(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 March 2013 (07.03.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/032437 A1

(51) International Patent Classification:
G06F 12/08 (2006.01) *G06F 12/02* (2006.01)

(21) International Application Number:
PCT/US2011/049584

(22) International Filing Date:
29 August 2011 (29.08.2011)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, MS: RNB-4-150, Santa Clara, California 95052 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **KACEVAS, Nicolas** [IL/US]; 140 Dunstable Way, Folsom, California 95630 (US).

(74) Agents: **TROP, Timothy N.** et al.; Trop, Pruner & Hu, P.C., 1616 S. Voss Rd., Ste. 750, Houston, Texas 77057-2631 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ,

[Continued on next page]

(54) Title: PROGRAMMABLY PARTITIONING CACHES

(57) Abstract: Agents may be assigned to discrete portions of a cache. In some cases, more than one agent may be assigned to the same cache portion. The size of the portion, the assignment of agents to the portion and the number of agents may be programmed dynamically in some embodiments.

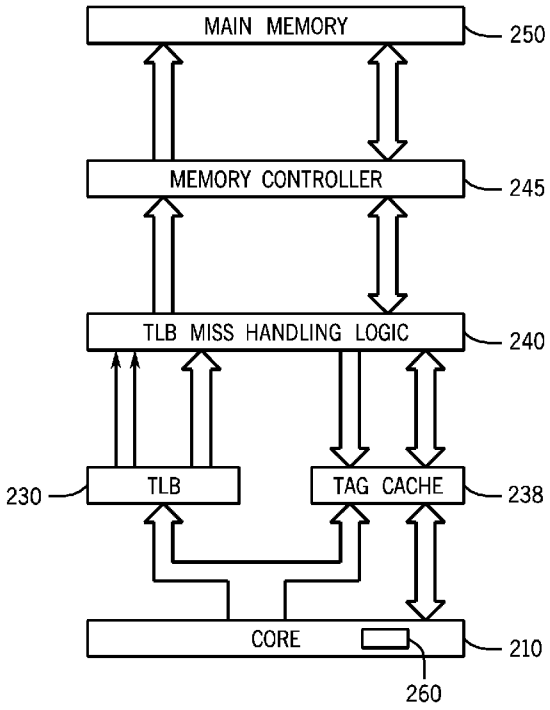

FIG. 1

WO 2013/032437 A1

TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

PROGRAMMABLY PARTITIONING CACHES

Background

This relates generally to the use of storage in electronic devices and, particularly, to the use of storage in connection with processors.

A processor may use a cache to store frequently reused material. By storing frequently reused information in the cache, the information may be accessed more quickly.

In modern processors, translation lookaside buffers (TLBs) store address translations from a virtual address to a physical address. These address translations are generated by the operating system and stored in memory within page table data structures, which are used to populate the translation lookaside buffer.

Brief Description of the Drawings

Figure 1 is a system depiction for one embodiment of the present invention;

Figure 2 is a schematic depiction of cache partitioning in accordance with one embodiment of the present invention;

Figure 3 is a schematic depiction of a cache partition assignment and replacement algorithm in accordance with one embodiment of the present invention; and

Figure 4 is a flow chart for one embodiment of the present invention.

Detailed Description

In accordance with some embodiments, a cache may be broken up into addressable partitions that may be programmably configured. The cache size may be configured programmably, as may be the assignment of agents to particular partitions within the cache. In addition, it may be programmably determined whether or not two or more agents may be assigned to use the same cache partition at any time period.

In this way, more effective utilization of available cache space may be achieved in some embodiments. This may result in more efficient accessing of information from the cache, in some cases, which may improve access time and may improve the amount of information that can be stored within a cache.

Programming of the partitioning of the cache may be done statically, in that it is set from the beginning and is not changed. Partitioning may also be done dynamically, programmably adjusting to changing conditions during operation of an associated processor or controller.

While the following example refers to a translation lookaside buffer, the present

invention is applicable to a wide variety of caches used by processors. In any case where multiple clients or agents request access to a cache, partitioning the cache in a programmable way may prevent clients from thrashing each other to access the cache.

As used herein, an "agent" may be code or hardware that stores or retrieves code or data in a cache.

In some embodiments, the cache may be fully associative. However, in other embodiments, the cache may be any cache with a high level of associativity. For example, caches with associativity higher than four-way associativity may benefit more from some aspects of the present invention.

