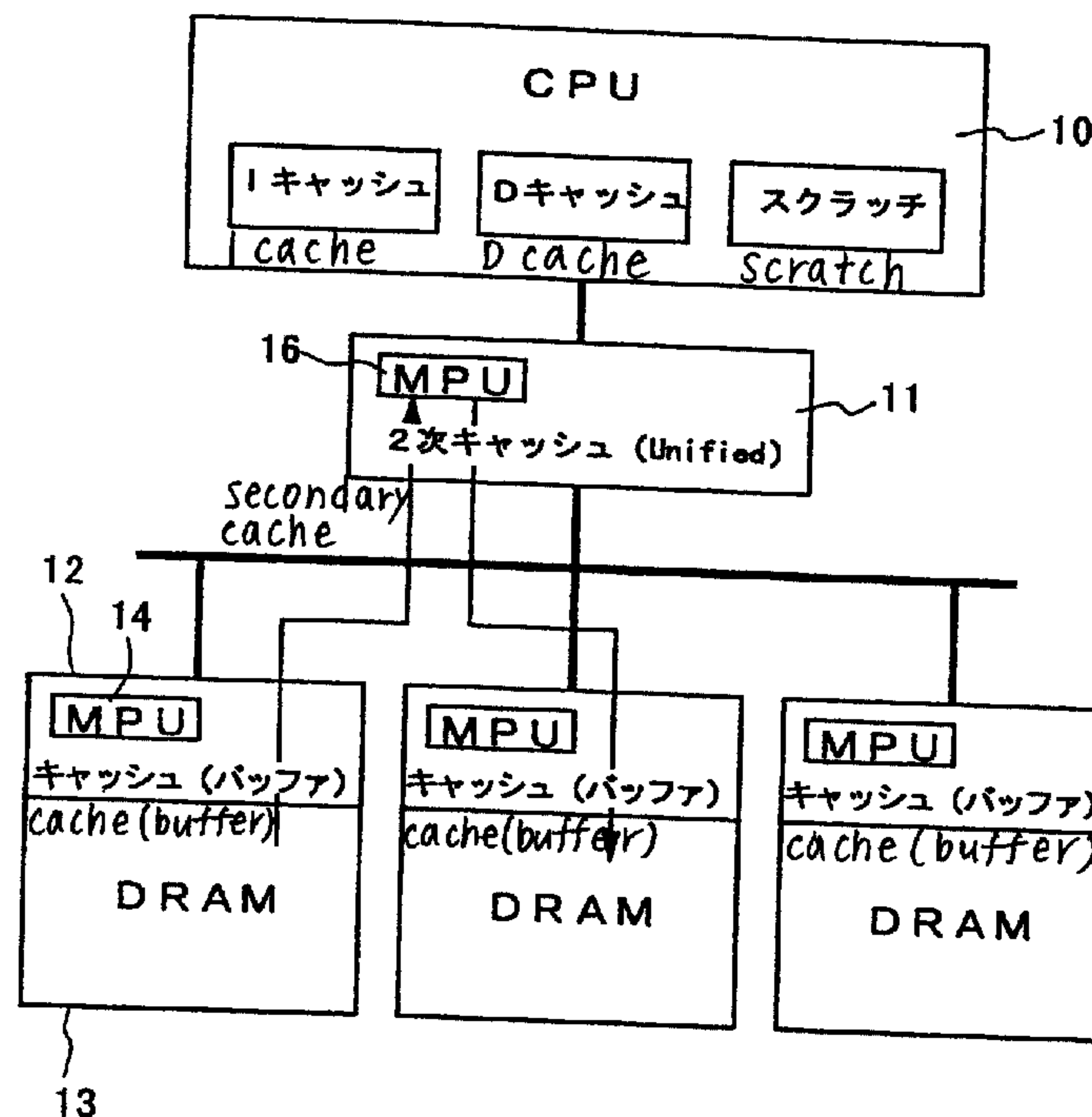




(86) Date de dépôt PCT/PCT Filing Date: 2000/01/21
(87) Date publication PCT/PCT Publication Date: 2000/07/27
(85) Entrée phase nationale/National Entry: 2001/07/20
(86) N° demande PCT/PCT Application No.: JP 2000/000278
(87) N° publication PCT/PCT Publication No.: 2000/043902
(30) Priorité/Priority: 1999/01/21 (11/13486) JP

