



(19) **United States**

(12) **Patent Application Publication**
Li et al.

(10) **Pub. No.: US 2016/0147666 A1**
(43) **Pub. Date: May 26, 2016**

(54) **MULTILEVEL CACHE-BASED DATA READ/WRITE METHOD AND APPARATUS, AND COMPUTER SYSTEM**

(52) **U.S. Cl.**
CPC **G06F 12/0897** (2013.01); **G06F 3/0604** (2013.01); **G06F 3/064** (2013.01); **G06F 3/0683** (2013.01); **G06F 12/0811** (2013.01); **G06F 12/1027** (2013.01); **G06F 2212/50** (2013.01); **G06F 2212/60** (2013.01)

(71) Applicant: **Huawei Technologies Co., Ltd.**,
Shenzhen (CN)

(72) Inventors: **Ye Li**, Beijing (CN); **Lixin Zhang**,
Beijing (CN); **Rui Hou**, Beijing (CN);
Ke Zhang, Beijing (CN)

(57) **ABSTRACT**

(21) Appl. No.: **15/010,376**

(22) Filed: **Jan. 29, 2016**

Related U.S. Application Data

(63) Continuation of application No. PCT/CN2014/083467, filed on Jul. 31, 2014.

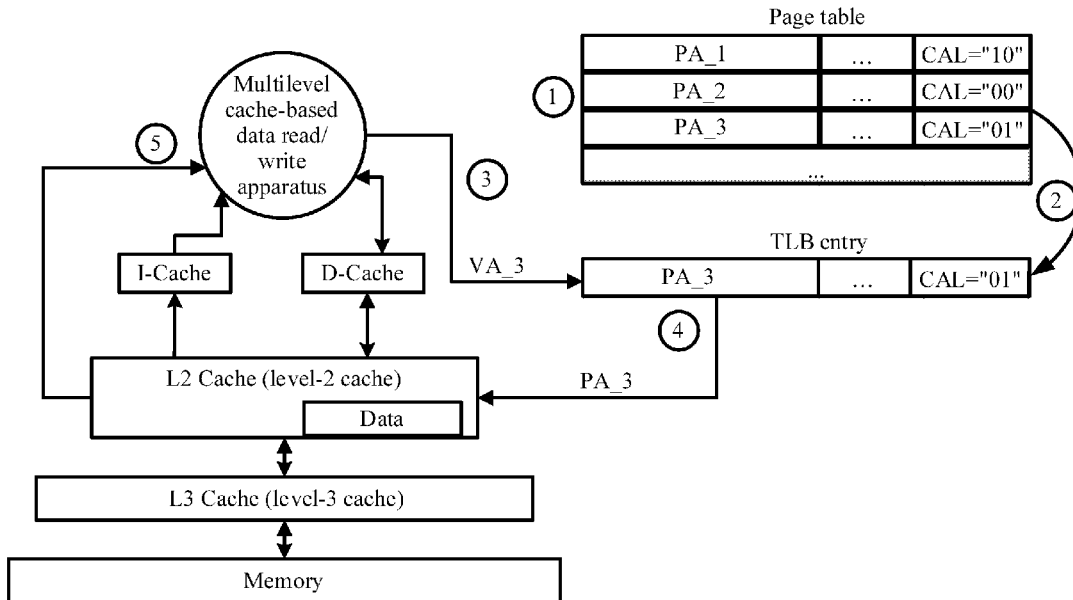
Foreign Application Priority Data

Jul. 31, 2013 (CN) 201310331344.6

Publication Classification

(51) **Int. Cl.**
G06F 12/08 (2006.01)
G06F 12/10 (2006.01)
G06F 3/06 (2006.01)

A multilevel cache-based data read/write method and a computer system. The method includes acquiring a query address of a physical memory data block in which data is to be read/written, acquiring a cache location attribute of the physical memory data block, querying whether a cache is hit until one cache is hit or all caches are missed, where the querying is performed according to the query address in descending order of levels of caches storable for the physical memory data block, and the levels of the caches are indicated by the cache location attribute, and if one cache is hit, reading/writing the data in the query address of the physical memory data block in the hit cache; or, if all caches are missed, reading/writing the data in the query address of the physical memory data block in a memory.



