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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 February 2001 (08.02.2001)

PCT

(10) International Publication Number
WO 01/09729 A1

(51) International Patent Classification⁷: G06F 12/08

(21) International Application Number: PCT/US00/20496

(22) International Filing Date: 27 July 2000 (27.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/363,789 29 July 1999 (29.07.1999) US

(71) Applicant: MICRON TECHNOLOGY, INC. [US/US]; Lynch, Michael L., Legal Dept., 8000 S. Federal Way, Boise, ID 83706-9632 (US).

(72) Inventor: JEDDELOH, Joseph, M.; 3518 Edmund Boulevard, Minneapolis, MN 55406 (US).

(74) Agent: HU, Dan, C.; Trop, Pruner & Hu, P.C., Suite 100, 8554 Katy Freeway, Houston, TX 77024 (US).

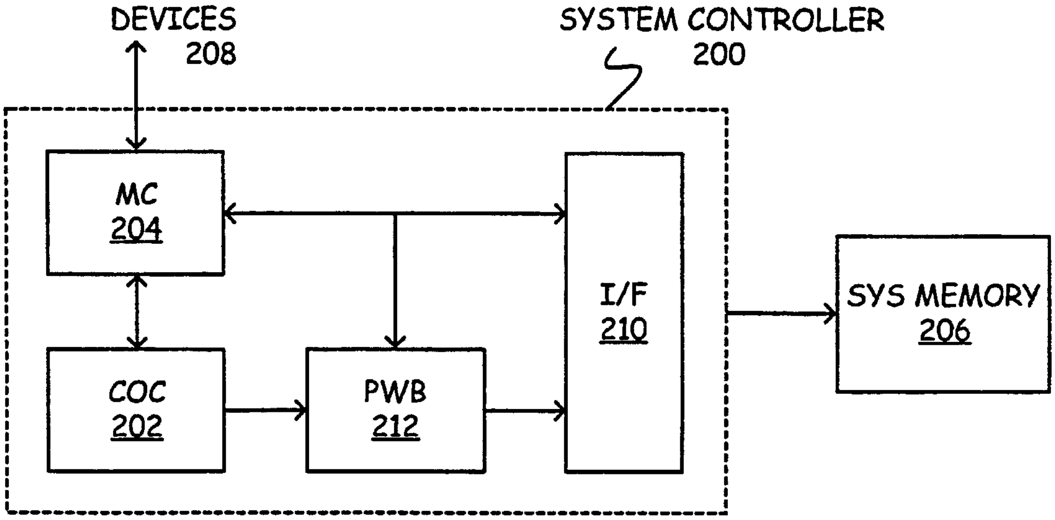
(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: CAST-OUT CACHE



(57) Abstract: Methods and devices to reduce processor-to-system memory access latency through the use of a memory buffer (202) for the storage of cache lines flushed (cast out) from conventional level-1 (L1) and/or level-2 (L2) processor caches are described. The memory buffer (202), referred to as a cast-out cache, may be incorporated within a system controller (200) and/or memory controller device (204).

Cast-Out Cache

Background

The invention relates generally to computer memory systems and more particularly, but not by way of limitation, to a caching technique to improve host processor memory access operations.

In a typical computer system, program instructions and data are read from and written to system memory at random addresses. To combat this random nature of memory access operations level-1 (L1) and level-2 (L2) cache memories have been used to decrease the time, or number of clock cycles, a given processor must spend communicating with system memory during memory read and write operations.

Cache memories rely on the principle of access locality to improve the efficiency of processor-to-memory operations and, therefore, overall computer system performance. In particular, when a processor accesses system memory for program instructions and/or data, the information retrieved includes not only the targeted instructions and/or data, but additional bytes of information that surround the targeted memory location. The sum of the information retrieved and stored in the cache is known as a "cache line." (A typical cache line may comprise 32 bytes.) The principle of access locality predicts that the processor will very probably use the additional retrieved bytes subsequent to the use of the originally targeted program instructions. During such operations as the execution of program loops, for example, information in a single cache line may be used multiple times. Each processor initiated memory access that may be satisfied by information already in a cache (referred to as a "hit"), eliminates the need to access system memory and, therefore, improves the operational speed of the computer system. In contrast, if a processor initiated memory access can not be satisfied by information already in a cache (referred to as a "miss"), the processor must access system memory – causing a new cache line to be brought into the cache and, perhaps, the removal of an existing cache line.

Referring to FIG. 1, many modern computer systems 100 utilize processor units 102 that incorporate small L1 cache memory 104 (e.g., 32 kilobytes, KB) while also providing larger external L2 cache memory 106 (e.g., 256 KB to 612 KB). As shown, processor unit 102, L1 cache 104 and L2 cache 106 are coupled to system memory 108 via processor bus 110 and system controller 112. As part of processor unit 102 itself, L1 cache 104 provides the fastest possible access to stored cache line information. Because of its relatively small size however, cache miss operations may occur frequently. When a L1 cache miss occurs, L2 cache 106 is searched for the targeted program data and/or program instructions (hereinafter collectively referred to as data). If L2 cache 106 contains the targeted data, the appropriate cache line is transferred to L1 cache 104. If L2 cache 106 does not contain the targeted data, an access operation to system memory 108 (typically mediated by system controller 112) is initiated. The time between processor unit 102 initiating a search for target data and the time that data is acquired or received by the processor unit (from L1 cache 104, L2 cache 106 or memory 108) is known as read latency. A key function of caches 104 and 106 is to reduce the processor unit 102's read latency.

