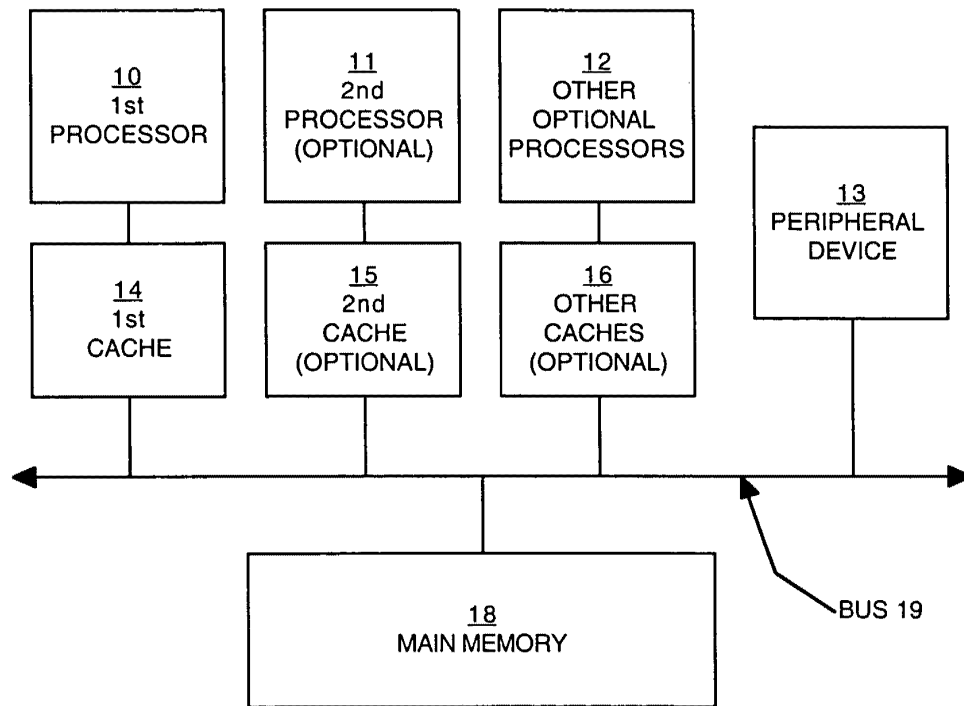


Figure 1

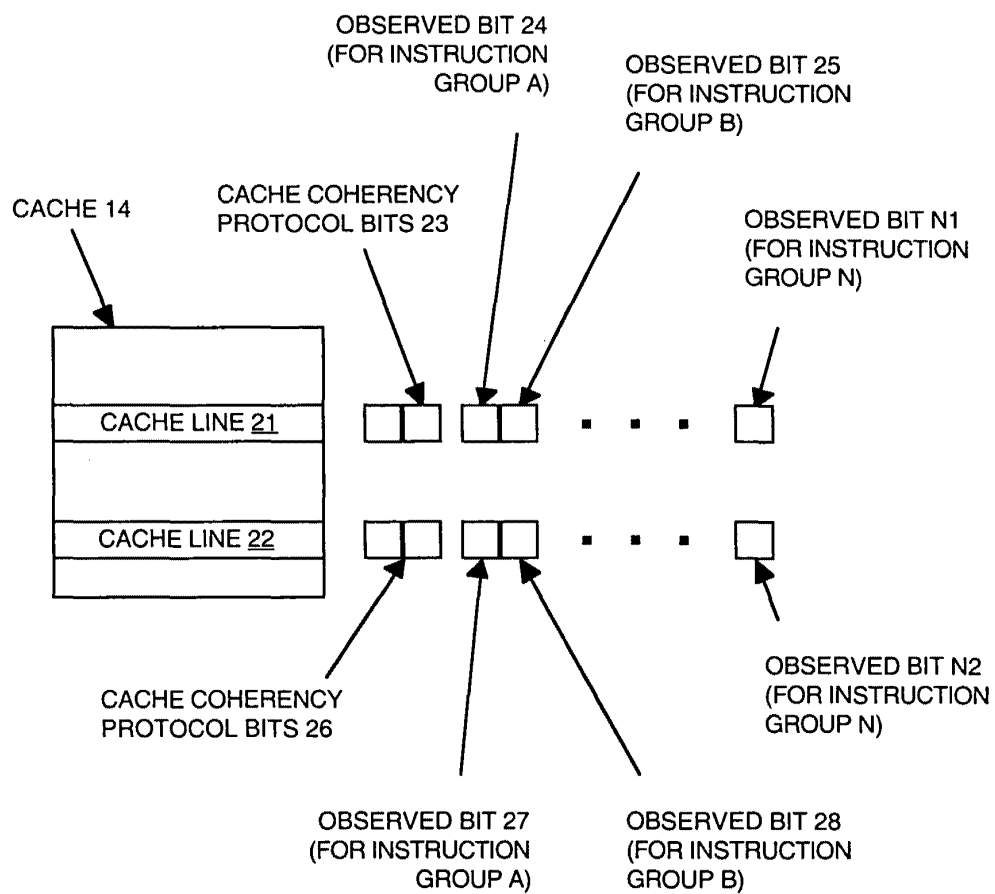


Figure 2

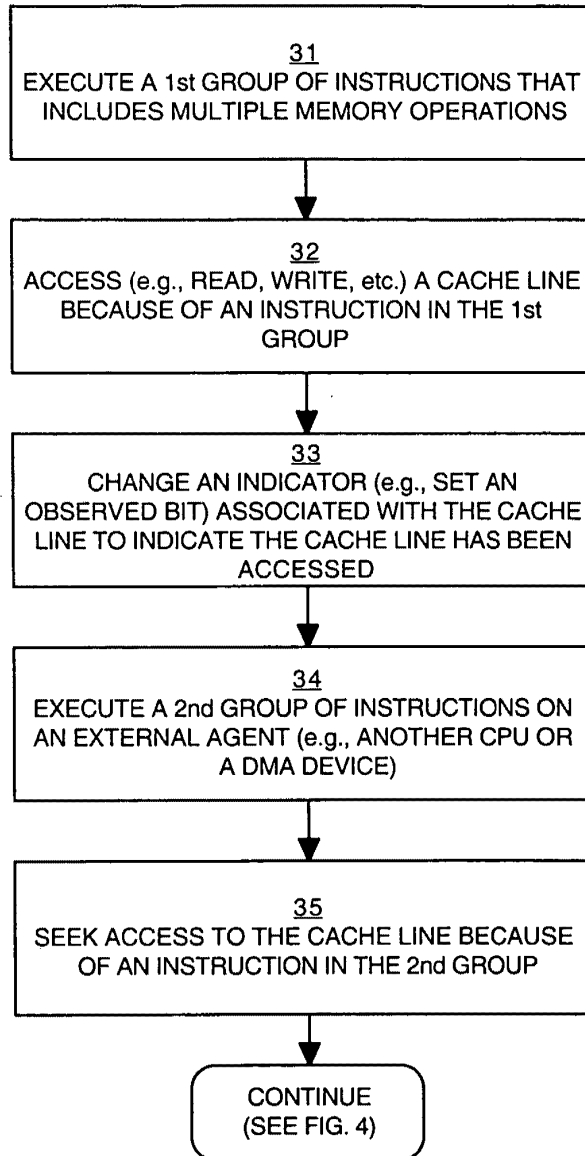


Figure 3

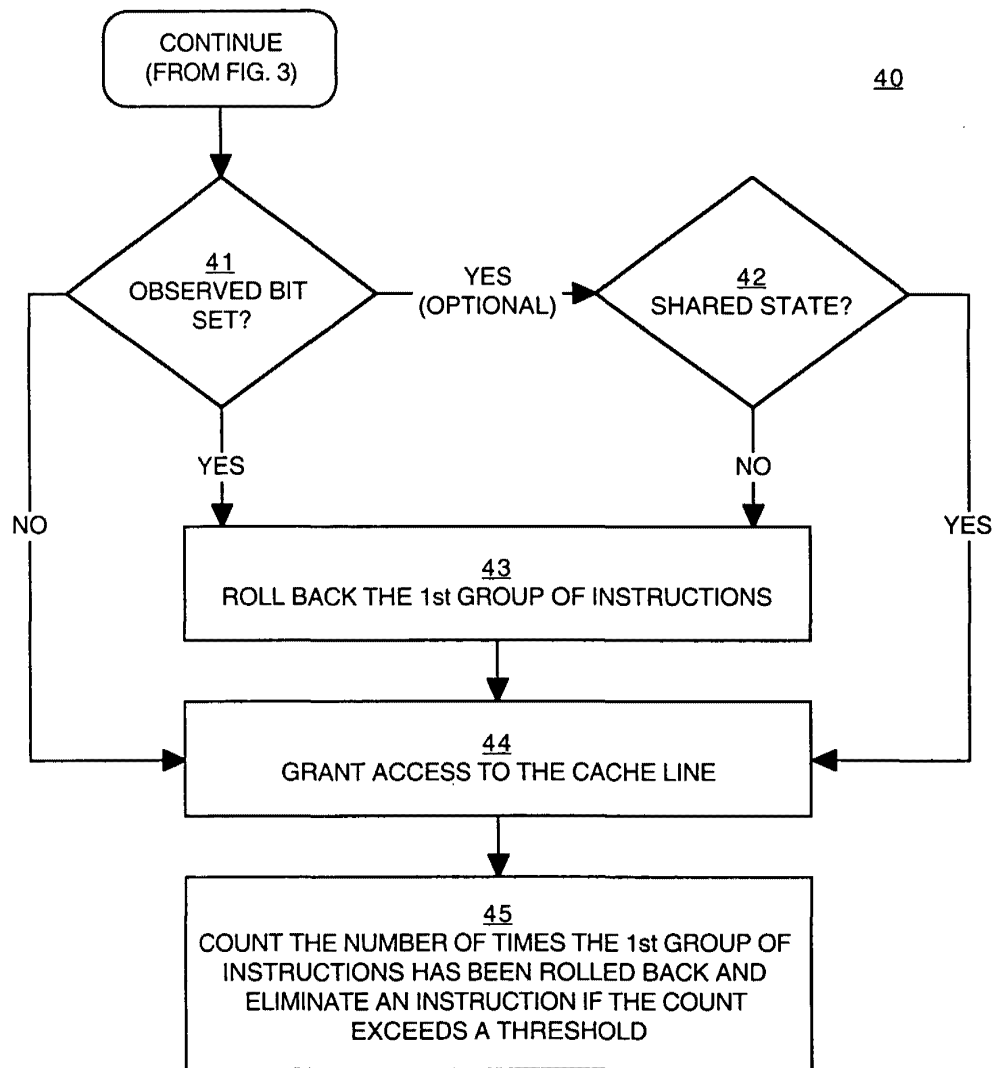


Figure 4

50

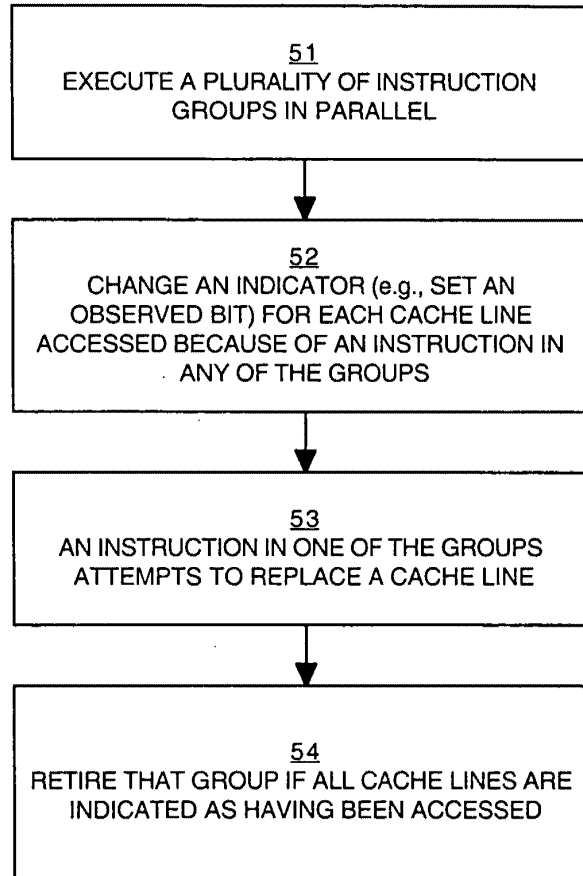


Figure 5

Declaration and Power of Attorney for a Patent Application

Declaration

As below named inventor, I hereby declare that my residence post office address, and citizenship are as stated below my name. Further, I hereby declare that I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY

the specification of which:

☒ is attached hereto, or
_____ was filed on _____ as application serial no. _____ : and
_____ was amended on _____

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above; and

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56.

Foreign Priority Claim

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Number	Country	Date Filed	Priority Claimed
_____	_____	_____	_____ yes _____ no
_____	_____	_____	_____ yes _____ no

U.S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 and 119(e) of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Serial Number	Filing Date	Status (patented/pending/abandoned)