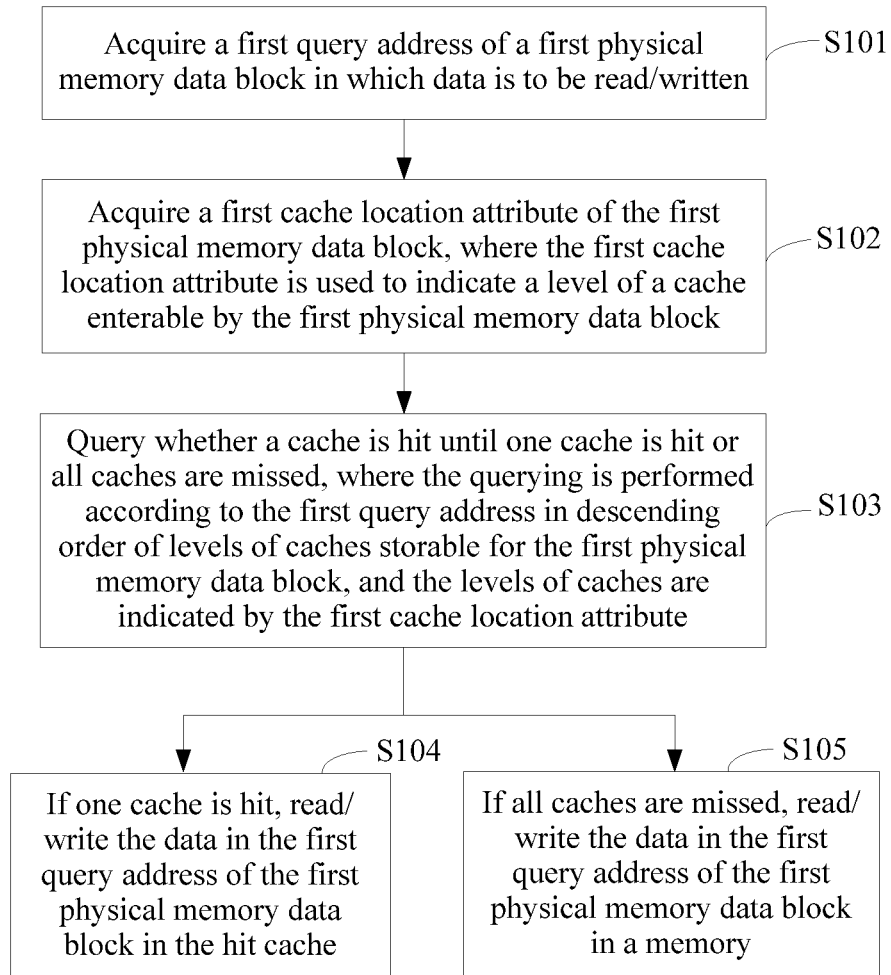


FIG. 1

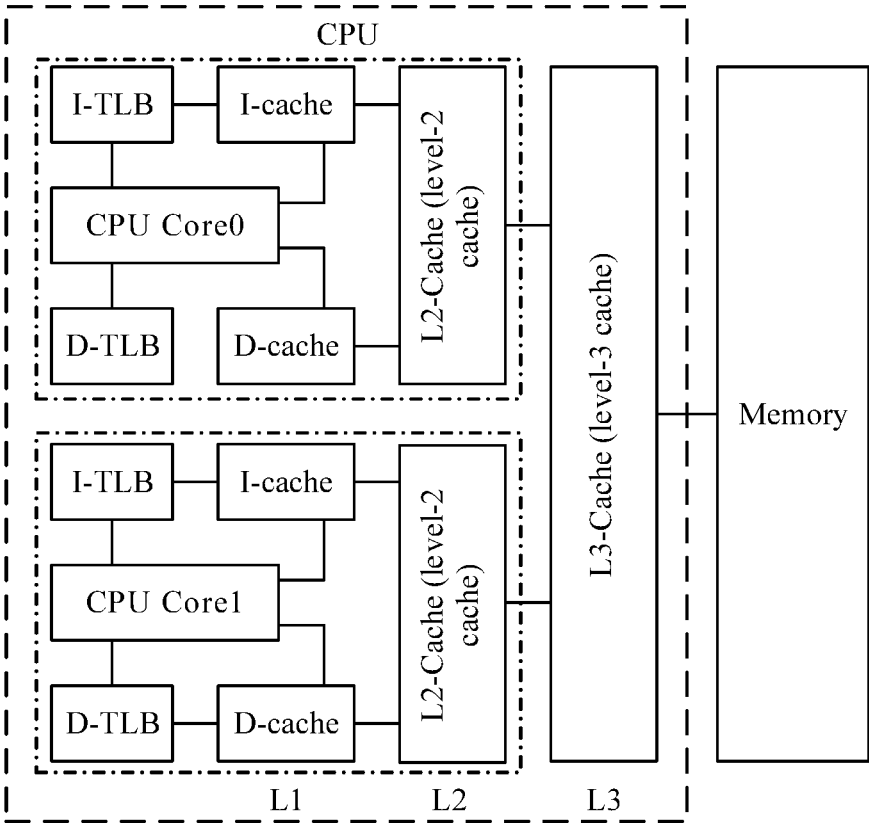


FIG. 2

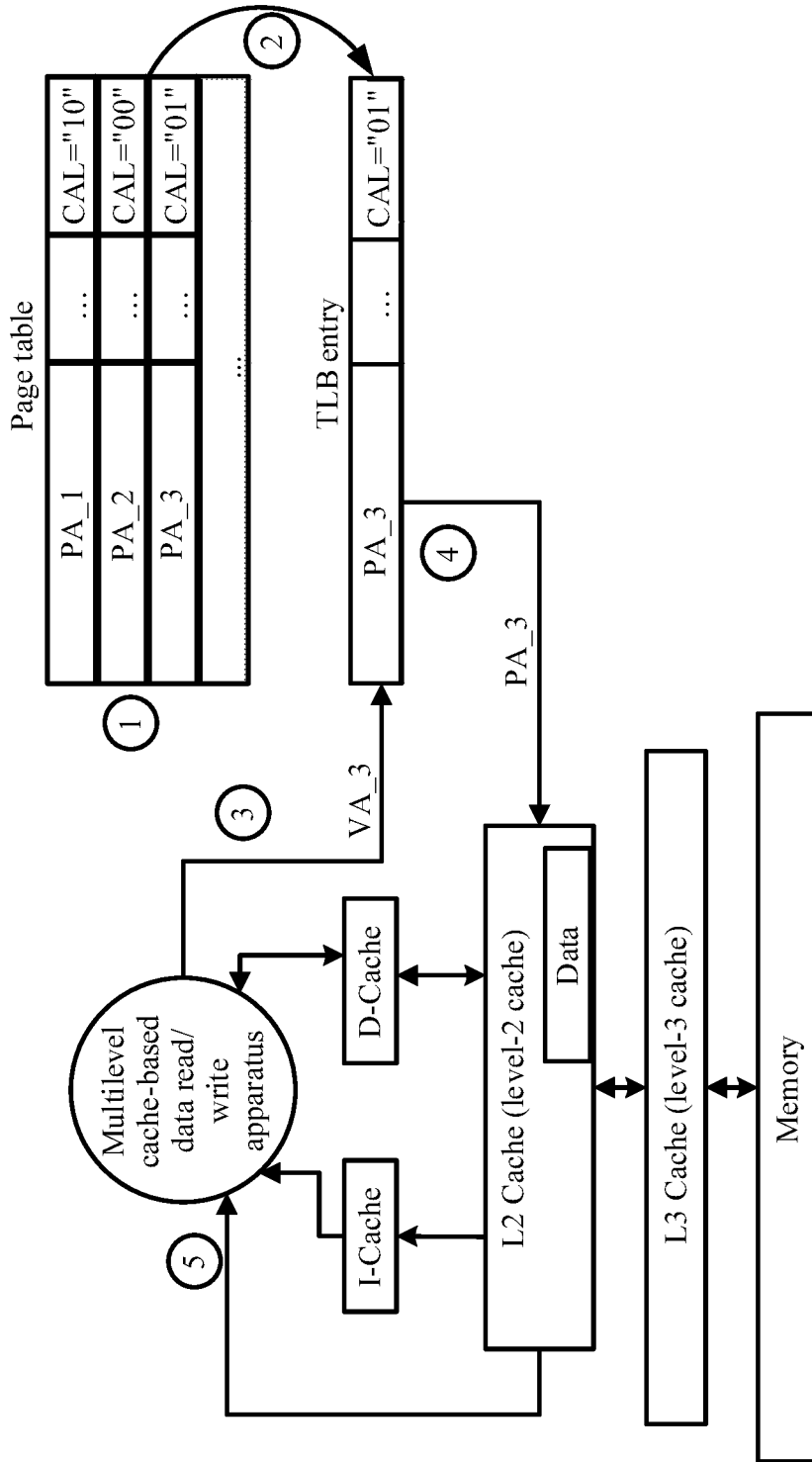


FIG. 3

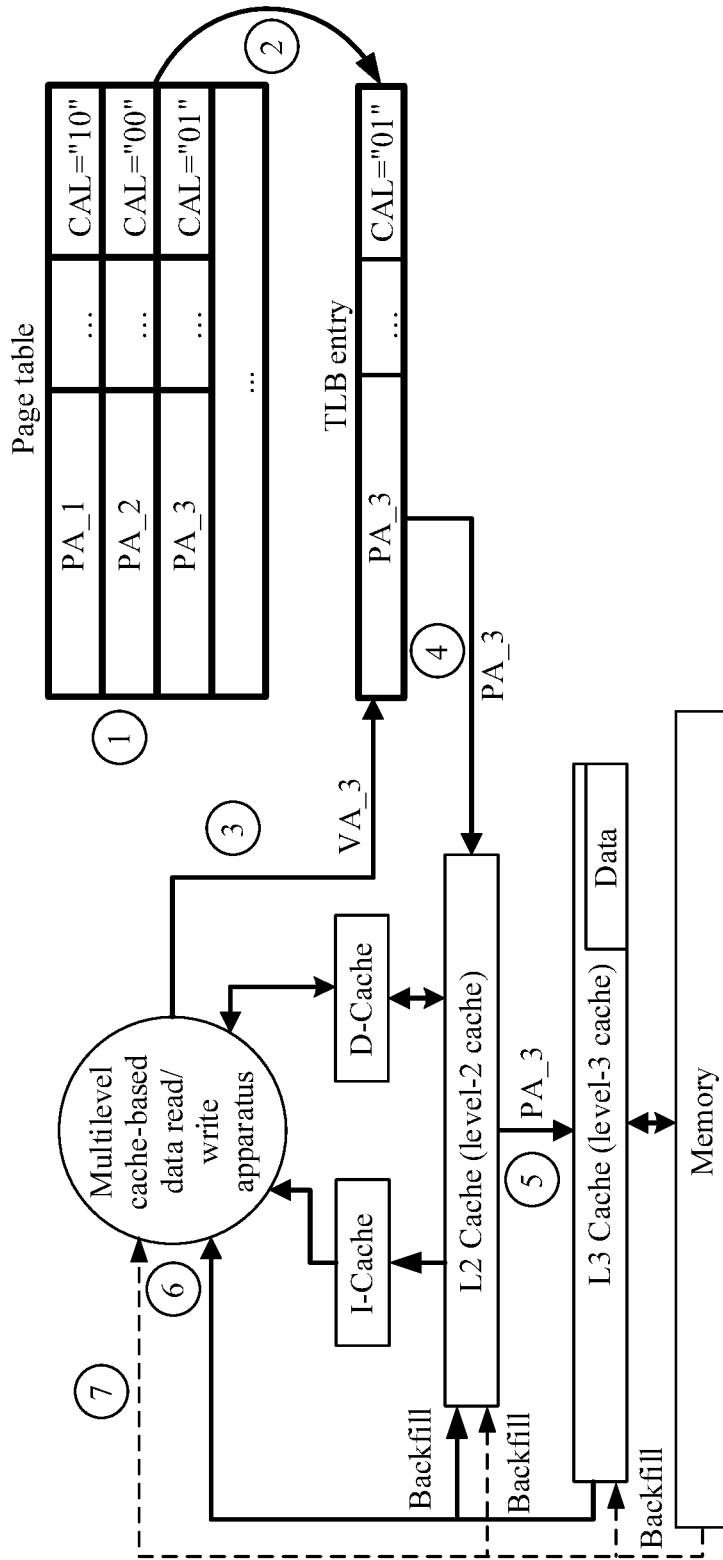


FIG. 4

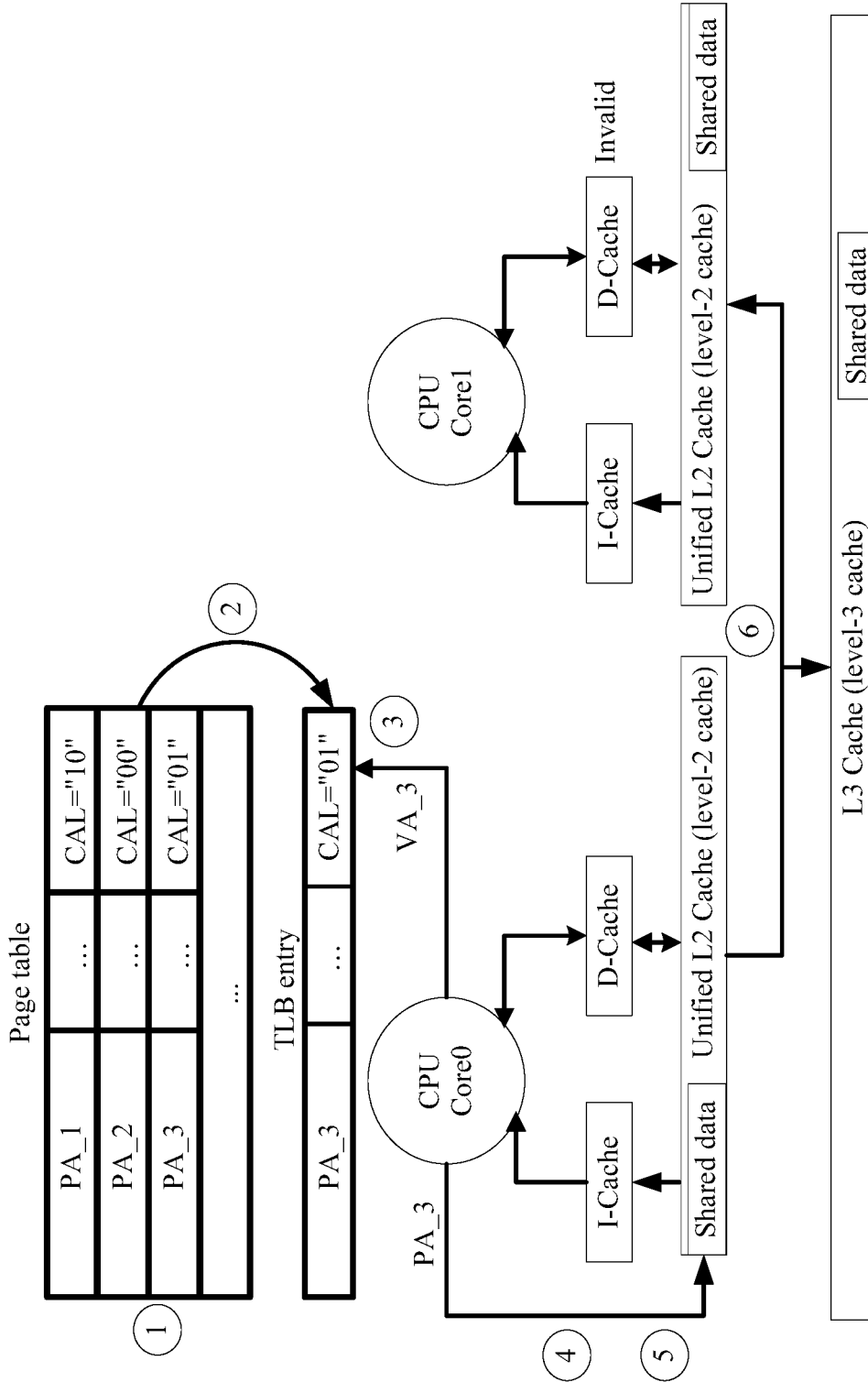


FIG. 5

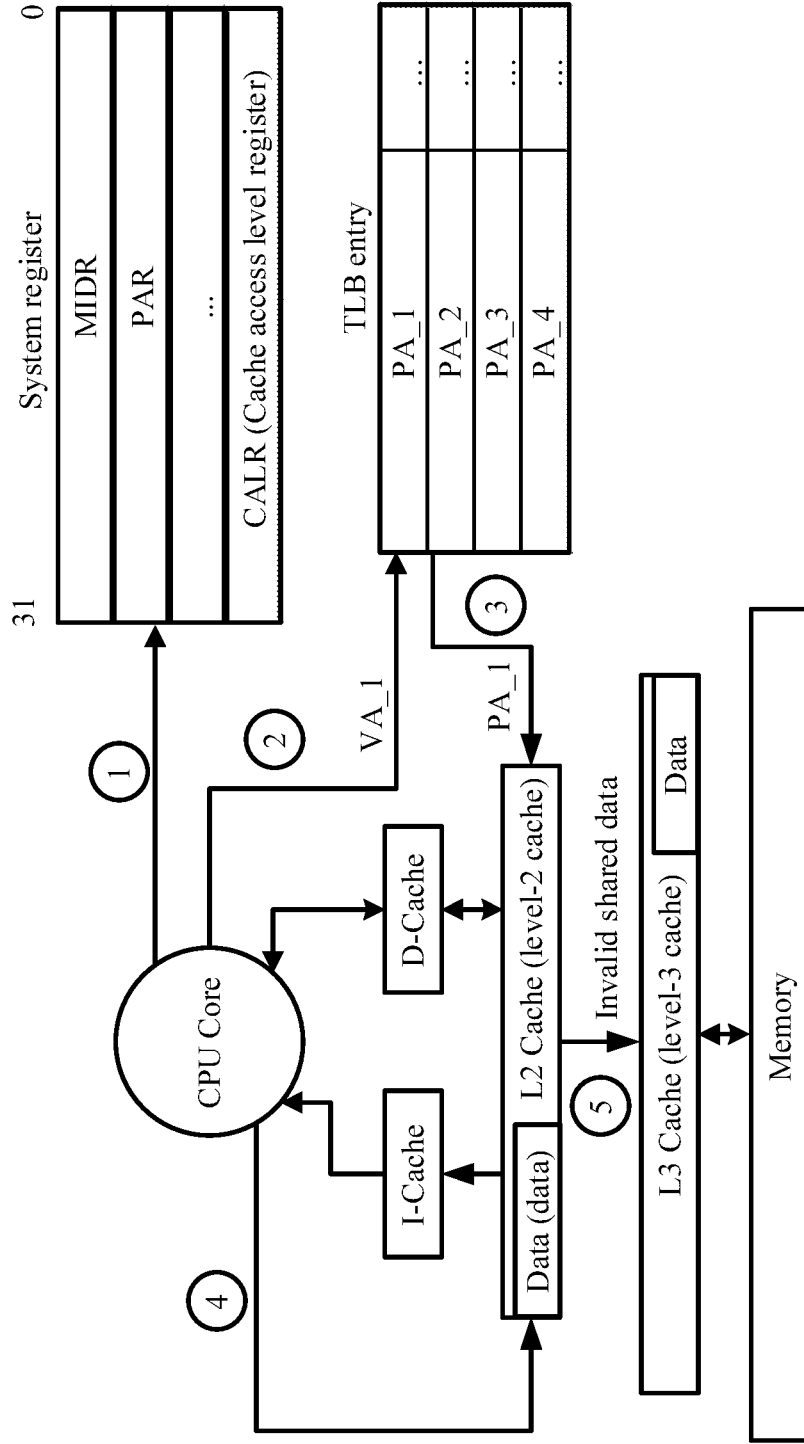


FIG. 6

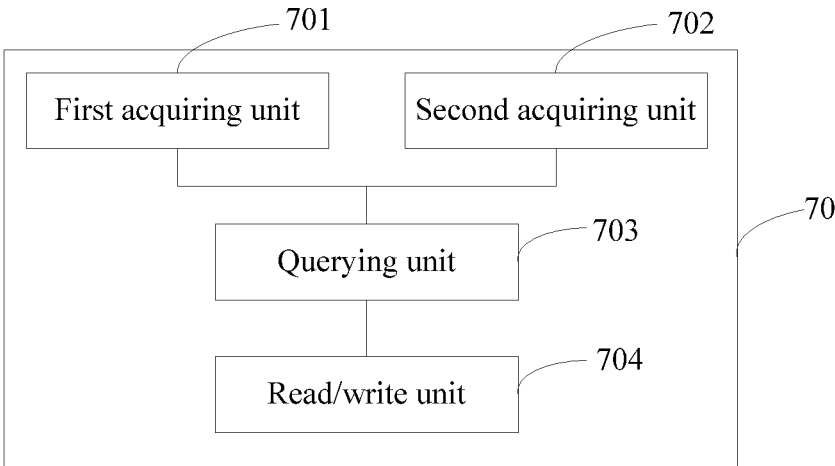


FIG. 7

**MULTILEVEL CACHE-BASED DATA
READ/WRITE METHOD AND APPARATUS,
AND COMPUTER SYSTEM**

**CROSS-REFERENCE TO RELATED
APPLICATIONS**

[0001] This application is a continuation of International Application No. PCT/CN2014/083467, filed on Jul. 31, 2014, which claims priority to Chinese Patent Application No. 201310331344.6, filed on Jul. 31, 2013, both of which are hereby incorporated by reference in their entireties.

TECHNICAL FIELD

[0002] The present disclosure relates to the field of data read/write of a computer system, and in particular, to a multilevel cache-based data read/write method and apparatus, and a computer system.

BACKGROUND

[0003] A cache (also referred to as a high-speed cache memory) is located between a central processing unit (CPU) and a memory and is primarily used to solve a problem of mismatch between a speed of CPU operation and a speed of memory read/write. Currently, a computer uses at most a three-level cache, where a high-speed small-capacity cache is used at level 1 cache (L1 cache), a medium-speed larger-capacity cache is used at level 2 cache (L2 cache), and a low-speed large-capacity cache is used at level 3 cache (L3 cache).

[0004] Based on the multi-level cache, a specific process of executing a read operation by a CPU in the prior art is as follows.

[0005] When a CPU core sends a data read request, a level-by-level search manner is applied. That is, data search begins in the L1 cache first; if there is no searching data in the L1 cache (cache miss), a search is performed in the L2 cache, and so on, until data is searched out.