If L1 cache 104 is full when a new cache line is brought in for storage, a selected cache line is removed (often referred to as flushed). If the selected cache line has not been modified since being loaded into L1 cache 104 (i.e., the selected cache line is "clean"), it may be replaced immediately by the new cache line. If the selected cache line has been modified since being placed into L1 cache 104 (i.e., the selected cache line is "dirty"), it may be flushed to L2 cache 106. If L2 cache 106 is full when a L1 cache line is brought in for storage, one of its cache lines is selected for replacement. As with L1 cache 104, if the selected cache line is clean it may be replaced immediately. If the selected cache line is dirty, however, it may be flushed to posted write buffer 114 in system controller 112. The purpose of posted write buffer 114 is to provide short-term storage of dirty cache lines that are in the process of being written to system memory 108. (Posted write buffers 114 are typically only large enough to store a few, e.g., 8, cache lines.)

While reasonably large by historical standards, the size of both L1 cache 104 and L2 cache 106 are small relative to the amounts of data accessed by modern software applications. Because of this, computer systems employing conventional L1 and L2 caches (especially those designed for multitasking operations) may exhibit unacceptably high cache miss rates. One effect of high cache miss rates is to increase the latency time of processor unit read operations. Thus, it would be beneficial to provide a mechanism to reduce the memory latency time experienced by host processor units.

Summary

In one embodiment the invention provides a computer system comprising a processor, a level-1 cache (operatively coupled to the processor), a level-2 cache (operatively coupled to the processor), a system memory, and a system controller (operatively coupled to the processor, level-1 cache, level-2 cache and system memory), wherein the system controller has a memory buffer adapted to store cache lines flushed (cast out) from one or more processor caches. The memory buffer, referred to herein as a cast-out cache, may be configured as a set associative or fully associative memory and may comprise dynamic or static random access memory integrated into the system controller.

In another embodiment, the invention provides a method to control memory access transactions. The method includes receiving a memory access request signal from a device, identifying the device, selecting a cache structure based on the identified device, using the selected cache structure to satisfy the memory access request. The acts of selecting a cache structure and using the selected cache structure may comprise selecting a cache structure if the identified device is a processor unit, otherwise accessing a system memory to satisfy the memory request. Methods in accordance with the invention may be stored in any media that is readable and executable by a computer system.

Brief Description of the Drawings

Figure 1 shows a block diagram of a prior art computer system having a memory architecture incorporating level-1 and level-2 cache memories.

Figure 2 shows a block diagram of a system controller that incorporates a cast-out cache in accordance with one embodiment of the invention.

Figure 3 shows a flow diagram format of how a memory controller processes a new cast-out cache entry in accordance with one embodiment of the invention.

Figure 4 shows a flow diagram of how a memory controller processes a memory access request using a cast-out cache in accordance with one embodiment of the invention.

Figure 5 shows a block diagram of a computer system having a cast-out cache in accordance with one embodiment of the invention.

Figure 6 shows a modification to FIG. 4 wherein a cast-out cache is used only for transactions associated with a processor unit.

Figure 7 shows another modification to FIG. 4 wherein a memory controller may access two or more cast-out cache structures.

Detailed Description

Techniques (including methods and devices) to reduce processor-to-system memory access latency through the use of a memory buffer for the storage of cache lines flushed from conventional level-1 (L1) and/or level-2 (L2) caches are described. The following embodiments of the invention, described in terms of a memory buffer incorporated within a system controller device, are illustrative only and are not to be considered limiting in any respect.

Referring to FIG. 2, system controller 200 in accordance with one embodiment of the invention incorporates a memory buffer for the storage of cache lines flushed -- cast out -- from a processor's L1 and/or L2 caches (hereinafter referred to as cast-out cache 202). Memory controller 204 mediates data transfers (wherein "data" includes program data and program instructions) between system memory 206 and devices 208 via memory interface 210, posted write buffer 212 and cast-out cache 202. In accordance with the invention, as a cache line is flushed from a processor's cache(s) it is stored in cast-out cache 202 rather than posted write buffer 212 as in conventional computer systems. Subsequent reads to cache lines stored in cast-out cache may be returned to the processor without incurring the latency associated with a full memory access. Illustrative devices 208 include processor units, L1 cache units, L2 cache units, graphics devices, and peripheral or input-output (I/O) devices.

Figure 3 shows, in flow diagram format, how memory controller 204 processes a new cast-out cache entry in accordance with one embodiment of the invention. On receiving a cache line (block 300), system controller 200 determines if cast-out cache 202 has sufficient room to accept the new entry. If cast-out cache 202 does have sufficient room (the "yes" prong of diamond 302), the newly received cache line is stored (block 304) in cast-out cache 202. Each cache line stored in cast-out cache 202 comprises a data component and a tag component, where the tag component further includes a status portion and an address portion. The status portion includes indication of an entries state (e.g., dirty or clean). The address portion includes an indication of the data component's address in memory 206. As would be known to those of ordinary skill, the address portion may be used to organize cast-out cache 202 into a set associative memory (e.g., 2-way, 4-way, and 8-way) or a fully associative memory.