_____	_____	_____
_____	_____	_____

Power of Attorney

As a named inventor, I hereby appoint practitioners associated with this Customer Number:

45590

as attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(b).

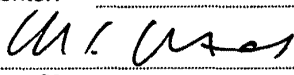
Send Correspondence to:

WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, California 95113
(408) 938-9060

Signatures

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor: Guillermo J. Rozas

Inventor's Signature 

Date April 6, 2005

Residence Los Gatos, CA

Citizenship USA

(City State)

P.O. Address 104 Magneson Terrace, Los Gatos, CA 95032

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

04/12/2005 AADOF01 00000005 11102127
01 FC:1011 300.00 OP
02 FC:1111 500.00 OP
03 FC:1311 200.00 OP
04 FC:1201 200.00 OP
05 FC:1202 600.00 OP

PTO-1556
(5/87)

*U.S. Government Printing Office: 2002 — 489-257/69033

PATENT APPLICATION FEE DETERMINATION RECORD

Effective December 8, 2004

111 02127

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	32	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	32 minus 20 =	12
INDEPENDENT CLAIMS	4 minus 3 =	1
MULTIPLE DEPENDENT CLAIM PRESENT ☐		

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus	**
Independent	*	Minus	***
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus	**
Independent	*	Minus	***
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT C	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus	**
Independent	*	Minus	***
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

* 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.

SMALL ENTITY TYPE ☐

OR OTHER THAN SMALL ENTITY

RATE	FEE
BASIC FEE	150.00
X\$ 25=	
X100=	
+180=	
TOTAL	

RATE	FEE
BASIC FEE	300.00
X\$50=	600
X200=	200
+360=	
TOTAL	1100

SMALL ENTITY OR

OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE
X\$ 25=	
X100=	
+180=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$50=	
X200=	
+360=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$ 25=	
X100=	
+180=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$50=	
X200=	
+360=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$ 25=	
X100=	
+180=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$50=	
X200=	
+360=	
TOTAL ADDIT. FEE	

Inventor(s): Guillermo J. Rozas

Title: MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE
CONSISTENCY

**REQUEST AND CERTIFICATION
UNDER 35 U.S.C. 122(b)(2)(B)(i)**

I hereby certify that the invention disclosed in the attached application **has not and will not be** the subject of an application filed in another country, or under a multilateral agreement, that requires publication at eighteen months after filing. I hereby request that the attached application not be published under 35 U.S.C. 122(b).

Date: 4/7/05

By: William A. Zarbis
William A. Zarbis
Reg. No. 46,120

This request must be signed in compliance with 37 CFR 1.33(b) and submitted with the application **upon filing**.

Applicant may rescind this nonpublication request at any time. If applicant rescinds a request that an application not be published under U.S.C. 122(b), the application will be scheduled for publication at eighteen months from the earliest claimed filing date for which a benefit is claimed.