[0006] In the foregoing process, the cache is generally hit after the search is performed multiple times. For example, assuming that data on which a read operation is to be performed is stored in the L3 cache, a data search is performed in the L3 cache only when both the L1 cache and the L2 cache are missed, which leads to low efficiency of cache access.

SUMMARY

[0007] Embodiments of the present disclosure provide a multilevel cache-based data read/write method and apparatus, and a computer system, which are conducive to improving cache access efficiency and therefore conducive to reducing a cache miss rate in a data read/write process.

[0008] To achieve the foregoing objective, the following technical solutions are provided in the embodiments of the present disclosure.

[0009] According to a first aspect, the present disclosure provides a multilevel cache-based data read/write method, including acquiring a first query address of a first physical memory data block in which data is to be read/written, acquiring a first cache location attribute of the first physical memory data block, where the first cache location attribute is used to indicate a level of a cache storable for the first physical memory data block, querying whether a cache is hit until one cache is hit or all caches are missed, where the querying is performed according to the first query address in descending

order of levels of caches storable for the first physical memory data block, and the levels of the caches are indicated by the first cache location attribute, and if one cache is hit, reading/writing the data in the first query address of the first physical memory data block in the hit cache, or if all caches are missed, reading/writing the data in the first query address of the first physical memory data block in a memory.