The cache 230, shown in Figure 1, is illustrated as a translation lookaside buffer, but the present invention is in no way limited to translation lookaside buffers and it is applicable to caches in general.

The system shown in Figure 1 may be a desktop or mobile device. For example, the system may be a laptop, a tablet, a mobile Internet device (MID), or a smart phone, to mention some examples.

The core 210 may be any processor, controller, or even a direct memory access (DMA) controller core. The core 210 may include a storage 260 that may store software for controlling the programming of partitions within the translation lookaside buffer 230. In other embodiments, the programming may be stored externally of the core. The core may also communicate with a tag cache 238 in an embodiment that uses stored kernel accessible bits that include state information or metadata for each page of memory. Connected to the translation lookaside buffer and tag cache is a translation lookaside buffer miss handling logic 240, in turn, coupled to a memory controller 245 and main memory 250, such as a system memory.

The core may request information in a particular page of main memory 250. Accordingly, core 210 may provide an address to both the translation lookaside buffer 230 and tag cache 238. If the corresponding physical to virtual translation is not present in the translation lookaside buffer 230, a translation lookaside buffer miss may be indicated and provided to the miss handling logic 240. The logic 240, in turn, may provide the requested address to the memory controller 245 to enable loading of a page table entry into the translation lookaside buffer 230. A similar methodology may be used if a requested address does not hit a tag cache entry in the tag cache, as a request may be made through the miss handling logic 240 and memory controller 245 to obtain tag information from its dedicated storage in main memory 250 and to provide it for storage in the tag cache 238.

The cache 238 may be partitioned, as shown in Figure 2. In this example, there are four agents, agents A-D. Any number of agents may be accommodated in other embodiments. The

lowermost cache (i.e., the one with lower numbered addresses) is assigned to agents A and B, the middle cache is assigned to agent C, and the top cache is assigned to agent D, in this example. The partitions are defined, in this example, using minimum and maximum addresses called LRA0, LRA1, LRA2, minimum and maximum. The bottom and top cache lines of a partition may be identified by their addresses for each partition in one embodiment.

While an example is given wherein the cache is divided into partitions or portions based on cache line addresses, caches may also be partitioned based on other granularities of memory, including blocks, sets of blocks, and conventional partitions.

Thus, the size of each partition may be defined by its minimum and maximum addresses in the example illustrated in Figure 2. Likewise, the assignments of agents to partitions may be determined programmably. Finally, whether or not to use overlapping (where more than one agent is assigned to the same partition) may be determined programmably.

For example, with respect to overlapping, it may be determined whether two or more agents are likely to use a partition at the same time. If so, it may be more efficient to assign the agents to different partitions. However, if the agents are likely to use the partition at different times, the usage of the partition is more effectively allocated if the same agents are assigned to the same partition. Other rationales for assigning overlapping agents to a partition, or not, may also be used.

In addition, different agents may be provided with partitions of different programmable size. A wide variety of considerations may go into programming partition size, including known relationships with respect to how much cache space is used by a particular agent or type of agent. Moreover, the size of the partition may be adjusted dynamically during the course of partition usage. For example, based on rate of cache line storage, more lines may be allocated. Likewise, agents may be reassigned to partitions dynamically and overlapping may be applied or undone dynamically, based on various conditions that may exist during processing.

The partitions may also overlap in other ways. For example, an agent A may use half of the available entries of a partition, agent B may use the other half, and agent C may use all of the entries. In this case, the partition is split between two agents, each of which uses a portion of the partition, while another agent overlaps with each of those agents. To implement such an arrangement, LRA A is mapped to the lower half, LRA B is mapped to the upper half, and LRA C is mapped to the whole partition, overlapping with the regions A and B. This type of mapping may be useful if the agents A and B are active at the same time, while agent C is active at a different time.

Referring to Figure 3, in accordance with some embodiments, an algorithm for assigning agents to cache partitions and a cache replacement policy is described. In some embodiments, a

least recently allocated (LRA) cache replacement policy may be used.

In the upper right hand corner (at 10), the agents are programmably assigned to cache partitions. This may be done by assigning minimum and maximum addresses labeled LRA, followed by a number, and a minimum and a maximum address. Thus, a partition for use by agent A is assigned at block 20, a partition for use by agent B is assigned at block 22, a partition for use by agent C is assigned at block 24, and a partition for use by agent D is assigned at block 26.