(51) Cl.Int.⁷/Int.Cl.⁷ G06F 15/78, G06F 12/08
(71) Demandeur/Applicant:
SONY COMPUTER ENTERTAINMENT INC., JP
(72) Inventeur/Inventor:
OHBA, AKIO, JP
(74) Agent: MBM & CO.

(54) Titre : PROCESSEUR A GRANDE VITESSE, METHODE D'UTILISATION ET SUPPORT D'ENREGISTREMENT
(54) Title: HIGH-SPEED PROCESSOR SYSTEM, METHOD OF USING THE SAME, AND RECORDING MEDIUM



(57) Abrégé/Abstract:

The invention is aimed at providing a high-speed processor system capable of performing distributed concurrent processing without requiring modification of conventional programming styles. The processor system in accordance with the invention has a CPU, a plurality of parallel DRAMs, and a plurality of cache memories arranged in a hierarchical configuration. Each of the cache memories is provided with an MPU which is binarily-compatible with the CPU and which has a function to serve as a processor.



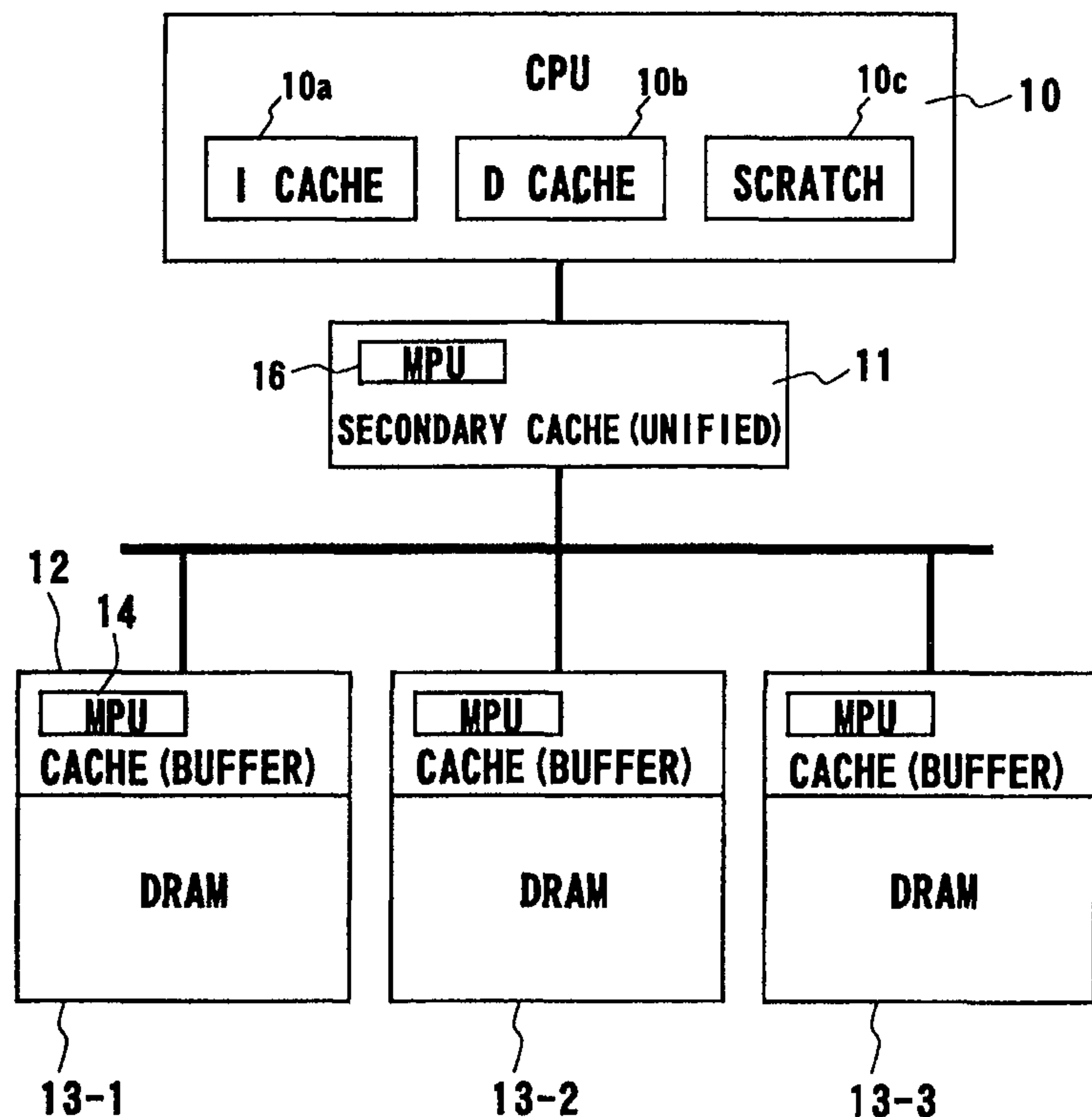
**PCT**WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 15/78, 12/08	A1	(11) International Publication Number: WO 00/43902 (43) International Publication Date: 27 July 2000 (27.07.00)
(21) International Application Number: PCT/JP00/00278 (22) International Filing Date: 21 January 2000 (21.01.00) (30) Priority Data: 11/13486 21 January 1999 (21.01.99) JP (71) Applicant: SONY COMPUTER ENTERTAINMENT INC. [JP/JP]; 1-1, Akasaka 7-chome, Minato-ku, Tokyo 107-0052 (JP). (72) Inventor: OHBA, Akio; Sony Computer Entertainment Inc., 1-1, Akasaka 7-chome, Minato-ku, Tokyo 107-0052 (JP). (74) Agent: YAMAMOTO, Toshitake; 301, Ogikubo Sunny Garden, 28-9, Ogikubo 4-chome, Suginami-ku, Tokyo 167-0051 (JP).		(81) Designated States: AU, BR, CA, CN, JP, KR, MX, NZ, RU, SG, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i>