[0010] In a first possible implementation manner, according to the first aspect, the first cache location attribute is set in a translation lookaside buffer (TLB) for the first physical memory data block according to an access status of the first physical memory data block, and the acquiring a first cache location attribute of the first physical memory data block includes querying, according to attribute information that is of a physical memory data block and stored in the TLB, a first cache location attribute corresponding to the first query address, where the TLB stores the attribute information of the physical memory data block and the attribute information includes a correspondence between a query address of a physical memory data block and a cache location attribute of the physical memory data block.

[0011] In a second possible implementation manner, according to the first aspect, the first cache location attribute is set in a system register for the first physical memory data block according to an access status of the first physical memory data block, and the acquiring a first cache location attribute of the first physical memory data block includes reading a cache location attribute stored in a current system register, and using the cache location attribute as the first cache location attribute of the first physical memory data block, where the system register is used to store the cache location attribute.

[0012] In a third possible implementation manner, according to the first aspect, the reading/writing the data in the first query address of the first physical memory data block in the hit cache includes reading/writing the data in the first query address of the first physical memory data block in the hit cache by using a direct data path to the hit cache, or the reading/writing the data in the first query address of the first physical memory data block in a memory includes reading/writing the data in the first query address of the first physical memory data block in the memory using a direct data path to the memory.

[0013] In a fourth possible implementation manner, with reference to the first aspect or any of the first three possible implementation manners, during data reading, if a level of the hit cache is a non-highest level in the levels, of the enterable caches, indicated by the first cache location attribute, the method further includes refilling, with to-be-read data, a cache whose level is higher than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute, or if a level of the hit cache is a highest level in the levels, of the enterable caches, indicated by the first cache location attribute, skip refilling another cache with data, or if all caches are missed, the method further includes refilling, with to-be-read data, an enterable cache indicated by the first cache location attribute.

[0014] In a fifth possible implementation manner, with reference to the first aspect or any of the first three possible implementation manners, during data writing, after the writing the data in the first query address of the first physical memory data block in the hit cache, the method further includes, if the written data is shared data, initiating a shared data consistency request to another cache whose level is equal

to and/or lower than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute.

[0015] In a sixth possible implementation manner, with reference to the first aspect or any of the first five possible implementation manners, the first cache location attribute identifies the highest level among the levels of the caches storable for the first physical memory data block, or identifies a level of a cache that is not storable for the first physical memory data block.

[0016] According to a second aspect, the present disclosure provides a multilevel cache-based data read/write apparatus, where the apparatus is a central processing unit that includes multiple levels of caches, and the apparatus includes a first acquiring unit configured to acquire a first query address of a first physical memory data block in which data is to be read/written a second acquiring unit configured to acquire a first cache location attribute of the first physical memory data block, where the first cache location attribute is used to indicate a level of a cache storable for the first physical memory data block a querying unit configured to query whether a cache is hit until one cache is hit or all caches are missed, where the querying is performed according to the first query address obtained by the first acquiring unit and is performed in descending order of levels of caches storable for the first physical memory data block, and the levels of the caches are indicated by the first cache location attribute obtained by the second acquiring unit; and a read/write unit configured to if one cache is hit, read/write the data in the first query address of the first physical memory data block in the cache hit by the querying unit, or if a query result of the querying unit is that all caches are missed, read/write the data in the first query address of the first physical memory data block in a memory.

[0017] In a first possible implementation manner, according to the second aspect, the first cache location attribute is set in a TLB for the first physical memory data block according to an access status of the first physical memory data block, the second acquiring unit is configured to query, according to attribute information that is of a physical memory data block and stored in the TLB, a first cache location attribute corresponding to the first query address, where the TLB stores the attribute information of the physical memory data block and the attribute information includes a correspondence between a query address of a physical memory data block and a cache location attribute of the physical memory data block.

[0018] In a second possible implementation manner, according to the second aspect, the first cache location attribute is set in a system register for the first physical memory data block according to an access status of the first physical memory data block, and the second acquiring unit is configured to read a cache location attribute stored in a current system register, and use the cache location attribute as the first cache location attribute of the first physical memory data block, where the system register is used to store the cache location attribute.

[0019] In a third possible implementation manner, according to the second aspect, the read/write unit is configured to read/write the data in the first query address of the first physical memory data block in the hit cache using a direct data path to the hit cache, or the read/write unit is configured to read/write the data in the first query address of the first physical memory data block in the memory using a direct data path to the memory.

[0020] In a fourth possible implementation manner, with reference to the second aspect or any of the first three possible implementation manners, the read/write unit is further configured to, during data reading, if a level of the hit cache is a non-highest level in the levels, of the enterable caches, indicated by the first cache location attribute, refill, with to-be-read data, a cache whose level is higher than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute, or if a level of the hit cache is a highest level in the levels, of the enterable caches, indicated by the first cache location attribute, skip refilling another cache with data, or if all caches are missed, refill, with to-be-read data, an enterable cache indicated by the first cache location attribute.

[0021] In a fifth possible implementation manner, with reference to the second aspect or any of the first three possible implementation manners, the read/write unit is further configured to, during data writing, if the written data is shared data, initiate a shared data consistency request to another cache whose level is equal to and/or lower than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute.

[0022] In a sixth possible implementation manner, with reference to the second aspect or any of the first five possible implementation manners, the first cache location attribute identifies the highest level among the levels of the caches storable for the first physical memory data block, or identifies a level of a cache that is not storable for the first physical memory data block.

[0023] According to a third aspect, the present disclosure provides a computer system, including a memory and the foregoing apparatus.

[0024] According to the multilevel cache-based data read/write method and apparatus and the computer system provided in the embodiments of the present disclosure, a cache location attribute indicates a level of a cache storable for a physical memory data block, and whether a cache is hit is queried according to the enterable level indicated by the cache location attribute. Therefore, it is supported that a cache that does not need to be accessed can be skipped, and a cache is hit quickly, which is conducive to improving cache access efficiency and therefore conducive to enhancing a cache hit rate, and in other words, conducive to reducing a cache miss rate in a data read/write process. Specially, by restricting a cache location attribute, an infrequently accessed physical memory data block can be prevented from entering a high-level cache with insufficient resources, which is therefore conducive to reducing the cache miss rate in the data read/write process.

BRIEF DESCRIPTION OF DRAWINGS

[0025] To describe the technical solutions in the embodiments of the present disclosure more clearly, the following briefly introduces the accompanying drawings required for describing the embodiments.

[0026] FIG. 1 is a schematic diagram of a multilevel cache-based data read/write method according to an embodiment of the present disclosure;

[0027] FIG. 2 is a schematic structural diagram of a computer system according to an embodiment of the present disclosure;

[0028] FIG. 3 is a schematic diagram of a multilevel cache-based data read operation method according to an embodiment of the present disclosure;

[0029] FIG. 4 is a schematic diagram of another multilevel cache-based data read operation method according to an embodiment of the present disclosure;

[0030] FIG. 5 is a schematic diagram of a multilevel cache-based data write operation method according to an embodiment of the present disclosure;

[0031] FIG. 6 is a schematic diagram of another multilevel cache-based data write operation method according to an embodiment of the present disclosure; and

[0032] FIG. 7 is a schematic structural diagram of a multilevel cache-based data read/write operation apparatus according to an embodiment of the present disclosure.

DESCRIPTION OF EMBODIMENTS

[0033] The following clearly describes the technical solutions in the embodiments of the present disclosure with reference to the accompanying drawings in the embodiments of the present disclosure.

[0034] The present disclosure provides a multilevel cache-based data read/write method, and the method may be executed by a CPU core in a CPU that includes multiple levels of caches or by a hardware device with a read/write function and in a CPU that includes multiple levels of caches. The method is applicable to a single-core computer system and also applicable to a multi-core computer system. As shown in FIG. 1, the method includes:

[0035] Step S101. Acquire a first query address of a first physical memory data block in which data is to be read/written.

[0036] The physical memory data block refers to one of several storage units that are obtained by dividing a memory and used to store data in a computer system. For example, a storage unit may be a memory page, a memory segment, or data information that directly comes from input/output (I/O) or the like. Sizes of the storage units are not limited and may be the same or different. The query address may be a high-order bit of a physical address (PA) of the physical memory data block, which, certainly, is not limited thereto.

[0037] For clear description, in this embodiment of the present disclosure, a related object in a data read/write process is preceded by "first". For example, the first physical memory data block refers to a physical memory data block in a data read/write process, and the first query address refers to the high-order bit of the physical address of the first physical memory data block.

[0038] For example, the CPU core may include a TLB; and this step may be: obtain, from the TLB, the high-order bit of the physical address in a data read/write process. However, the TLB is not necessarily set in the CPU core.

[0039] When a multilevel buffer-based data read/write apparatus in the CPU core sends a data read/write request, the apparatus gives a virtual address (VA), and uses several high-order bits of the VA to search the TLB, and the TLB is responsible for translating the virtual address into the high-order bit of the physical address of the physical memory data block, such that the apparatus obtains the high-order bit of the physical address.

[0040] Step S102. Acquire a first cache location attribute of the first physical memory data block, where the first cache location attribute is used to identify a level of a cache storable for the first physical memory data block.

[0041] The first cache location attribute is set for the first physical memory data block according to an access status of the first physical memory data block.

[0042] Generally, a frequently accessed physical memory data block exists in a high-level cache, and an infrequently accessed physical memory data block exists in a low-level cache. In order to set the cache location attribute properly, the cache location attribute may be set for the physical memory data block according to the access status of the physical memory data block. The access status of the physical memory data block may be but is not limited to access frequency of the physical memory data block. The cache location attribute may also be set according to an attribute of the data block itself such as streaming data. To a great extent, such data may be discarded once the data is accessed for one time. Therefore, such data also does not need to enter a high-level cache.