An agent selection input (e.g., use LRA2) is provided to the multiplexer 28 to select a particular agent to be served. Then the block 50, 52, or 54, assigned to that particular agent, is activated when the agent is currently being served. Thus, if the agent D is assigned to LRA2, as illustrated in Figure 2, then the line labeled "use LRA2" may be activated to activate the block 54, while the blocks 50 and 52 are inactive in one embodiment.

Each of the blocks 50, 52, and 54 may otherwise work the same way. Each block takes the minimum address and maximum address, such as LRA2 min and LRA2 max, in the case of block 54, and, on each use of the block, adds (block 32) one to a counter 38. Then a check at the multiplexer/ counter 40 determines whether that LRA block has actually been selected. If so, the counter 40 is incremented. When the maximum address (i.e. the top address, for example) is reached (block 36), then the count rolls over and the least recently allocated address is overwritten in this embodiment. Embodiments may overwrite based on other schemes as well, including least used address.

Each of the registers 30 and 34 may be rewritten to change the size of the partition. In addition, it is an easy matter to change which block is assigned to which agent so that the agents can be programmably reassigned. Overlapping may be achieved simply by assigning the same partition with the same LRA min and max to two or more agents.

Referring to Figure 4, in accordance with one embodiment, a cache configuration sequence 60 may be implemented in software, hardware, and/or firmware. In one embodiment, the cache configuration sequence 60 may be implemented in software as computer readable instructions stored in a non-transitory computer readable medium, such as an optical, magnetic, or semiconductor memory. As one example, the instructions may be stored in the storage 260 as part of a core 210. However, they may be stored instead independently of the core 210 and may be executed by the core 210 in some embodiments.

The core 210 may be any kind of processor, including a graphics processor, a central processing unit, or a microcontroller. The core 210 may be part of an integrated circuit which includes both graphics and central processing units integrated thereon or it may be part of any integrated circuit with multiple cores on the same integrated circuit. Similarly, the core 210 may

be on its own integrated circuit without other cores.

Continuing with Figure 4, first the core may determine whether to use overlapping, as indicated in block 62. Based on whether or not overlapping is used and based on characteristics of the agents that use the cache, agents may be assigned to partitions, as indicated in block 64. Then the partition size may be determined, as indicated in block 66, for example, by assigning minimum and maximum addresses. As mentioned previously, other partition assignment techniques may also be used, including assigning a given number of blocks or partitions to given agents.

In some embodiments, the order of the steps may be changed. Also, some of the steps may be dynamic and some may be static, in some embodiments. Some of the steps may be omitted in some embodiments. As still another example, different processors on the same integrated circuit may have different programmable configurations. In may also be possible for agents to share partitions associated with different processors, in some embodiments. In still other embodiments, a single partitioned cache may be used by more than one processor.

In some embodiments, registers may be provided for each agent to programmably store LRA min and LRA max, any overlapping and agent to cache partition assignments. The registers may also store partition granularity, for example, when partitions are made of a given number of regularly sized units, such as cache lines, blocks, or sets of blocks.

The graphics processing techniques described herein may be implemented in various hardware architectures. For example, graphics functionality may be integrated within a chipset. Alternatively, a discrete graphics processor may be used. As still another embodiment, the graphics functions may be implemented by a general purpose processor, including a multicore processor.

References throughout this specification to "one embodiment" or "an embodiment" mean that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one implementation encompassed within the present invention. Thus, appearances of the phrase "one embodiment" or "in an embodiment" are not necessarily referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be instituted in other suitable forms other than the particular embodiment illustrated and all such forms may be encompassed within the claims of the present application.

While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.

What is claimed is:

1. A method comprising:

programmably assigning agents to discrete portions of a cache.

2. The method of claim 1 including programmably assigning more than one agent to the same discrete cache portion.

3. The method of claim 1 including programmably setting the size of a cache portion.

4. The method of claim 1 including dynamically changing the assignments of one or more agents to a cache portion.

5. The method of claim 1 including assigning agents to discrete portions of a cache in the form of a translation lookaside buffer.

6. The method of claim 1 including using a cache having an associativity greater than four ways.

7. A non-transitory computer readable medium storing instructions to cause a core to:

assign more than one agent to a discrete part of a cache.

8. The medium of claim 7 further storing instructions to dynamically change the assignment of more than one agent to said discrete part of said cache.