If cast-out cache 202 does not have sufficient room (the "no" prong of diamond 302), a cast-out cache entry is selected (block 306) and flushed to posted write buffer 212 (block

308). Once the selected entry is flushed, the new cache line may be stored (block 304). Memory controller 204 may utilize posted write buffer 212 in a conventional manner; as a temporary staging area for data being written to system memory 206. For example, if cast-out cache 202 is full, the selected cast-out cache entry may be flushed to posted write buffer 212. Any desired cache line replacement algorithm may be employed. In one embodiment, for example, a least recently used (LRU) algorithm may be used to select that cast-out cache entry for removal (block 306). In another embodiment, clean cache lines are selected before dirty cache lines so as to avoid, or postpone, memory write operations. In yet another embodiment, these two techniques may be combined.

Figure 4 shows, in flow diagram format, how memory controller 204 processes a memory access request using cast-out cache 202 in accordance with one embodiment of the invention. After receiving a memory transaction request (block 400), memory controller 204 determines what type of request it is to process. If the received request is a memory read request (the "yes" prong of diamond 402), a check is made to determine if the requested data is in cast-out cache 202. If the requested data is in cast-out cache 202 (the "yes" prong of diamond 404), the requested data is retrieved from cast-out cache 202 (block 406) and returned to the requesting device (block 408) at which point the transaction is complete (block 410). If the requested data is not available in cast-out cache 202 (the "no" prong of diamond 404), the requested data is retrieved from system memory 206 (block 412) and returned to the requesting device (block 408). In one embodiment, cast-out cache 202 is populated with cache lines flushed (cast out) from processor caches only. In this embodiment, only processor unit reads are processed in accordance with FIG. 4 (acts 400 through 412).

If the received memory transaction request is a memory write request (the "no" prong of diamond 402), a test is made to determine if the targeted write address has an entry in cast-out cache 202 (diamond 414). If the targeted address has an associated cast-out cache entry (the "yes" prong of diamond 414), the entry is updated in accordance with the write request (block 416). If the targeted address does not have an associated cast-out cache entry (the "no" prong of diamond 414), a memory write operation is performed (block 418). In one embodiment cast-out cache 202 may be updated during memory write operations in accordance with FIG. 4 when either a processor unit or an input-output (I/O) bus master device writes to memory 206. In this sense, memory controller 204 "snoops" cast-out cache 202 during memory write operations. Devices other than processor units, however, do not generate cache line allocation actions during memory read operations (only cache lines cast out or flushed from processor caches are loaded into cast-out cache 202).

Referring to FIG. 5, computer system 500 in accordance with one embodiment of the invention includes processor unit 502 (incorporating an L1 cache structure, not shown) and L2 cache unit 504 coupled to system controller 200 via processor bus 506. System controller 200 couples accelerated graphics device 508 (via graphics bus 510) and expansion or I/O devices 512 (via system bus 514) to system memory 206 (via memory bus 516). Illustrative processor units (e.g., 502) include the PENTIUM® family of processors and the 80X86 families of processors from Intel Corporation. Illustrative expansion devices 512 include any device designed to operate in concert with system bus 514. For example, if system bus 514 operates in conformance with the peripheral component interconnect (PCI) standard, expansion devices 512 may be any PCI device (e.g., a network interface card). It will be recognized that additional bus structures and devices may be coupled to computer system 500. For example, if system bus 514 operates in accordance with the PCI standard, a PCI-to ISA bridge circuit may be used to couple one or more industry standard architecture (ISA) devices to computer system 500 (e.g., a keyboard controller and non-volatile memory). One illustrative PCI-to-ISA bridge circuit is the 82371AB PCI-to-ISA/IDE controller made by Intel Corporation.

Every memory access request satisfied from the contents of cast-out cache 202, allows memory controller 204 to reduce the memory transaction latency suffered by the requesting device (e.g., processor 502) by avoiding a system memory access operation. In addition, requests satisfied from cast-out cache 202 reduce memory bus 516 loading. The former benefit may be enhanced by making cast-out cache 202 relatively large, 1 to 4 megabytes for example. The latter benefit may further allow memory controller 204 to service multiple memory transaction requests (each associated with a different device) in parallel -- one from cast-out cache 202 and another from system memory 206.

While memory controller 204 may utilize cast-out cache 202 to service a memory request from any device (i.e., devices 208), in one embodiment only those transactions associated with a processor unit (e.g., 502) actually utilize cast-out cache 202. Referring to FIG. 6, for example, the flow diagram of FIG. 3 may be modified so that memory controller 204 determines what type of device issued the request. If the requesting device is a processor unit (the "yes" prong of diamond 600), act in accordance with FIG. 3 are performed. If, on the other hand, the requesting device is not a processor unit (the "no" prong of diamond 600), a system memory access operation is performed (block 602) and the results returned to the requesting device in a conventional manner (block 604).