[0043] Optionally, the memory generally stores attribute information of all physical memory data blocks. In this embodiment, several bits may be added in existing attribute information of a physical memory data block in the memory to identify a cache location attribute of each physical memory data block. In this case, the attribute information of the physical memory data block includes a correspondence between a query address of a physical memory data block and a cache location attribute of the physical memory data block, where the cache location attribute may be directly set by an operating system according to an access status of the physical memory data block during data block division. In addition, the memory may fill space of the TLB with the attribute information into which the cache location attribute is added, where the space is used to store an attribute item. Optionally, if a reserved bit (a bit that is not occupied) exists in original space of the TLB and is available for being filled with the cache location attribute, the space can be filled directly. Otherwise, width of the original space may be increased for being filled with the cache location attribute.

[0044] In this case, this step may be, when sending a data read/write request, the multilevel cache-based data read/write apparatus in the CPU core queries, according to the correspondence between the query address of the physical memory data block and the attribute information of the physical memory data block, a first cache location attribute corresponding to the first query address, where the correspondence is stored in the TLB.

[0045] Alternatively, optionally, in this embodiment of the present disclosure, the cache location attribute may be set to a system register included by the CPU. That is, a system register is added for storing the cache location attribute. A new instruction is added or some bits are added in an existing instruction set to specially set the register of the cache location attribute.

[0046] In this case, this step may be, when sending a data read/write request, the multilevel cache-based data read/write apparatus in the CPU core reads a cache location attribute stored in a current system register, and uses the cache location attribute as the first cache location attribute of the first physical memory data block.

[0047] Further, the first cache location attribute identifies a highest level among levels of caches storable for the first physical memory data block, or identifies a level of a cache that is not storable for the first physical memory data block.

[0048] For example, assuming that there are three levels of caches in total in a computer system, and for differentiation, 00, 01, and 10 may be used to represent a L1 cache, a L2 cache, and a L3 cache respectively. If the cache location attribute is 01, it indicates that the memory data block can enter the L2 cache at most, that is, the memory data block may

enter the L2 cache or the L3 cache, but cannot enter the L1 cache. If the cache location attribute is 01, it indicates that the memory data block cannot enter the L2 cache, and correspondingly, the cache location attribute controls that the levels of the caches storable for the physical memory data block include the L1 cache and the L3 cache.

[0049] Because a one-to-one correspondence exists between the first physical memory data block and its first cache location attribute, for step S101 and step S102, the first cache location attribute of the first physical memory data block in which the data is to be read/written may be acquired first, and then the first query address of the first physical memory data block is acquired. Regardless of a manner of acquiring, the first query address and the first cache location attribute of the first physical memory data block need to be acquired.

[0050] Step S103. Query whether a cache is hit until one cache is hit or all caches are missed, where the querying is performed according to the first query address in descending order of levels of caches storable for the first physical memory data block, and the levels of caches are indicated by the first cache location attribute.

[0051] For a computer system with N levels of caches, the level-1 cache is the highest level, and the level-N cache is the lowest level. From the level-1 cache to the level-N cache, a cache level descends.

[0052] A process of querying, according to the first query address, whether a level-i ($1 \leq i \leq N$) cache is hit, may be searching the level-i cache by using several bits in the middle of the virtual address, and selecting a group of cache line tags (Tag), where the Tags may be high-order bits of the physical address (PA) and then comparing the first query address obtained from the TLB with the selected group of Tags, if a Tag of a cache line matches a high-order bit of the PA, the level-i cache is hit. Otherwise, the level-i cache is missed.

[0053] For example, it is assumed that there are three levels of caches in total in the computer system, the cache location attribute acquired in step S102 is 01, and it is indicated that the memory data block can enter the L2 cache at most. In this case, the multilevel cache-based data read/write apparatus in the CPU core ignores the L1 cache according to the cache location attribute, and queries directly whether the L2 cache is hit, and if the L2 cache is hit, the query process is ended, or if the L2 cache is missed, queries whether the L3 cache is hit, and if the L3 cache is hit, the query process is ended, or if the L3 cache is missed, because the L3 cache is a last level of cache, it means that all enterable caches under control of the first cache location attribute are missed, and the query process is also ended.

[0054] For another example, it is assumed that there are three levels of caches in total in the computer system, the cache location attribute acquired in step S102 is 01, and it is indicated that the memory data block cannot enter the L2 cache. Correspondingly, the cache location attribute controls that the levels of the caches storable for the physical memory data block include L1 and L3. In this case, the multilevel cache-based data read/write apparatus in the CPU core first queries, according to the cache location attribute, whether the L1 cache is hit, and if the L1 cache is hit, the query process is ended, or if the L1 cache is missed, queries whether the L3 cache is hit, and because the L3 cache is the last level of cache, the query process is ended no matter whether the L3 cache is hit or missed.

[0055] If one cache is hit in this step, step S104 is performed, and if all caches are missed in this step, step S105 is performed.

[0056] Step S104. If one cache is hit, read/write the data in the first query address of the first physical memory data block in the hit cache.

[0057] Preferably, a direct data path from each level of cache to the multilevel cache-based data read/write apparatus in the CPU core may be added in advance. In this case, this step may be read/write the data in the first query address of the first physical memory data block in the hit cache using a direct data path between the apparatus and the hit cache.

[0058] Certainly, the data may also be read/written in the first query address of the first physical memory data block in the hit cache using a data path established in the prior art.

[0059] Step S105. If all caches are missed, read/write the data in the first query address of the first physical memory data block in a memory.

[0060] Because the first physical memory data block can certainly be found in the memory, in a case that all caches are missed, this step may be used to implement data reading/writing.

[0061] According to the multilevel cache-based data read/write method provided in this embodiment of the present disclosure, a cache location attribute indicates a level of a cache storable for a physical memory data block, and whether a cache is hit is queried according to the enterable level indicated by the cache location attribute. Therefore, it is supported that a cache that does not need to be accessed can be skipped, and a cache is hit quickly, which is conducive to improving cache access efficiency and therefore conducive to enhancing a cache hit rate, and in other words, conducive to reducing a cache miss rate in a data read/write process. By restricting a cache location attribute, an infrequently accessed physical memory data block can be prevented from entering a high-level cache with insufficient resources, which is therefore conducive to reducing the cache miss rate in the data read/write process.

[0062] Further, during data reading, if a level of the hit cache is a non-highest level in the levels, of the enterable caches, indicated by the first cache location attribute, after the data is read in the first physical memory data block in the hit cache in step S104, the method further includes refilling, with to-be-read data, a cache whose level is higher than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute, or if a level of the hit cache is a highest level in the levels, of the enterable caches, indicated by the first cache location attribute, skipping refilling another cache with data, or if all caches are missed, the method further includes, refilling, with to-be-read data, an enterable cache indicated by the first cache location attribute.

[0063] For example, it is assumed that there are three levels of caches in total in the computer system, the cache location attribute acquired in step S102 is 01, and it is indicated that the memory data block can enter the L2 cache at most. That is, the L2 Cache is the highest enterable level indicated by the first cache location attribute.

[0064] If the hit cache in step S103 is L2 cache, because the L2 cache is the highest enterable level indicated by the first cache location attribute, another cache (for example, the L1 cache) is not refilled with data.

[0065] If the hit cache in step S103 is L3 cache, after and while the data is read in the first physical memory data block in the L3 cache in step S104, the multilevel cache-based data

read/write apparatus in the CPU core refills the L2 cache with the to-be-read data according to the enterable level indicated by the first cache location attribute. In this way, the L2 cache can be hit at a single attempt when the data is read/written in the first physical memory data block again.

[0066] If all caches are missed in step S103, after and while the data is read in the first physical memory data block in the memory in step S105, the multilevel cache-based data read/write apparatus in the CPU core refills the L3 cache and the L2 cache with the to-be-read data. In this way, the L2 cache can be hit at a single attempt of query in a process of reading the data for a next time.

[0067] By restricting the cache location level properly for the physical memory data block, for an application that leads to a high cache miss rate, infrequently accessed data can be prevented from being put into a cache with a high miss rate, so as to avoid futile cache data refill.

[0068] Further, to maintain consistency of shared data, during data writing, after the data is written in the first query address of the first physical memory data block in the hit cache in step S104, the method further includes, if the written data is shared data, initiating a shared data consistency request to another cache whose level is equal to and/or lower than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute.

[0069] The shared data is that, in a cache structure of a multi-core processor, if one of CPU cores modifies a piece of data and the data also exists in a cache of another CPU core, the data is shared data between different CPU cores.

[0070] For example, assuming that two CPU cores (CPU core 0 and CPU core 1) exist in a computer system and each CPU core includes three levels of caches, if the CPU core 0 modifies data in an L1 cache among its caches and the data also exists in an L1 cache of the CPU core 1, the data is shared data of the CPU core 0 and the CPU core 1.

[0071] For example, it is assumed that there are three levels of caches in total in the computer system, the cache location attribute acquired in step S102 is 01, and it is indicated that the memory data block can enter the L2 cache at most. That is, the levels, of the enterable caches, indicated by the first cache location attribute include L2 and L3.