9. The medium of claim 8 further storing instructions to programmably set the size of a cache part.

10. The medium of claim 8 further storing instructions to assign agents to discrete parts of a cache.

11. The medium of claim 10 further storing instructions to change the assignments of one or more agents to a cache part.

12. The medium of claim 8 further storing instructions to assigning agents to discrete parts of a cache in the form of a translation lookaside buffer.

13. The medium of claim 8 further storing instructions to use a cache having an associativity greater than four ways.

14. An apparatus comprising:

a processor core; and

a cache coupled to said core, said core to assign agents to discrete portions of a cache.

15. The apparatus of claim 14, said core to programmably assign more than one agent to the same discrete cache portion.

16. The apparatus of claim 14, said core to programmably set the size of a cache portion.

17. The apparatus of claim 14, said core to dynamically change the assignment of one or more agents to a cache portion.

18. The apparatus of claim 14 wherein said cache is a translation lookaside buffer.

19. The apparatus of claim 14, said cache having an associativity greater than four ways.

20. The apparatus of claim 14 wherein said core is a graphics core and said cache is a translation lookaside buffer.

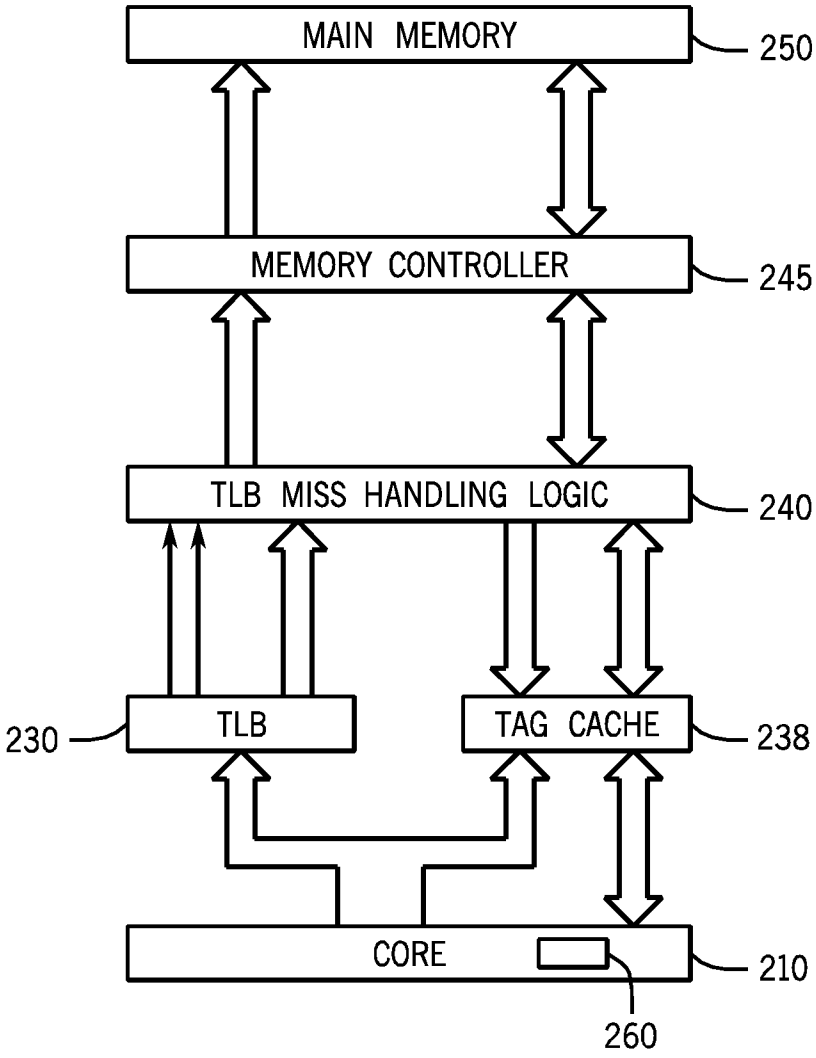
MAIN MEMORY
250
MEMORY CONTROLLER
245
TLB MISS HANDLING LOGIC
240
230
TLB
TAG CACHE
238
CORE
210
260