In another embodiment of the invention, separate cast-out cache structures may be provided for processor units and I/O devices. Referring to FIG. 7, for example, the flow diagrams of FIGS. 4 and 6 may be modified to account for multiple cast-out cache structures. Following receipt of a memory access request (block 400), a series of tests are performed to determine what device issued the request. If the requesting device is a processor unit (the "yes" prong of 600), the processor cast-out cache is selected (block 700) and processing continues as outlined in FIG. 4. If, on the other hand, the requesting device is not a processor unit (the "no" prong of diamond 600), the appropriate cast-out cache structure is selected (block 702) where after processing continues as outlined in FIG. 4. As indicated, there may be two or more cast-out cache structures. In one embodiment, there is a cast-out cache structure for a processor unit and another cast-out cache structure for I/O devices (e.g., devices 512 coupled to system bus 514).

Various changes in the materials, components, circuit elements, as well as in the details of the illustrated operational methods are possible without departing from the scope of the claims. For instance, cast-out cache 202 may incorporate additional buffer memory to serve as temporary storage for cache lines moving in and out of the cache. One such buffer storage may act as a posted-write buffer for entries associated with the cast-out cache. In addition, while cast-out cache 202 and memory controller 204 have been shown as incorporated within system controller 200, it is possible to embody them in a device external to system controller 200. In one embodiment cast-out cache 202 may be a large dynamic random access memory (DRAM) array and memory controller 204 may be a programmable control device integrated, as shown, into system controller 200. In another embodiment, cast-out cache 202 and memory controller 204 may be implemented external to system controller 200 and coupled directly to system bus 514.

As a programmable control device, memory controller 204 may be a single computer processor, a plurality of computer processors coupled by a communications link, or a custom designed state machine. Custom designed state machines may be embodied in a hardware device such as a printed circuit board comprising discrete logic, integrated circuits, or specially designed application specific integrated circuits (ASICs). In addition, acts in accordance with FIGS. 4 through 7 may be performed by a programmable control device executing instructions organized into a program module and stored in a storage device. Storage devices suitable for tangibly embodying program instructions include all forms of non-volatile memory including, but not limited to: semiconductor memory devices such as electrically programmable read only memory (EPROM), electrically erasable programmable read only memory (EEPROM), and flash devices.

While the invention has been disclosed with respect to a limited number of embodiments, numerous modifications and variations will be appreciated by those skilled in the art. It is intended, therefore, that the following claims cover all such modifications and variations that may fall within the true sprit and scope of the invention.

What is claimed is:

1. A computer system, comprising:

a processor unit;

a level-1 cache operatively coupled to the processor unit;

a level-2 cache operatively coupled to the processor unit;

system memory; and

a system controller operatively coupled to the processor unit, level-1 cache, level-2 cache and system memory, the system controller having a memory buffer adapted to store data associated with processor unit initiated transactions to system memory.

2. The computer system of claim 1, wherein the memory buffer is organized as a cache memory.

3. The computer system of claim 2, wherein the cache memory comprises a set-associative cache memory.

4. The computer system of claim 2, wherein the cache memory comprises a fully associative cache memory.

5. The computer system of claim 1, wherein the memory buffer comprises between approximately 1 and 4 megabytes of volatile memory.

6. The computer system of claim 1, wherein the system controller comprises an application specific integrated circuit.

7. The computer system of claim 1, further comprising:

a peripheral component interconnect bus coupled to the system controller; and

one or more devices coupled to the peripheral component interconnect.

8. An integrated circuit system controller, comprising:

a processor interface adapted to communicate with a processor;

a memory interface adapted to communicate with a system memory;

a memory control circuit adapted to mediate memory access operations between a device and the system memory; and

a memory buffer operatively coupled to the memory controller and adapted to store data associated with system memory transactions initiated by the processor.

9. The integrated circuit system controller of claim 8, further comprising an accelerated graphics port interface adapted to communicate with an accelerated graphics device.

10. The integrated circuit system controller of claim 9, wherein the memory controller further comprising a posted write buffer operatively coupled to the memory controller.

11. The integrated circuit system controller of claim 8, wherein the memory buffer is configured as a fully associative cache memory.

12. The integrated circuit system controller of claim 8, wherein the memory buffer is configured as a set associative cache memory.

13. The integrated circuit system controller of claim 12, wherein the set associative cache memory is configured as a 2-way set associative cache memory.

14. The integrated circuit system controller of claim 11, wherein the random access memory comprises dynamic random access memory.

15. The integrated circuit system controller of claim 14, wherein the dynamic random access memory comprises between approximately 1 and 4 megabytes.

16. A memory control method executed by a memory control device having one or more cache structures, the method comprising:

receiving a memory access request signal from a device;

identifying the device;

selecting a cache structure based on the identified device; and

using the selected cache structure to satisfy the memory access request.

17. The method of claim 16, wherein the act of identifying the device comprises determining if the device is a processor unit.

18. The method of claim 16, wherein the act of selecting a cache structure comprises:

selecting a first cache structure if the identified device is a processor unit, else selecting a second cache structure.

19. The method of claim 16, wherein the acts of selecting a cache structure and using the selected cache structure comprise:

selecting a cache structure if the identified device is a processor unit, else accessing a system memory to satisfy the memory request.

20. The method of claim 19, wherein the act of using the selected cache structure comprises:

satisfying the memory request from an entry in the selected cache structure if possible, else

accessing a system memory to satisfy the memory request.