If applicant subsequently files an application directed to the invention disclosed in the attached application in another country, or under a multilateral international agreement, that requires publication of applications eighteen months after filing, the applicant must notify the United States Patent and Trademark Office of such filing within forty-five (45) days after the date of the filing of such foreign or international application. **Failure to do so will result in abandonment of this application (35 U.S.C. 122(b)(2)(B)(iii)).**

17236 U.S. PTO
040705

Attorney Docket No.: TRAN-P084

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

I hereby certify that this transmittal of the below described documents is being deposited with the United States Postal Service in an envelope bearing Express Mail Postage and an Express Mail label, with the below serial number, addressed to the Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450, on the below date of deposit.			
Express Mail Label No.:	EV561340528US	Name of Person Making the Deposit:	Anthony Chou
Date of Deposit:	04/07/05	Signature of the Person Making the Deposit:	<i>Anthony Chou</i>

112948 U.S. PTO
11/102127

040705

Inventor(s): Guillermo J. Rozas

Title: MEMORY MANAGEMENT METHODS AND SYSTEMS THAT SUPPORT CACHE CONSISTENCY

Mail Stop Patent Application
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
Sir:

Transmittal of a Patent Application
(Under 37 CFR §1.53)

Transmitted herewith is the above identified patent application, including:

- ☒ Specification, claims and abstract, totaling 30 pages.
- ☐ Formal drawings, totaling _____ pages.
- ☒ Informal drawings, totaling 5 pages.
- ☒ Declaration and Power of Attorney.
- ☐ Information Disclosure statement.
- ☐ Form 1449
- ☒ Assignment(s)
- ☒ Assignment Recordation Form (duplicate)
- ☒ Request and Certification under 35 U.S.C. 122(b)(2)(B)(i)

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

Application Type	FILING FEES		SEARCH FEES		EXAMINATION FEES		Fees Paid (\$)
	Fee (\$)	Small Entity	Fee (\$)	Small Entity	Fee (\$)	Small Entity	
Utility	300	150	500	250	200	100	\$1,000.00
Design	200	100	100	50	130	65	\$
Plant	200	100	300	150	160	80	\$
Reissue	300	150	500	250	600	300	\$
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description

Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent
Each independent claim over 3 or, for reissues, each independent claim more than in the original patent
Multiple dependent claims

Small Entity	
Fee (\$)	Fee (\$)
50	25
200	100
360	180

Total Claims	Extra Claims	Fee (\$)	Fee Paid (\$)	Multiple Dependent Claims
32 -20 or HP=	12	x \$50.00	\$600.00	Fee (\$)
HP= highest number of total claims paid for, if greater than 20				
Indep. Claims	Extra Claims	Fee (\$)	Fee Paid (\$)	
4 -3 or HP=	1	x \$200	\$200.00	
HP= highest number of independent claims paid for, if greater than 3				

3. APPLICATION SIZE FEE

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 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets						
Total Sheets		Extra Sheets		Number of each additional 50 or fraction thereof	Fee \$	Fee Paid \$
36	-100	0	/50		=	\$0.00

4. OTHER FEE(S)

Non-English specification, \$130 fee (no small entity discount)
Other: Assignment Recordation Fee: \$40.00

Total Fees Due (\$)

\$1,840.00

5. PAYMENT OF FEES

The full fee due in connection with this communication is provided as follows:

1. Not enclosed
 - ☐ No filing fee is to be paid at this time.
2. Enclosed
 - ☒ Filing fee
 - ☒ Recording assignment
 - ☒ The Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No.: 23-0085. A duplicate copy of this authorization is enclosed.
 - ☒ A check in the amount of \$1,840.00
 - ☐ Charge any fees required or credit any overpayments associated with this filing to Deposit Account No.: 23-0085.

This application is filed pursuant to 37 C.F.R. § 1.53 in the name of the above-identified Inventor(s).

[X] Customer No: 45590

Please direct all correspondence concerning the above-identified application to the following address:

WAGNER, MURABITO & HAO LLP
Two North Market Street, Third Floor
San Jose, California 95113
(408) 938-9060

[X] This transmittal ends with this page.

Respectfully submitted,

Date: 4/7/05

By: W.A. Zarpis
William A. Zarpis
Reg. No. 46,120