FIG. 1

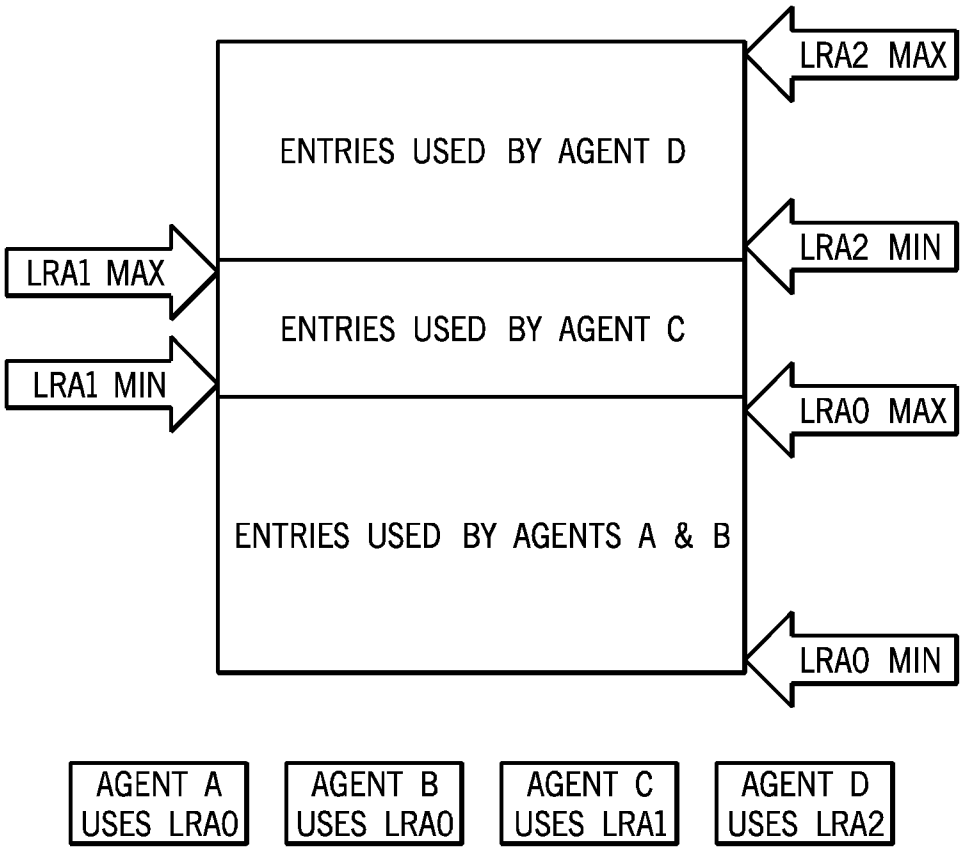
LRA2 MAX
ENTRIES USED BY AGENT D
LRA2 MIN
LRA1 MAX
ENTRIES USED BY AGENT C
LRA1 MIN
LRA0 MAX
ENTRIES USED BY AGENTS A & B
LRA0 MIN
AGENT A USES LRA0
AGENT B USES LRA0
AGENT C USES LRA1
AGENT D USES LRA2