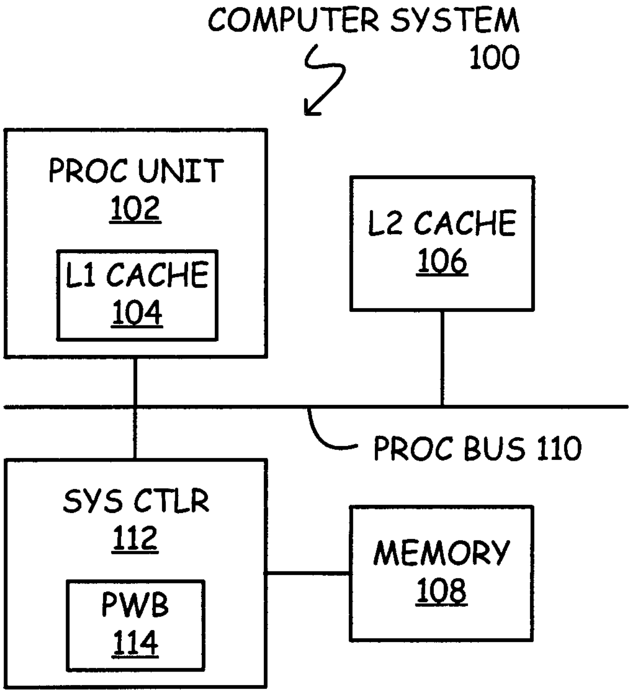


FIG. 1
(Prior Art)