(54) Title: HIGH-SPEED PROCESSOR SYSTEM, METHOD OF USING THE SAME, AND RECORDING MEDIUM**(57) Abstract**

The invention is aimed at providing a high-speed processor system capable of performing distributed concurrent processing without requiring modification of conventional programming styles. The processor system in accordance with the invention has a CPU, a plurality of parallel DRAMs, and a plurality of cache memories arranged in a hierarchical configuration. Each of the cache memories is provided with an MPU which is binarily-compatible with the CPU and which has a function to serve as a processor.



[Detailed Description of the Invention]**[0001]****[Technical Field Pertinent to the Invention]**

This invention relates to a hierarchically-configured parallel computer system and, more particularly, to an information processing device that can perform high-speed and efficient parallel processing without requiring modification of existing programming styles..

[Prior Art]**[0002]**

A high -speed processor system that has a low-speed large-capacity DRAM and first, second and third cache memories has been known as a system for high-speed processing of large-sized data. Such a known high-speed processor system has, as shown in Fig. 1, a plurality of parallel DRAMs 2 connected to the CPU 1 through a common bus line and a secondary cache 3 equipped in each of the DRAM 2, and a cache memory 3 equipped in each of the DRAMs 2 in order to make a processing speed of the DRAM 2 closer to that of CPU 1.

[0003]

In the operation of the circuitry shown in Fig. 1, contents of one of the DRAMs 2 are read and processed in accordance with an instruction given by the CPU 1, and writing of information into the DRAM 2 also is executed in accordance with an instruction from the CPU 1. If the desired content to be read from the DRAM 2 is held in the cache 3, the CPU can perform high-speed data processing. However, in case of a miss-hit, the cache is required to read the target content from the DRAM 2. Anyway, the arrangement shown in Fig. 1 allows high-speed data transfer through a hierarchy-type cache.