[0072] If the hit cache in step S103 is L2 cache and the written data in step S104 is shared data, (1) in a case in which the computer system includes not only the hit L2 cache but also another L2 cache, the multilevel cache-based data read/write apparatus in the CPU core initiates a shared data consistency request to the another L2 cache and the L3 cache; and (2) in a case in which the computer system does not include another L2 cache, the apparatus initiates a shared data consistency request only to the L3 cache.

[0073] If the hit cache in step S103 is L3 cache and the written data in step S104 is shared data, (1) in a case in which the computer system includes not only the hit L3 cache but also another L3 cache, the multilevel cache-based data read/write apparatus in the CPU core initiates a shared data consistency request only to the another L3 cache; and (2) in a case in which the computer system does not include another L3 cache, the apparatus does not initiate a shared data consistency request.

[0074] Shared data consistency is maintained primarily according to the following two manners.

[0075] One is a write-back manner. After new data is written into the hit cache, and after a shared data consistency request is initiated, according to the foregoing description, to

another cache whose level is equal to and/or lower than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute, shared data in the another cache that contains the shared data is set to be invalid.

[0076] For example, if the hit cache in step S103 is L2 cache and the written data in step S104 is shared data, the multilevel cache-based data read/write apparatus in the CPU core initiates a shared data consistency request to another L2 cache and the L3 cache. In this case, shared data in the another L2 cache and the L3 cache are set to be invalid.

[0077] Another is a write-through manner. After a shared data consistency request is initiated, according to the foregoing description, to another cache whose level is equal to and/or lower than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute, shared data in the another cache whose level is equal to the level of the hit cache is set to be invalid. If the level of the hit cache is not a lowest level among the levels, of the enterable caches, indicated by the first cache location attribute, shared data in a lower-level cache is updated by newly written data.

[0078] For example, if the hit cache in step S103 is L2 cache and the written data in step S104 is shared data, the multilevel cache-based data read/write apparatus in the CPU core initiates a shared data consistency request to another L2 cache and the L3 cache. In this case, the shared data in the another L2 cache is set to be invalid, and the shared data in the L3 cache is updated.

[0079] Therefore, in a multi-core system environment, by setting a cache location attribute for a physical memory data block, a consistency request can be initiated pertinently to a cache that may have shared data, so as to reduce search and snooping for the cache.

[0080] An embodiment of the present disclosure provides a computer system, including a memory and a multilevel cache-based data read/write apparatus, where the apparatus is a CPU that includes multiple levels of caches, and the apparatus is used to perform steps in the foregoing method.

[0081] The present disclosure further provides specific instances to describe in detail the foregoing data read/write method. In each specific instance, a dual-core computer system is used as an example, and the dual-core computer system includes three levels of caches. Referring to FIG. 2, the computer system includes three levels of caches, an L1 cache that includes an instruction-cache (I-cache), a data-cache (D-cache), an L2 cache, and an L3 cache; a memory, and a CPU core.

[0082] The computer system may further include TLBs: an instruction-TLB (I-TLB), and a data-TLB (D-TLB). The TLB is responsible for translating a virtual address sent by the CPU core into a high-order bit of a physical address of a physical memory data block.

[0083] Furthermore, the L1 is a high-speed small-capacity cache, the L2 is a medium-speed larger-capacity cache, and the L3 is a low-speed large-capacity cache. The L1/L2 cache is a private cache of each CPU core, and the L3 cache is shared by multiple CPU cores. The private cache is privately owned by a CPU core, and the shared cache is shared by multiple CPU cores. In FIG. 2, the L1 and L2 caches in a dotted line box in the upper half part of FIG. 2 are private caches of the CPU core 0, the L1 and L2 caches in a dotted line box in the

lower half part of FIG. 2 are private caches of the CPU core 1, and the L3 cache is a shared cache of the CPU core 0 and the CPU core 1.

[0084] In addition, in each embodiment below, the description uses an example in which the physical memory data block is a memory page, attribute information of a memory page is stored in a page table, and a memory stores page tables of all memory pages. The attribute information of the memory page may indicate whether the memory page is shared, whether the memory page can enter a cache, and the like.

Embodiment 1

[0085] The following provides a specific implementation solution for a multilevel cache-based data read method according to the foregoing computer system.

[0086] In the multilevel cache-based data read method provided in this embodiment, attribute information in the foregoing memory page further includes cache location attribute information of a physical memory page, where the cache location attribute information is set according to an access status of the physical memory page.

[0087] As shown in FIG. 3, the multilevel cache-based data read method may further include the following steps.

[0088] Step (1): Set a cache location attribute of a memory page.

[0089] In this specific embodiment, according to an access status of the physical memory page, several bits are added in a page table to identify cache access level (CAL) information, where the CAL information is used to identify a highest level of an enterable cache among levels of caches storable for the memory page. The CAL information may be set by an operating system directly when pages are allocated. In this embodiment, the page table of a memory is shown in FIG. 3. CAL information corresponding to a high-order bit PA_1 of a physical address is "10", CAL information corresponding to a high-order bit PA_2 of the physical address is "00", and CAL information corresponding to a high-order bit PA_3 of the physical address is "01", and so on.

[0090] Step (2): A CPU core fills a TLB entry with the foregoing page table in the memory, and uses a reserved bit in the TLB entry to identify the CAL information. If no reserved bit exists, width of the TLB entry may be increased.

[0091] It should be noted that in this embodiment, PA_3 is used as an example for description. Therefore, in the TLB entry of FIG. 3, only the page table corresponding to PA_3 is shown, but another page table also exists in the TLB entry.

[0092] The foregoing two steps may be performed when a system is initialized or before a program runs, and are not necessarily performed every time when data is read/written.

[0093] Step (3): A multilevel cache-based data read/write apparatus in the CPU core queries a high-order bit of the physical address of the memory page and the CAL information.

[0094] In this specific embodiment, when initiating a data read request, the multilevel cache-based data read/write apparatus in the CPU core gives a virtual address VA_3, and queries a TLB to find a high-order bit PA_3 of a physical address corresponding to the virtual address VA_3, and "01" serving as CAL information corresponding to the PA_3.

[0095] Step (4): According to PA_3 and "01" that serves as CAL information corresponding to PA_3, query whether an L2 cache is hit.

[0096] In this specific embodiment, because the CAL information "01" identifies that the highest level among the levels

of the caches storable for the memory page is an L2 cache, an L1 cache is ignored, and an access request is initiated to the L2 cache directly, and the high-order bit PA_3 of the physical address is sent. The high-order bit PA_3 of the physical address is compared with a Tag in the L2 cache to obtain a result indicating whether the L2 cache is hit.

[0097] If the L2 cache is hit, step (5) is further performed.

[0098] Step (5): Transmit required data to the multilevel cache-based data read/write apparatus in the CPU core using a direct data path.

[0099] According to the multilevel cache-based data read operation method provided in this embodiment of the present disclosure, a cache location attribute indicates a level of a cache storable for a physical memory data block, so that a CPU core may query, according to the enterable level indicated by the cache location attribute, whether a cache is hit, which is conducive to improving a cache hit rate and conducive to reducing a cache miss rate in a data read operation process.

Embodiment 2

[0100] In view of that an L2 cache is missed in step (4) in Embodiment 1, another embodiment is further provided. For the first 4 steps in this embodiment, reference may be made to the first 4 steps in Embodiment 1, which are not described herein again.

[0101] If the L2 cache in step (4) in Embodiment 1 is missed, as shown in FIG. 4, this embodiment further includes:

[0102] Step (5): The multilevel cache-based data read/write apparatus in the CPU core uses the high-order bit PA_3 of a physical address to query whether a next level, namely, L3 cache, is hit.

[0103] The multilevel cache-based data read/write apparatus in the CPU core compares the high-order bit PA_3 of the physical address with a Tag in the L3 cache to obtain a result indicating whether the L3 cache is hit.

[0104] If the L3 cache is hit, step (6) is further performed. If the L3 cache is missed, step (7) is further performed.

[0105] Step (6): If the L3 cache is hit, transmit to-be-read data to the multilevel cache-based data read/write apparatus in the CPU core using a direct data path, and refill the L2 cache with the data at the same time.

[0106] Step (7): If the L3 cache is missed, the multilevel cache-based data read/write apparatus in the CPU core searches a memory page corresponding to the memory PA_3 for to-be-read data, transmits the data to the apparatus using a direct data path, and at the same time, refills the data to the L2 cache and the L3 cache.

[0107] According to the multilevel cache-based data read operation method provided in this embodiment of the present disclosure, a cache location attribute indicates a level of a cache storable for a physical memory data block, and whether a cache is hit may be queried according to the enterable level indicated by the cache location attribute. Therefore, it is supported that a cache that does not need to be accessed can be skipped, and a cache is hit quickly, which is conducive to improving cache access efficiency and therefore conducive to enhancing a cache hit rate, and in other words, conducive to reducing a cache miss rate in a data read operation process. By restricting a cache location attribute, an infrequently accessed physical memory data block can be prevented from entering a high-level cache with insufficient resources, which is therefore conducive to reducing the cache miss rate in the data read operation process and also avoids futile cache data refill.

Embodiment 3

[0108] The following provides a specific implementation solution for a multilevel cache-based data write method according to the foregoing computer system.

[0109] Settings of a cache location attribute of a memory page in this embodiment are the same as those in Embodiment 1. That is, page tables are the same. Therefore, in this embodiment, when the multilevel cache-based data read/write apparatus in the CPU core initiates a data read/write request, for the first 4 steps, reference may be made to the first 4 steps in Embodiment 1, and a difference is that a data read request is changed to a data write request. Other content is not repeated herein. If the L2 cache in step (4) is hit, as shown in FIG. 5, this embodiment further includes:

[0110] Step (5): Write the data into the L2 cache.