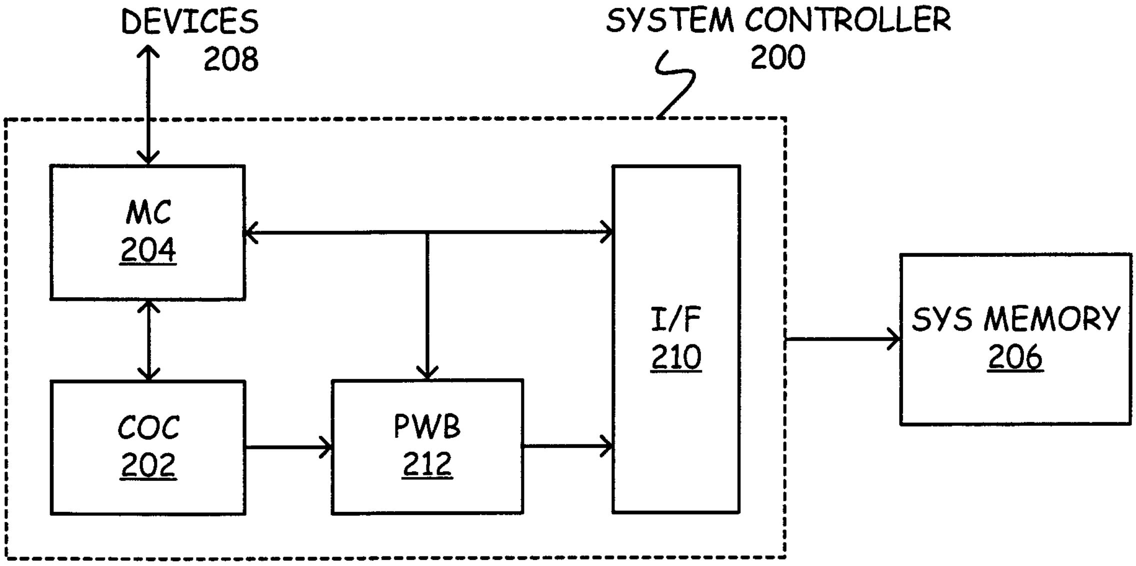


FIG. 2

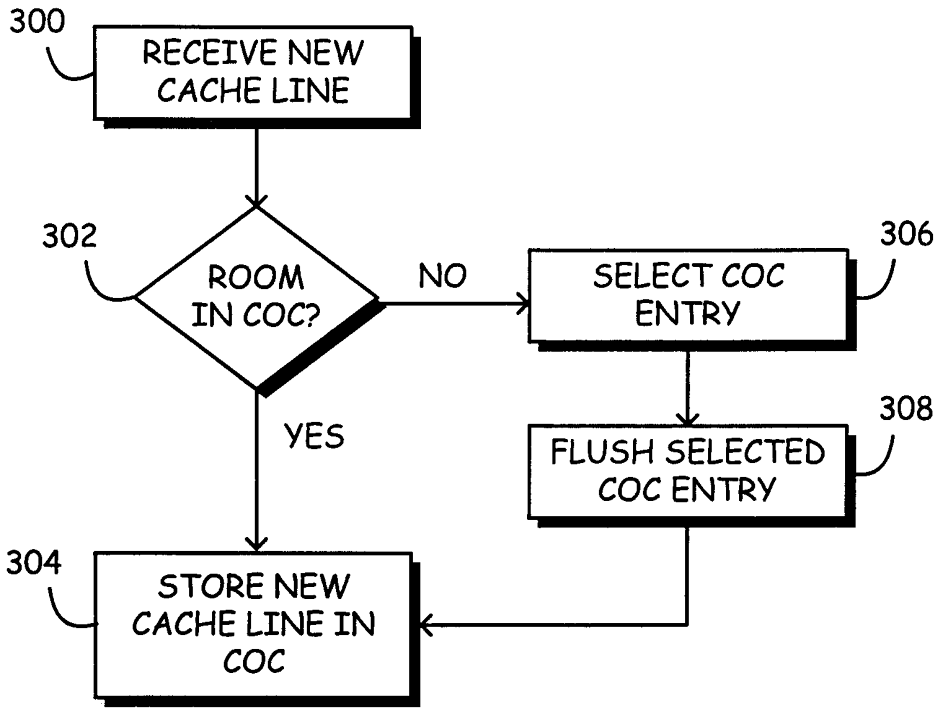

300
RECEIVE NEW CACHE LINE
302
ROOM IN COC?
NO
SELECT COC ENTRY
306
YES
FLUSH SELECTED COC ENTRY
308
304
STORE NEW CACHE LINE IN COC