[0004]**[Problem that this invention is to solve]**

however, as described above, the described configuration of the high-speed processor system having a processor, DRAMs, and caches is nowadays the dominant one which advantageously permits the use of an ordinary programming style for the control but cannot perform a parallel processing because it has only one CPU. Specifically, although a parallel processing system employing a special parallel words can be can be produced when a parallel processing is intended, ordinary programming style is not inherently intended for parallel processing and hence a parallel processing system cannot easily be controlled with having a current programming style unless it is modified, thus causing an impediment in practical use

[0005]

In view of the foregoing, an object of the present invention is to provide an information processing device that implements parallel processing having the same structure as that of an ordinary cache without requiring any change of modification of a conventional programming style.

[0006]

[Means for Solving the Problems]

In order to achieve the above objects, the present invention has the following specific items.

[0007]

According to a first aspect of the present invention, an information processing device is characterized by comprising a CPU, and cache memories arranged in parallel to each other on a hierarchical level with respect to said CPU, wherein said cache memories have MPUs serving as both a cache logic and a processor for a lower-level cache memory.

[0008]

According to a second aspect of the present invention, an information processing device is characterized by comprising a CPU, cache memories arranged in

parallel to each other on a hierarchical level with respect to said CPU, and another cache memories provided between said CPU and said respective cache memories which can perform the processing of said CPU, wherein said cache memories have MPUs serving as both a cache logic and a processor for said another cache memory and its lower-level cache memory and said cache memory and its lower-level cache memory.

[0009]

According to a first aspect of the present invention, an information processing device is characterized in that a cache memory which is an SRAM directly connected to a sense amplifier of a DRAM serving as a main memory and connected to a data input/output circuit serves as a unified cache for an MPU and a secondary or tertiary cache for the CPU.

[0010]

According to a first aspect of the present invention, an information processing device is characterized in that said another cache memory serves as a primary cache when accessed from said MPU, and serves as a secondary cache when accessed from the CPU.

[0011]

According to a first aspect of the present invention, an information processing device is characterized in that the data is transferred from the CPU to the MPU in accordance with a control protocol such as a prefetch instruction or an intelligent prefetch instruction.

[0012]

[Mode for carrying out the Invention]

An embodiment of the present invention will be described with reference to Figs. 2 to 9. Referring to Fig. 2, an information processing device has a CPU 10 which includes an I cache (Instruction Cache) 10a, a D cache (Data Cache) 10b, and a scratch

pad memory 10c, a unified cache memory 11 (also referred to as a "secondary cache"), and a plurality of unified cache memories 12 (also referred to as "tertiary caches") that are connected in parallel through a bus line at the lowest hierarchical level, and DRAMs 13-1 to 13-3.

[0013]

Each hierarchical level of the configuration shown in Fig. 2 employs a cache or caches for the purpose of achieving high-speed processing. Since the line size, i.e., the burst read/write length, is greater for the cache memories of lower hierarchical levels than for those of higher hierarchical levels, the secondary cache 11, therefore, is provided other than the tertiary cache 12. In the configuration shown in Fig. 2, the secondary cache 11 is not essential. Namely, the device may have such a configuration that includes the CPU 10 and a plurality of tertiary caches 12 and DRAMs 13.

[0014]

In the arrangement shown in Fig. 2, the MPUs 14, 16 that are binary-compatible with the CPU 10 are provided with each of the secondary cache 12 and the tertiary cache 13. Each of the MPU 14 and the MPU 16 has a double role: namely, a function to serve as the cache logic and a function to serve as a processor. That is, the cache function is a function for controlling the cache memory under the control of the CPU 10, while the processor function is a function to serve as a sub-CPU for a dispersed parallel system.