FIG. 2

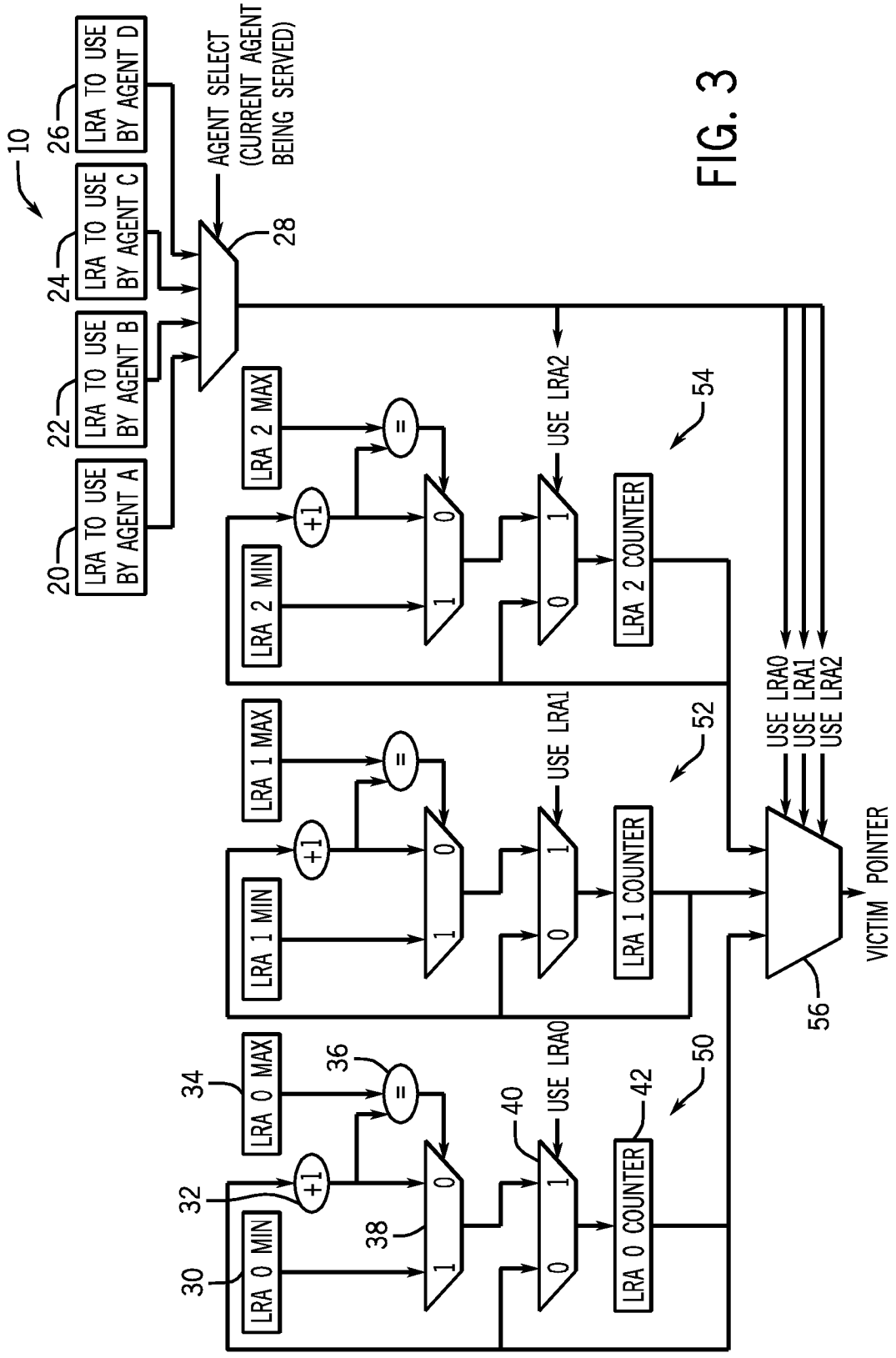
10
20
LRA TO USE BY AGENT A
22
LRA TO USE BY AGENT B
24
LRA TO USE BY AGENT C
26
LRA TO USE BY AGENT D
28
AGENT SELECT (CURRENT AGENT BEING SERVED)
30
LRA 0 MIN
32
+1
34
LRA 0 MAX
36
=
38
40
USE LRA0
42
LRA 0 COUNTER
50
LRA 1 MIN
+1
LRA 1 MAX
=
USE LRA1
52
LRA 1 COUNTER
LRA 2 MIN
+1
LRA 2 MAX
=
USE LRA2
54
LRA 2 COUNTER
USE LRA0
USE LRA1
USE LRA2
56
VICTIM POINTER

FIG. 3

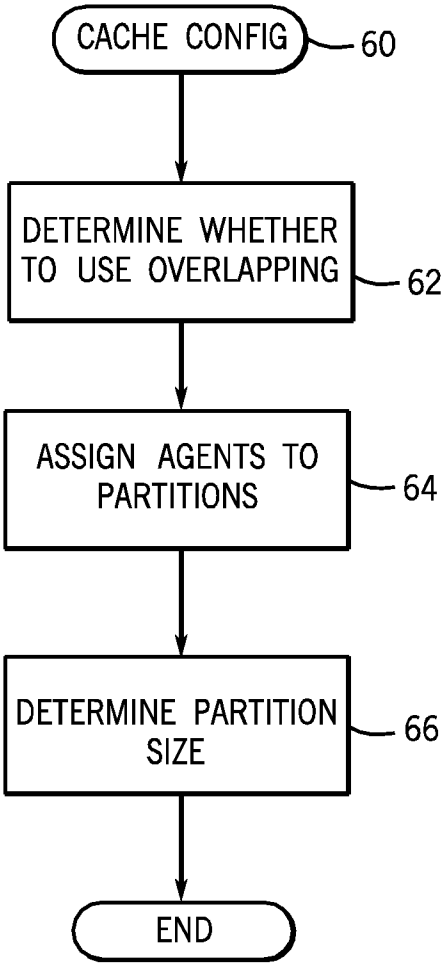
CACHE CONFIG
60
DETERMINE WHETHER TO USE OVERLAPPING
62
ASSIGN AGENTS TO PARTITIONS
64
DETERMINE PARTITION SIZE
66
END