FIG. 3

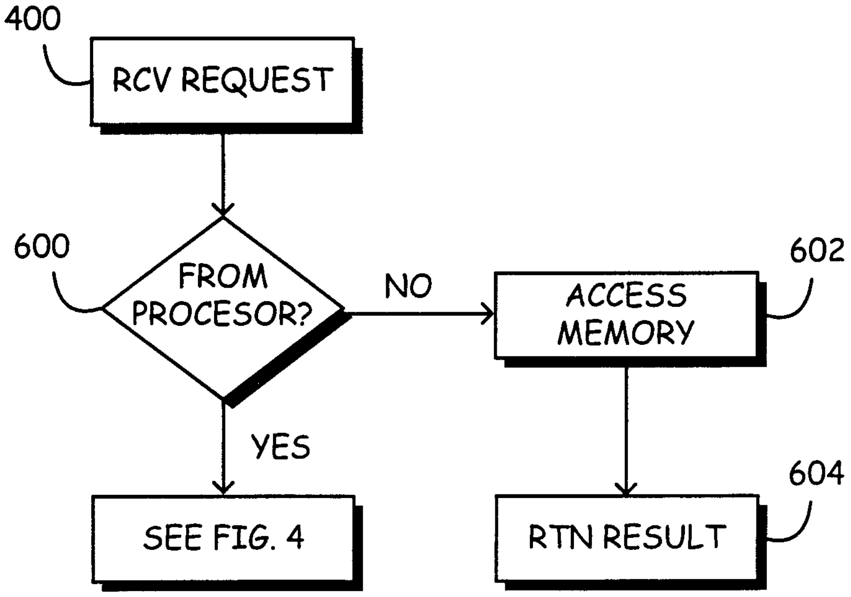

400
RCV REQUEST
600
FROM PROCESOR?
NO
ACCESS MEMORY
602
YES
SEE FIG. 4
604
RTN RESULT

FIG. 6

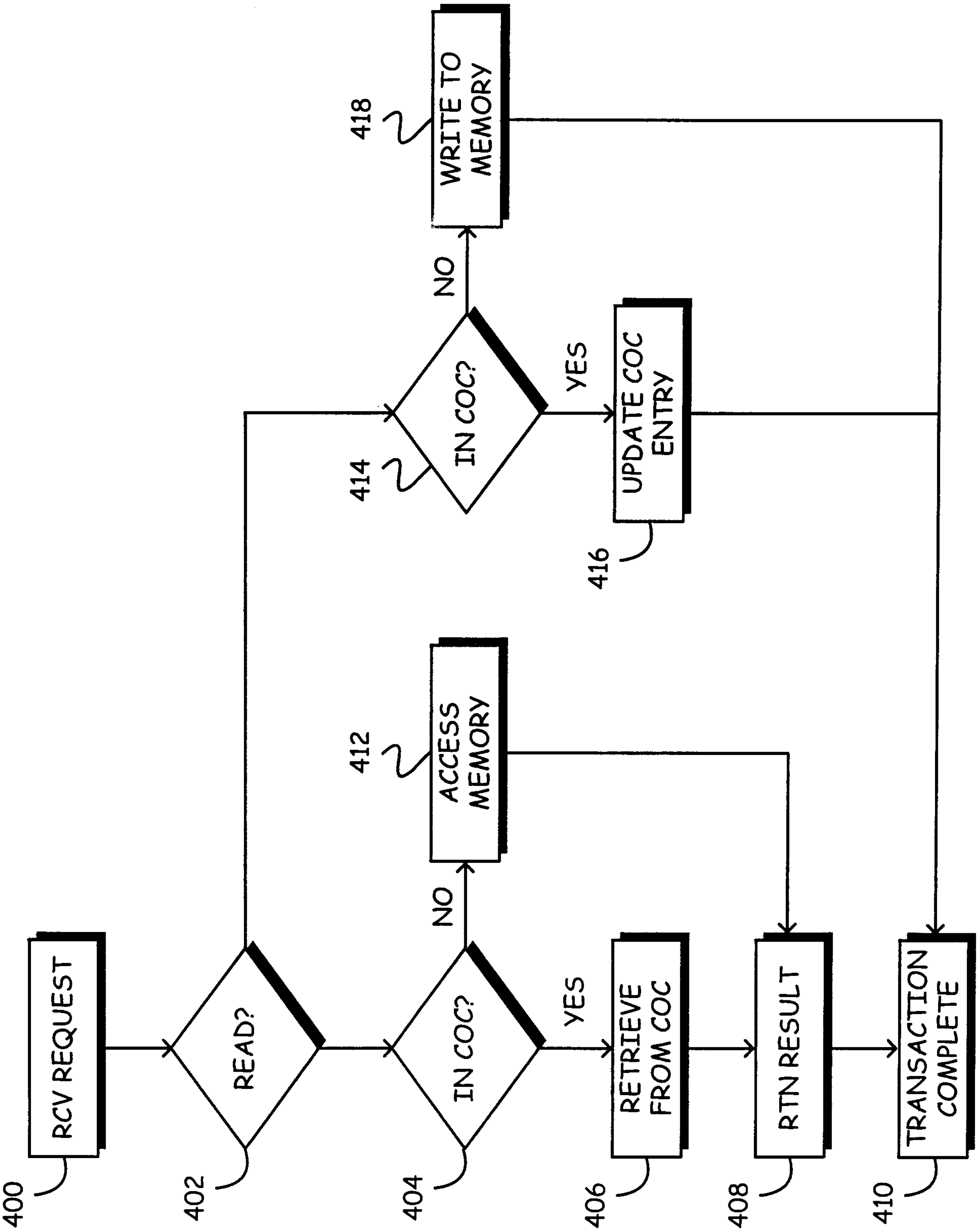

400
RCV REQUEST
402
READ?
404
IN COC?
NO
YES
412
ACCESS MEMORY
406
RETRIEVE FROM COC
408
RTN RESULT
410
TRANSACTION COMPLETE
414
IN COC?
NO
YES
418
WRITE TO MEMORY
416
UPDATE COC ENTRY