[0015]

Fig. 3 shows a specific chip 15 having the tertiary cache 12 shown in Fig. 2 and the MPU 14 and the DRAM therefor. The chip 15 has a DRAM array 13a constituting the DRAM 13 as a major part, a sense amplifier 13b, a row address 13c, a column address 13d, a control circuit 13e, and a data input/output circuit 13f. The chip arranged as shown in Fig. 3 employs an SRAM 12 serving as the cache memory. The

SRAM 12 is directly coupled to the sense amplifier 13b that inputs and outputs data to and from the DRAM array 13a. The SRAM 12 exchanges data between itself and the data input/output circuit 13f. The cache memory serving as the SRAM 12 is controlled by the MPU 14 that has both the cache logic function and the processor function. For the MPU 14, the SRAM 12 serves as a simple unified cache, so that read/write operations on the DRAM array 13a are performed via the SRAM 12. For the CPU 10, the SRAM 12 serves as a tertiary cache, so that the MPU 14 performs operations such as the execution of an object constituted by a program and data stored in the DRAM 13a by sending a control signal to the MPU 14 and pre-read of data in accordance with a predetermined prefetch instruction for the cache.

[0016]

The MPU 14 is driven by the prefetch instruction. In general, the processing speed of a processor system depends on a cache which interconnects a CPU and a memory and which serves as a high-speed memory, and this is the reason why positive control of the cache becomes a trend. More particularly, the CPU pre-reads data by using a prefetch instruction. In the present invention, the prefetch instruction for the cache control is further applied to the MPU 14, so that the MPU 14 is also enabled to perform processing.

[0017]

Practically, the MPU 14 employs a scalable RISC (Restricted Instruction Set Computer) CPU core which can be implemented by a comparatively small core such as an ARM (Advanced RISC Machines) processor or a MIPS (Microprocessor without interlocked Pipe Stage) processor, and the MPU is incorporated in a cache memory logic in the system.

[0018]

Fig. 4 illustrates a practical arrangement in the secondary cache 11 which are shown in Fig. 2. Basically, the secondary cache can be regarded as being a processor

which incorporates a unified cache 11a. The MPU 16 is connected to a position of a secondary cache for the CPU 10 and hence can perform the processing function serves as the secondary cache memory. The unified cache 11a in the secondary cache is constituted by an SRAM. The unified cache 11a, when accessed by the CPU 10, serves as the secondary cache for the CPU 10 and, when accessed by the MPU 16, serves as a primary cache for the MPU 16.

[0019]

As stated before, the MPU 16 has a greater burst read/write length than the CPU 10. Therefore, the secondary cache 11 and the MPU 16 are provided. In accordance with control protocols given by the CPU 10, the MPU 16 works as the secondary cache and, at the same time, performs processing of objects constituted by programs and data stored in the tertiary cache and a main memory, the processing in this case being mainly that requiring frequent data transfer between the memories, rather than complicated arithmetic processing. The MPU also executes, in accordance with commands given by the CPU 10, prefetch instructions of a type which are more generic and more sophisticated than those executed by the MPU 14 of each tertiary cache 12 of the above-mentioned DRAM 13, for instance, a prefetch instruction which involves a plurality of DRAMs. In Fig. 4, reference numerals 17 represents a memory interface connected to the DRAM 13.

[0020]

Fig. 5 shows the flow of data as observed when the circuitry shown in Fig. 2 operates in an ordinary cache mode, that is, wherein the MPUs 14 and 16 perform only the cache logic functions without performing the processing functions. When the CPU 10 processes the data contained in the DRAM 13, the data stored in the DRAM 13 is transferred to the secondary cache 11 from the tertiary cache 12 of the lowest hierarchical level and which is designed to have a comparatively large transfer size and a comparatively low transfer frequency, and further transferred to the primary

cache. Then, the data is delivered to the CPU 10 for processing. Writing of data into the DRAM 13 is performed by tracing backward the above-described route.

Consequently, access to the data is performed many times. This fashion of making access may seem to be efficiently managed by a presently-available stack function of a CPU 10. Actually, however, a problem is encountered in that cache-out of data that has to be frequently accessed is caused by data which has to be accessed only once by the CPU 10, such as data for image processing and large-sized data to be retrieved. This leads to an increase in the number of wasteful accesses. The described concept of cache control in accordance with the present invention is based upon the demands for eliminating or reducing the number of such wasteful accesses. Presently available processing systems are designed as shown in Fig. 5, assuming the presence of many access paths. From a practical point of view, the memory architecture of Fig. 5 operable under ordinary programming is very useful.