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2011/049584

A. CLASSIFICATION OF SUBJECT MATTER

G06F 12/08(2006.01)i, G06F 12/02(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 12/08; G06F 12/10; G06F 12/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: cache, TLB, partition

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005-0055510 A1 (DAVID, T. HASS et al.) 10 March 2005 See [0008] - [0142] and figure 5C.	1-20
A	US 2008-0104362 A1 (WILLIAM M. BUROS et al.) 01 May 2008 See [0017] - [0039].	1-20
A	US 2010-0250853 A1 (ORRAN Y. KRIEGER et al.) 30 September 2010 See [0016] - [0042].	1-20
A	US 2004-0015675 A1 (ALAN KYKER et al.) 22 January 2004 See [0025] - [0055].	1-20
A	US 2009-0300319 A1 (EHUD COHEN et al.) 03 December 2009 See [0015] - [0045].	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 APRIL 2012 (27.04.2012)	**01 MAY 2012 (01.05.2012)**

Name and mailing address of the ISA/KR
Korean Intellectual Property Office
Government Complex-Daejeon, 189 Cheongsa-ro,
Seo-gu, Daejeon 302-701, Republic of Korea
Facsimile No. 82-42-472-7140

Authorized officer
KWON, Oh Seong
Telephone No. 82-42-481-8526

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2011/049584

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005-0055510 A1	10.03.2005	CN 100498757 C	10.06.2009
		CN 101878475 A	03.11.2010
		CN 1842781 A0	04.10.2006
		EP 2174229 A1	14.04.2010
		JP 04-498356 B2	23.04.2010
		JP 2007-500886 A	18.01.2007
		JP 2009-026320 A	05.02.2009
		JP 2010-079921 A	08.04.2010
		KR 10-2006-0132538 A	21.12.2006
		US 2004-0103248 A1	27.05.2004
		US 2005-0027793 A1	03.02.2005
		US 2005-0033831 A1	10.02.2005
		US 2005-0033832 A1	10.02.2005
		US 2005-0033889 A1	10.02.2005
		US 2005-0041651 A1	24.02.2005
		US 2005-0041666 A1	24.02.2005
		US 2005-0044308 A1	24.02.2005
		US 2005-0044323 A1	24.02.2005
		US 2005-0044324 A1	24.02.2005
		US 2005-0055502 A1	10.03.2005
		US 2005-0055503 A1	10.03.2005
		US 2005-0055504 A1	10.03.2005
		US 2005-0055540 A1	10.03.2005
		US 2005-0086361 A1	21.04.2005
		US 2007-0204130 A1	30.08.2007
		US 2008-0062927 A1	13.03.2008
		US 2008-0126709 A1	29.05.2008
		US 2008-0140956 A1	12.06.2008
		US 2008-0184008 A1	31.07.2008
		US 2008-0216074 A1	04.09.2008
		US 2010-0042785 A1	18.02.2010
		US 2010-0077150 A1	25.03.2010
		US 2010-0318703 A1	16.12.2010
		US 2011-225398 A1	15.09.2011
		US 7334086 B2	19.02.2008
		US 7346757 B2	18.03.2008
		US 7461213 B2	02.12.2008
		US 7467243 B2	16.12.2008
		US 7509462 B2	24.03.2009
		US 7509476 B2	24.03.2009
		US 7627717 B2	01.12.2009
		US 7627721 B2	01.12.2009
		US 7864760 B2	04.01.2011
		US 7924828 B2	12.04.2011
		US 7941603 B2	10.05.2011
		US 7961723 B2	14.06.2011
		US 7984268 B2	19.07.2011
		US 7991977 B2	02.08.2011
		US 8015567 B2	06.09.2011

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2011/049584

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 8037224 B2	11.10.2011
		US 8065456 B2	22.11.2011
		WO 2005-013061 A2	10.02.2005
		WO 2005-013061 A3	10.02.2005
		WO 2009-017668 A1	05.02.2009
US 2008-0104362 A1	01.05.2008	None	
US 2010-0250853 A1	30.09.2010	None	
US 2004-0015675 A1	22.01.2004	US 6594734 B1	15.07.2003
US 2009-0300319 A1	03.12.2009	None	