FIG. 4

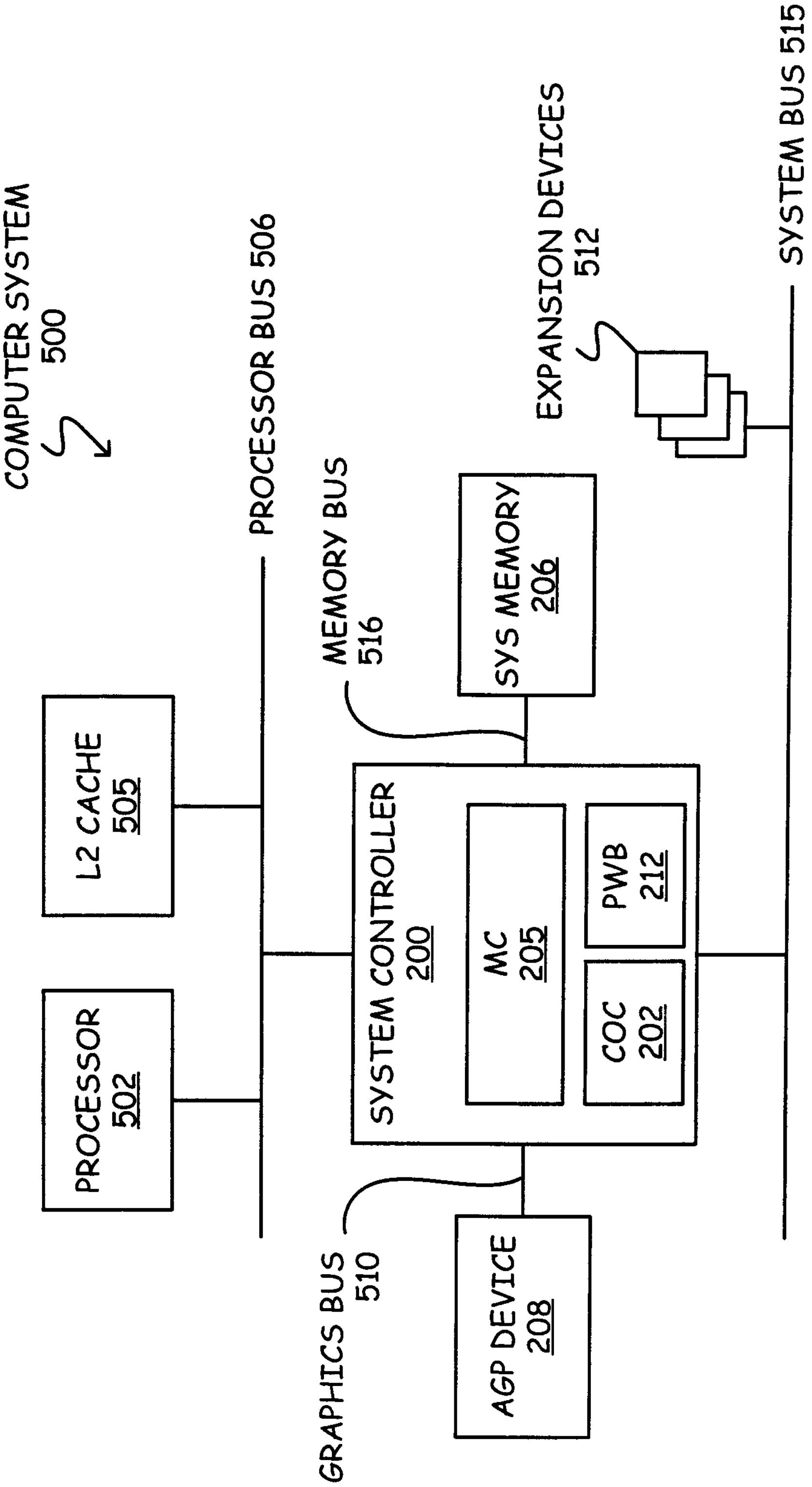

COMPUTER SYSTEM
500
PROCESSOR BUS 506
EXPANSION DEVICES
512
SYSTEM BUS 515
L2 CACHE
505
PROCESSOR
502
MEMORY BUS
516
SYS MEMORY
206
SYSTEM CONTROLLER
200
MC
205
PWB
212
COC
202
GRAPHICS BUS
510
AGP DEVICE
208

FIG. 5

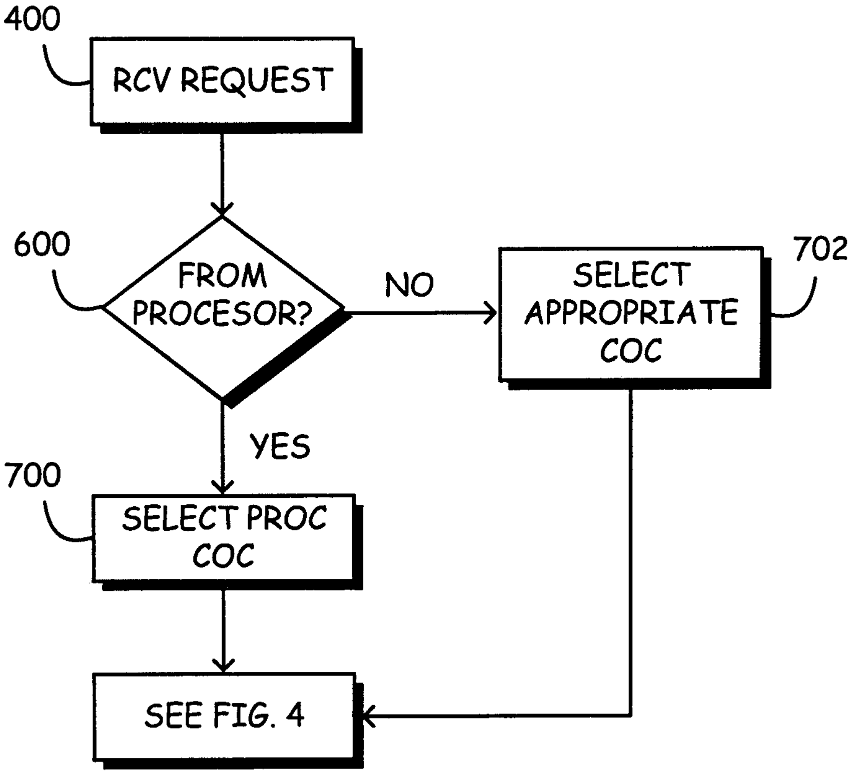

400
RCV REQUEST
600
FROM PROCESOR?
NO
SELECT APPROPRIATE COC
702
YES
700
SELECT PROC COC
SEE FIG. 4

FIG. 7

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/20496

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F12/08

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 470 736 A (NCR CO) 12 February 1992 (1992-02-12) the whole document ---	1-4, 8-13, 16-20
X	GB 2 215 887 A (INT COMPUTERS LTD) 27 September 1989 (1989-09-27) page 2, line 4 -page 3, line 5 ---	1-5,8, 11-15
X	EP 0 800 137 A (IBM) 8 October 1997 (1997-10-08) column 2, line 39 -column 3, line 5; figures 1,2 ---	1,2,6-8
X	EP 0 657 819 A (ADVANCED MICRO DEVICES INC) 14 June 1995 (1995-06-14) column 2, line 56 -column 3, line 18; figure 2 --- -/--	1-3,8

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 6 November 2000	Date of mailing of the international search report 14/11/2000
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Nielsen, O

INTERNATIONAL SEARCH REPORT

Interna' ıl Application No
PCT/US 00/20496

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 681 241 A (COMPAQ COMPUTER CORP) 8 November 1995 (1995-11-08) page 2, line 26 - line 54; figure 2 -----	1-5

1

INTERNATIONAL SEARCH REPORT

ı...ɾrmation on patent family members

Interna al Application No

PCT/uS 00/20496

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0470736 A	12-02-1992	US 5835945 A	10-11-1998
		DE 69131338 D	22-07-1999
		DE 69131338 T	25-11-1999
		JP 4250543 A	07-09-1992
GB 2215887 A	27-09-1989	AU 607594 B	07-03-1991
		AU 3122089 A	14-09-1989
		ZA 8900345 A	25-10-1989
EP 0800137 A	08-10-1997	US 5778422 A	07-07-1998
		JP 10055306 A	24-02-1998
EP 0657819 A	14-06-1995	US 5623627 A	22-04-1997
		JP 7200399 A	04-08-1995
EP 0681241 A	08-11-1995	US 5561779 A	01-10-1996
		CA 2148186 A	05-11-1995
		US 5829027 A	27-10-1998