[0021]

Fig. 6 shows a case where the MPU 14 actually performs its task as a processor and performs distributed processing of a local object. Thus, a local object that need not be processed by the CPU 10 is processed by the MPU 14 in accordance with a control protocol included in a prefetch instruction given by the CPU 10. For instance, an executable program and an object as data are classified in processing in a single DRAM block in case of processing such as merely an incrementing computation or determination of a maximum value. It is thus possible to execute distributed concurrent processing by using the MPU 14. It is to be understood that the DRAM block on which the object processing is executed is cached out from the commanding cache by block during the execution of the distributed concurrent processing.

[0022]

Fig. 7 shows the processing executed by the MPU 16 in the secondary cache, a global transfer processing, and a low-level-computation and high-rate transfer

processing. Namely, if cached contents is not necessary to be supplied to the CPU 10, , e.g., data is transferred between memories, the MPU 16 in the secondary cache undertakes the processing according to the arrangement shown in Fig. 7. Basically, the MPU 16 can make access to all the memories, so that it can serve as a multiprocessor system which executes processing in place of the CPU 10. The MPU 16, however, can most suitably be used for a large-size transfer such as a global transfer of large-sized data, because its computation ability is much lower than that of the CPU 10. Therefore, the MPU 16 can selectively perform processing of the kind which does not require the high computation ability of the CPU 10 or a sophisticated function of the commanding primary cache. The processing performed by the MPU 16 in the secondary cache also is under the control of the control protocol given by the CPU 10.

[0023]

Fig. 8 used to explain specifically an intelligent prefetch instruction. The intelligent prefetch instruction (IPREF) is used as means for enabling control of the MPUs 16 and 14 which are subordinate to the CPU 10 without requiring any change in conventional programming styles. Referring to Fig. 8, the CPU 10 has the I cache 10a and the D cache 10b. A problem of cache coherence is encountered with the use of the MPU 16 as the processor. Namely, inadvantageously, data changed as a result of execution of a program by the MPU 16 may not conform with the data held in the D cache 10b of the CPU 10. In order to avert from this problem, when the CPU 10 instructs the MPU 16 to execute a job, the data in the D cache 10b of the CPU 10 is cached out, so that the content of the D cache 10 is updated with new data (designated data) obtained as a result of execution of the program by the MPU 16. The MPU 16 is inherently a cache, so that it is controllable to function as a cache. To this end, MPU 16 conducts a job in accordance with the IPREF instruction in the same manner as an ordinary cache works in accordance with a prefetch instruction. It is therefore possible to simultaneously control both a cache and the MPU 16 by means of the IPREF

instruction. More specifically, the MPU 16 functions as a cache in response to a prefetch instruction, and conducts a job in response to the IPREF instruction.

[0024]

In other words, referring to Fig. 8, the IPREF is an extended instruction given by the CPU 10. When executed, this extended instruction effects cache-out of a designated area in the D cache 10b and sends a control protocol to the cache of the lower hierarchical level having an MPU. Upon receipt of the control protocol, the designated MPU of the lower hierarchical level executes a program designated by the protocol, and also, by making access to the DRAM or memory blocks of lower hierarchical level, puts desired data in the cache memory.

[0025]

An example of retrieval for determining maximum value data is shown below.