[0111] Further, if the data written in step (5) is shared data, the following step may be further performed to maintain consistency of the shared data.

[0112] Step (6): The multilevel cache-based data read/write apparatus in the CPU core initiates a shared data consistency request to another L2 cache and an L3 cache.

[0113] If the L2 cache uses a write-back manner, the shared data in the L3 cache and the shared data in the L2 cache of another CPU core are set to be invalid. If the L2 cache uses a write-through manner, the shared data in the L3 cache is updated by data that is newly written into the L2 cache, and the shared data in the L2 cache of another CPU core is set to be invalid.

[0114] According to the multilevel cache-based data write operation method provided in this embodiment of the present disclosure, a cache location attribute indicates a level of a cache storable for a physical memory data block, and whether a cache is hit is queried according to the enterable level indicated by the cache location attribute. Therefore, it is supported that a cache that does not need to be accessed can be skipped, and a cache is hit quickly, which is conducive to improving cache hit efficiency and therefore conducive to enhancing a cache hit rate, and in other words, conducive to reducing a cache miss rate in a data write operation process. By restricting a cache location attribute, an infrequently accessed physical memory data block can be prevented from entering a high-level cache with insufficient resources, which is therefore conducive to reducing the cache miss rate in the data write operation process. For shared data, a consistency request can be sent pertinently to a cache that possibly owns the shared data, thereby reducing cache search and snooping.

Embodiment 4

[0115] CAL information in Embodiments 1, 2 and 3 may also be set in an instruction manner. Instead of storing the CAL information in a page table or a TLB, a system register is set for the CAL information. That is, a system register is added to store a cache location attribute so that data can be read/written according to cache location attribute information stored in the system register.

[0116] As shown in FIG. 6, this embodiment provides a multilevel cache-based data write operation method. Specific steps include:

[0117] Step (1): A CAL register (CALR) is applied for in a system register file. A new instruction may be added or some bits may be added in an existing instruction set to further set the CALR, and the CALR keeps unchanged until it is set again. As shown in FIG. 6, the system register includes a main

identifier register (MIDR) (that is device identifier register), a physical address register (PAR), a CALR, and the like. In this embodiment, the CALR is set to 1 (namely, a value in the CALR is 1, or in other words, the CAL information is 1), and it is indicated that a memory page can enter an L2 cache at most.

[0118] Step (2): A multilevel cache-based data read/write apparatus in a CPU core obtains information CALR=1 from the system register in a process of initiating a data write request, and sends a virtual address VA_1 to the TLB.

[0119] Step (3): The multilevel cache-based data read/write apparatus in the CPU core queries a high-order bit of a physical address of a memory page.

[0120] In this specific embodiment, the multilevel cache-based data read/write apparatus in the CPU core initiates a data write request, and sends a virtual address VA_1 to the TLB. The TLB is responsible for translating the virtual address sent by the apparatus into a high-order bit of the physical address. As shown in FIG. 6, the TLB stores attribute information of a memory page, including PA_1, PA_2, PA_3, and PA_4 of a physical address and the like, but without CAL information. The apparatus uses the virtual address VA_1 to search the TLB to obtain the high-order bit PA_1 of the physical address corresponding to the virtual address VA_1.

[0121] Step (4): According to the CALR=1 obtained in step (2) and the PA_1 obtained in step (3), query whether an L2 cache is hit.

[0122] Furthermore, the high-order bit PA_1 of the physical address is compared with a Tag in the L2 cache to obtain a result indicating whether the L2 cache is hit.

[0123] If the L2 cache is hit, step (5) is further performed.

[0124] Step (5): Write the data into the L2 cache directly.

[0125] If the data written in step (5) is shared data, reference may be made to the write-back or write-through manner in Embodiment 3 to maintain consistency of the shared data, and no repeated description is given herein again.

[0126] In the multilevel cache-based data read/write operation method provided in this embodiment of the present disclosure, a system register CALR is set, a cache location attribute indicates a level of a cache storable for a physical memory data block, and whether a cache is hit is queried according to the enterable level indicated by the cache location attribute. Therefore, it is supported that a cache that does not need to be accessed can be skipped, and a cache is hit quickly, which is conducive to improving cache access efficiency and therefore conducive to enhancing a cache hit rate, and in other words, conducive to reducing a cache miss rate in a data write operation process. By restricting a cache location attribute, an infrequently accessed physical memory data block can be prevented from entering a high-level cache with insufficient resources, which is therefore conducive to reducing the cache miss rate in the data write operation process.

[0127] An embodiment of the present disclosure further provides a multilevel cache-based read/write apparatus. The apparatus may be a functional module of a CPU core, and each functional module in the apparatus corresponds to a step in the foregoing method, and is not detailed herein again.

[0128] As shown in FIG. 7, an apparatus 70 includes a first acquiring unit 701 configured to acquire a first query address of a first physical memory data block in which data is to be read/written, a second acquiring unit 702 configured to acquire a first cache location attribute of the first physical memory data block, where the first cache location attribute is used to indicate a level of a cache storable for the first physical

memory data block, a querying unit **703** configured to query whether a cache is hit until one cache is hit or all caches are missed, where the querying is performed according to the first query address obtained by the first acquiring unit **701** and is performed in descending order of levels of caches storable for the first physical memory data block, and the levels of the caches are indicated by the first cache location attribute obtained by the second acquiring unit **702**, and a read/write unit **704** configured to, if one cache is hit, read/write the data in the first query address of the first physical memory data block in the cache hit by the querying unit **703**, or, if a query result of the querying unit **703** is that all caches are missed, read/write the data in the first query address of the first physical memory data block in a memory.

[0129] Further, the first cache location attribute is set in a TLB for the first physical memory data block according to an access status of the first physical memory data block, and the second acquiring unit **702** is configured to query, according to attribute information that is of a physical memory data block and stored in the TLB, a first cache location attribute corresponding to the first query address, where the TLB stores the attribute information of the physical memory data block and the attribute information includes a correspondence between a query address of a physical memory data block and a cache location attribute of the physical memory data block.

[0130] Alternatively, further, the first cache location attribute is set in a system register for the first physical memory data block according to an access status of the first physical memory data block, and the second acquiring unit **702** is configured to read a cache location attribute stored in a current system register, and use the cache location attribute as the first cache location attribute of the first physical memory data block, where the system register is used to store the cache location attribute.

[0131] Further, the read/write unit **704** is configured to read/write the data in the first query address of the first physical memory data block in the hit cache by using a direct data path between the apparatus **70** and the hit cache; or is configured to read/write the data in the first query address of the first physical memory data block in the memory by using a direct data path between the apparatus **70** and the memory.

[0132] Further, the read/write unit **704** is further configured to, during data reading, if a level of the hit cache is a non-highest level in the levels, of the enterable caches, indicated by the first cache location attribute, refill, with to-be-read data, a cache whose level is higher than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute, or if a level of the hit cache is a highest level in the levels, of the enterable caches, indicated by the first cache location attribute, skip refilling another cache with data; or if all caches are missed, refill, with to-be-read data, an enterable cache indicated by the first cache location attribute.

[0133] Further, the read/write unit **704** is further configured to, during data writing, if the written data is shared data, initiate a shared data consistency request to another cache whose level is equal to and/or lower than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute.

[0134] Optionally, the first cache location attribute identifies the highest level among the levels of the caches storable for the first physical memory data block, or identifies a level of a cache that is not storable for the first physical memory data block.

[0135] According to the apparatus provided in this embodiment of the present disclosure, a cache location attribute indicates a level of a cache storable for a physical memory data block so that a querying unit in the apparatus can query, according to the enterable level indicated by the cache location attribute, whether a cache is hit. Therefore, it is supported that a cache that does not need to be accessed can be skipped, and a cache is hit quickly, which is conducive to improving cache access efficiency and therefore conducive to enhancing a cache hit rate, and in other words, conducive to reducing a cache miss rate in a data read/write process. By restricting a cache location attribute, an infrequently accessed physical memory data block can be prevented from entering a high-level cache with insufficient resources, which is therefore conducive to reducing the cache miss rate in the data read/write process.

[0136] The present disclosure further provides a computer system, where the system includes a processor and a memory connected to the processor.

[0137] The memory may be an internal memory, and the memory stores a group of program code. The processor is configured to invoke the program code stored in the memory, where the program code is used to implement any multilevel cache-based data read/write method provided above.

[0138] The processor is configured to invoke the program code for multilevel cache-based data read/write to acquire a first query address of a first physical memory data block in which data is to be read/written, acquire a first cache location attribute of the first physical memory data block, where the first cache location attribute is used to indicate a level of a cache storable for the first physical memory data block, query whether a cache is hit until one cache is hit or all caches are missed, where the querying is performed according to the first query address in descending order of levels of caches storable for the first physical memory data block, and the levels of the caches are indicated by the first cache location attribute; and if one cache is hit, read/write the data in the first physical memory data block in the hit cache, or if all caches are missed, read/write the data in the first physical memory data block in the memory.

[0139] Optionally, the processor is further configured to invoke the program code for multilevel cache-based data read/write to set a first cache location attribute in a TLB for the first physical memory data block according to an access status of the first physical memory data block, and query, according to attribute information that is of a physical memory data block and stored in the TLB, a first cache location attribute corresponding to the first query address, where the TLB stores the attribute information of the physical memory data block and the attribute information includes a correspondence between a query address of a physical memory data block and a cache location attribute of the physical memory data block.