IPREF DRAM0 /*Maximum value from data array in DRAM0 */

IPREF DRAM1 /*Maximum value from data array in DRAM1 */

IPREF DRAM2 /*Maximum value from data array in DRAM2 */

IPREF DRAM3 /*Maximum value from data array in DRAM3 */

[0026]

:

[0027]

:

Load r0 DRAM1-MAX /*Read maximum value in DRAM0 */

Load r1 DRAM1-MAX /*Read maximum value in DRAM1 */

Load r2 DRAM1-MAX /*Read maximum value in DRAM2 */

Load r3 DRAM1-MAX /*Read maximum value in DRAM3 */

Max r0, r0, r1

Max r2, r2, r3

Max r0, r0, r2 /* Retrieval of maximum value data end */

[0028]

This example is based on an assumption that the designated data shown in Fig. 8 has been registered in the DRAM0 to DRAM3. The instructions IPREF DRAM0 to IPREF DRAM3 are instructions for executing a designated program. The program that has been registered is executed in accordance with the IPREF instruction, after effecting cache-out of the contents of the designated area of the D cache 10b. The IPREF is executed on the DRAM0 to DRAM3, while the CPU 10 sends the control protocol to the DRAM1 to DRAM3. Load instructions are executed when maximum values have been set in the caches. In this case, it is possible to determine four maximum values by using eight instructions including four IPREF instructions and four Load instructions, although the number of maximum values obtainable depends on the transfer size of the DRAM. The true maximum value can be determined by checking the obtained maximum values with one another.

[0029]

Fig. 90 shows an arrangement of an I chip in a system formed by super LSI of giga-bit order DRAM. If an ASIC DRAM is provided away in the I chip, it similarly serves as an external cache in spite of the fact that it is provided in the chip, and hence a cache is required. Although the LSI of giga bit order can incorporate a main memory, arrangement in the chip needs consideration of propagation delay, and hence large costs in transfer from a far memory requires a cache memory. In the arrangement shown in Fig. 9, it becomes easy to use both a conventional programming model and dispersed parallel processing, and hence the chip can effectively be used.

[0030]

[Effect of the Invention]

As will be seen from the foregoing description, according to the present invention, a processor system is provided having cache memories each incorporating an MPU that serves both as a cache logic and a processor for a subordinate hierarchical

level. With this processor system, it is possible to effect a high-speed, efficient concurrent processing without requiring any modification of conventional programming styles.

[Brief Description of the Drawings]

[Fig. 1]

A block diagram of a conventional parallel processor.

[Fig. 2]

A block diagram of an embodiment of the present invention.

[Fig. 3]

A block diagram showing a practical example of the chip arrangement of DRAMs, an MPU, and caches.

[Fig. 4]

A block diagram showing internal structures of a secondary cache and the MPU.

[Fig. 5]

A diagram illustrating the flow of data in an ordinary cache mode.

[Fig. 6]

A diagram illustrating the flow of data in a local object distributed execution.

[Fig. 7]

A diagram illustrating the flow of data in a transfer processing performed by the secondary cache.

[Fig. 8]

A diagram showing an intelligent prefetch instruction.

[Fig. 9]

A diagram showing a chip system of an ASIC DRAM.

[Description of reference numerals]

10 CPU

11 secondary cache

13

12 tertiary cache

13 DRAM

14, 16 MPU

[Name of Document] SPECIFICATION

[Title of the Invention] INFORMATION PROCESSING DEVICE

[Scope of Claims for a Patent]

[Claim 1] An information processing device, comprising:

a CPU, and cache memories arranged in parallel to each other on a hierarchical level with respect to said CPU, wherein said cache memories have MPUs serving as both a cache logic and a processor for a lower-level cache memory.

[Claim 2] An information processing device, comprising:

a CPU, cache memories arranged in parallel to each other on a hierarchical level with respect to said CPU, and another cache memories provided between said CPU and said respective cache memories which can perform the processing of said CPU, wherein said cache memories have MPUs serving as both a cache logic and a processor for said another cache memory and its lower-level cache memory and said cache memory and its lower-level cache memory.

[Claim 3] In an information processing device as claimed in Claim 1 or 2,

said information processing device, wherein a cache memory which is an SRAM directly connected to a sense amplifier of a DRAM serving as a main memory and connected to a data input/output circuit serves as a unified cache for an MPU and a secondary or tertiary cache for the CPU.