[0140] Optionally, the processor is further configured to invoke the program code for multilevel cache-based data read/write to set a first cache location attribute in a system register for the first physical memory data block according to an access status of the first physical memory data block, read a cache location attribute stored in a current system register, and use the cache location attribute as the first cache location attribute of the first physical memory data block, where the system register is used to store the cache location attribute.

[0141] The processor is further configured to invoke the program code for multilevel cache-based data read/write, and

read/write the data in the first query address of the first physical memory data block in the hit cache by using a direct data path to the hit cache, or read/write the data in the first query address of the first physical memory data block in the memory using a direct data path to the memory.

[0142] Further, the processor is further configured to invoke the program code for multilevel cache-based data read, and if a level of the hit cache is a non-highest level in the levels, of the enterable caches, indicated by the first cache location attribute, the method further includes refilling, with to-be-read data, a cache whose level is higher than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute; or if a level of the hit cache is a highest level in the levels, of the enterable caches, indicated by the first cache location attribute, skipping refilling another cache with data; or if all caches are missed, the method further includes refilling, with to-be-read data, an enterable cache indicated by the first cache location attribute.

[0143] Further, the processor is further configured to invoke the program code for multilevel cache-based data write, and, if the written data is shared data, initiate a shared data consistency request to another cache whose level is equal to and/or lower than the level of the hit cache among the levels, of the enterable caches, indicated by the first cache location attribute.

[0144] In the several embodiments provided in the present application, it should be understood that the disclosed system, apparatus, and method may be implemented in other manners. For example, the described apparatus embodiment is merely exemplary. For example, the unit division is merely logical function division and may be other division in actual implementation. For example, a plurality of units or components may be combined or integrated into another system, or some features may be ignored or not performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented by using some interfaces. The indirect couplings or communication connections between the apparatuses or units may be implemented in electronic, mechanical, or other forms.

[0145] The units described as separate parts may or may not be physically separate, and parts displayed as units may or may not be physical units, may be located in one position, or may be distributed on a plurality of network units. Some or all of the units may be selected according to actual needs to achieve the objectives of the solutions of the embodiments.

[0146] In addition, functional units in the embodiments of the present disclosure may be integrated into one processing unit, or each of the units may exist alone physically, or two or more units are integrated into one unit. The integrated unit may be implemented in a form of hardware, or may be implemented in a form of hardware in addition to a software functional unit.

[0147] When the foregoing integrated unit is implemented in a form of a software functional unit, the integrated unit may be stored in a computer-readable storage medium. The software functional unit is stored in a storage medium and includes several instructions for instructing a computer device (which may be a personal computer, a server, or a network device) to perform some of the steps of the methods described in the embodiments of the present disclosure. The foregoing storage medium includes: any medium that can store program code, such as a universal serial bus (USB) flash

drive, a removable hard disk, a read-only memory (ROM), a random access memory (RAM), a magnetic disk, or an optical disc.

[0148] Finally, it should be noted that the foregoing embodiments are merely intended for describing the technical solutions of the present disclosure but not for limiting the present disclosure. Although the present disclosure is described in detail with reference to the foregoing embodiments, persons of ordinary skill in the art should understand that they may still make modifications to the technical solutions described in the foregoing embodiments or make equivalent replacements to some technical features thereof, without departing from the spirit and scope of the technical solutions of the embodiments of the present disclosure.

What is claimed is:

1. A cache-based data read/write method applied in a computer system, wherein the computer system comprises a processor, a memory and a plurality of cache memories organized in a hierarchy, and wherein the cache-based data read/write method comprises:

acquiring a first query address by the processor, wherein the first query address indicates a first physical memory data block in which data is to be read/written;

acquiring a first cache location attribute of the first physical memory data block, wherein where the first cache location attribute indicates a level in levels of caches storable for the first physical memory data block;

querying whether the level of cache is hit until one level of cache is hit or all levels of caches are missed, wherein the querying is performed according to the first query address in descending order of levels of caches storable for the first physical memory data block, and wherein the highest level of cache is indicated in the first cache location attribute; and

reading/writing the data in the first query address of the first physical memory data block in a hit level of cache when one level of cache is hit; and

reading/writing the data in the first query address of the first physical memory data block in the memory when all levels of caches are missed.

2. The method according to claim 1, wherein the first cache location attribute identifies a highest level in levels of caches storable for the first physical memory data block.

3. The method according to claim 1, wherein the first cache location attribute is set in a translation lookaside buffer (TLB) for the first physical memory data block according to an access status of the first physical memory data block, and wherein the method further comprises querying, according to attribute information that corresponding to physical memory data blocks and stored in the TLB, the first cache location attribute corresponding to the first query address, wherein the TLB stores the attribute information of a physical memory data block, and wherein the attribute information comprises a correspondence between a query address of the physical memory data block and a cache location attribute of the physical memory data block.

4. The method according to claim 1, wherein the first cache location attribute is set in a system register for the first physical memory data block according to the access status of the first physical memory data block, and wherein the method further comprises:

reading the cache location attribute stored in the system register; and

using the cache location attribute as the first cache location attribute of the first physical memory data block, wherein the system register is used to store the cache location attribute.

5. The method according to claim 1, wherein reading/writing the data in the first query address of the first physical memory data block in the hit level of cache comprises reading/writing the data in the first query address of the first physical memory data block in the hit level of cache using a direct data path to the hit level of cache.

6. The method according to claim 1, wherein reading/writing the data in the first query address of the first physical memory data block in the memory comprises reading/writing the data in the first query address of the first physical memory data block in the memory using a direct data path to the memory.

7. The method according to claim 1, wherein during data reading, the method further comprises:

refilling, with to-be-read data, a level of cache whose level is higher than the hit level of cache is among the levels of caches storable, indicated by the first cache location attribute when the hit level of cache is a non-highest level in the levels of caches storable, indicated by the first cache location attribute.

8. The method according to claim 1, wherein during data reading, the method further comprises:

skipping refilling another cache with data when the hit level of cache is the highest level in the levels of caches storable, indicated by the first cache location attribute.

9. The method according to claim 1, wherein during data reading, the method further comprises:

refilling, with to-be-read data, a storable cache indicated by the first cache location attribute when all levels of caches are missed.

10. The method according to claim 1, wherein during data writing, after writing the data in the first physical memory data block in the hit level of cache, the method further comprises initiating a shared data consistency request to another cache whose level is not greater than the level of the hit cache among the levels of caches storable, indicated by the first cache location attribute when the written data is shared data.

11. A computer system having a plurality of cache memories organized in a hierarchy, comprising:

a processor; and

a memory coupled to the processor and the plurality of cache memories organized in the hierarchy,

wherein the processor is configured to:

acquire a first query address, wherein the first query address is used to indicate a first physical memory data block in which data is to be read/written;

acquire a first cache location attribute of the first physical memory data block, where the first cache location attribute indicates a level in levels of caches storable for the first physical memory data block;

query whether the level of cache is hit until one level of cache is hit or all levels of caches are missed, wherein the querying is performed according to the first query address in descending order of levels of caches storable for the first physical memory data block, and wherein the highest level of cache is indicated in the first cache location attribute; and

read/write the data in the first query address of the first physical memory data block in a hit level of cache when one level of cache is hit; and

read/write the data in the first query address of the first physical memory data block in the memory when all levels of caches are missed.

12. The computer system according to claim 11, wherein the first cache location attribute identifies a highest level in levels of caches storable for the first physical memory data block.

13. The computer system according to claim 11, wherein the first cache location attribute is set in a translation look-aside buffer (TLB) for the physical memory data block according to an access status of the first physical memory data block, and wherein the processor is further configured to query, according to attribute information that corresponds to physical memory data blocks and stored in the TLB, the first cache location attribute corresponding to the first query address, wherein the TLB stores the attribute information of a physical memory data block, and wherein the attribute information comprises a correspondence between a query address of the physical memory data block and a cache location attribute of the physical memory data block.

14. The computer system according to claim 11, wherein the first cache location attribute is set in a system register for the first physical memory data block according to the access status of the first physical memory data block, and wherein the processor is further configured to:

read the cache location attribute stored in the system register; and

use the cache location attribute as the first cache location attribute of the first physical memory data block, wherein the system register is used to store the cache location attribute.

15. The computer system according to claim 11, wherein the processor is further configured to read/write the data in the first query address of the first physical memory data block in the hit level of cache using a direct data path to the hit level of cache.

16. The computer system according to claim 11, wherein the processor is further configured to read/write the data in the first query address of the first physical memory data block in the memory using a direct data path to the memory.

17. The computer system according to claim 11, wherein during data reading, the processor is further configured to:

refill, with to-be-read data, a level of cache whose level is higher than the hit level of cache is among the levels of caches storable, indicated by the first cache location attribute when a level of the hit cache is a non-highest level in the levels of the enterable caches, indicated by the cache location attribute.

18. The computer system according to claim 11, wherein during data reading, the processor is further configured to:

skip refilling another cache with data when the hit level of hit cache is the highest level in the levels of caches storable, indicated by the first cache location attribute.

19. The computer system according to claim 11, wherein during data reading, the processor is further configured to:

refill, with to-be-read data, a storable cache indicated by the first cache location attribute when all levels of caches are missed.

20. The computer system according to claim 11, wherein during data writing, the processor is further configured to initiate a shared data consistency request to another cache whose level is not greater than the level of the hit cache among the levels of caches storable, indicated by the first cache location attribute when the written data is shared data.