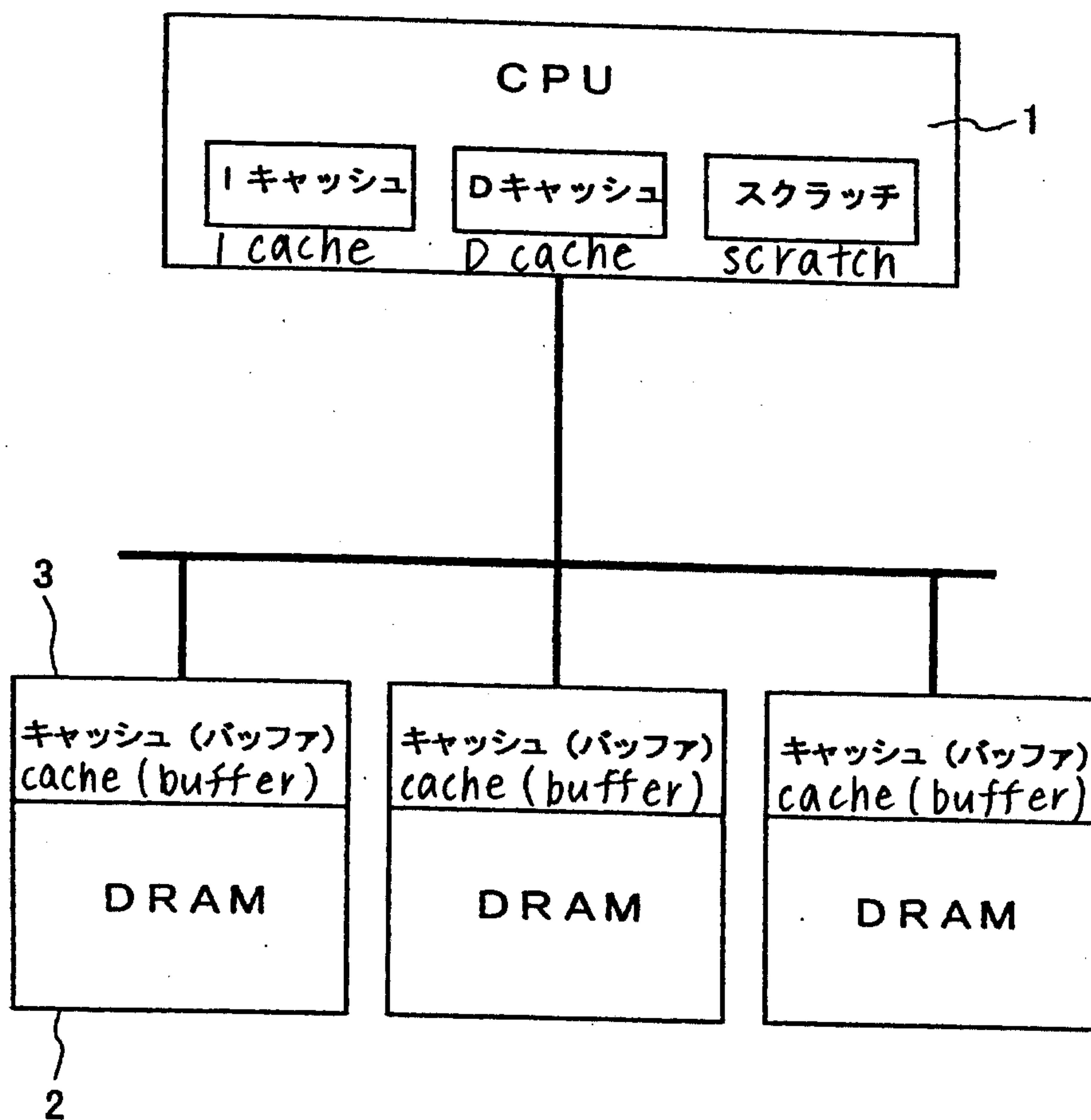
[Claim 4] In an information processing device as claimed in Claim 2,

said information processing device, wherein said another cache memory serves as a primary cache when accessed from said MPU, and serves as a secondary cache when accessed from the CPU.

[Claim 5] In an information processing device as claimed in Claim 1 or 2,

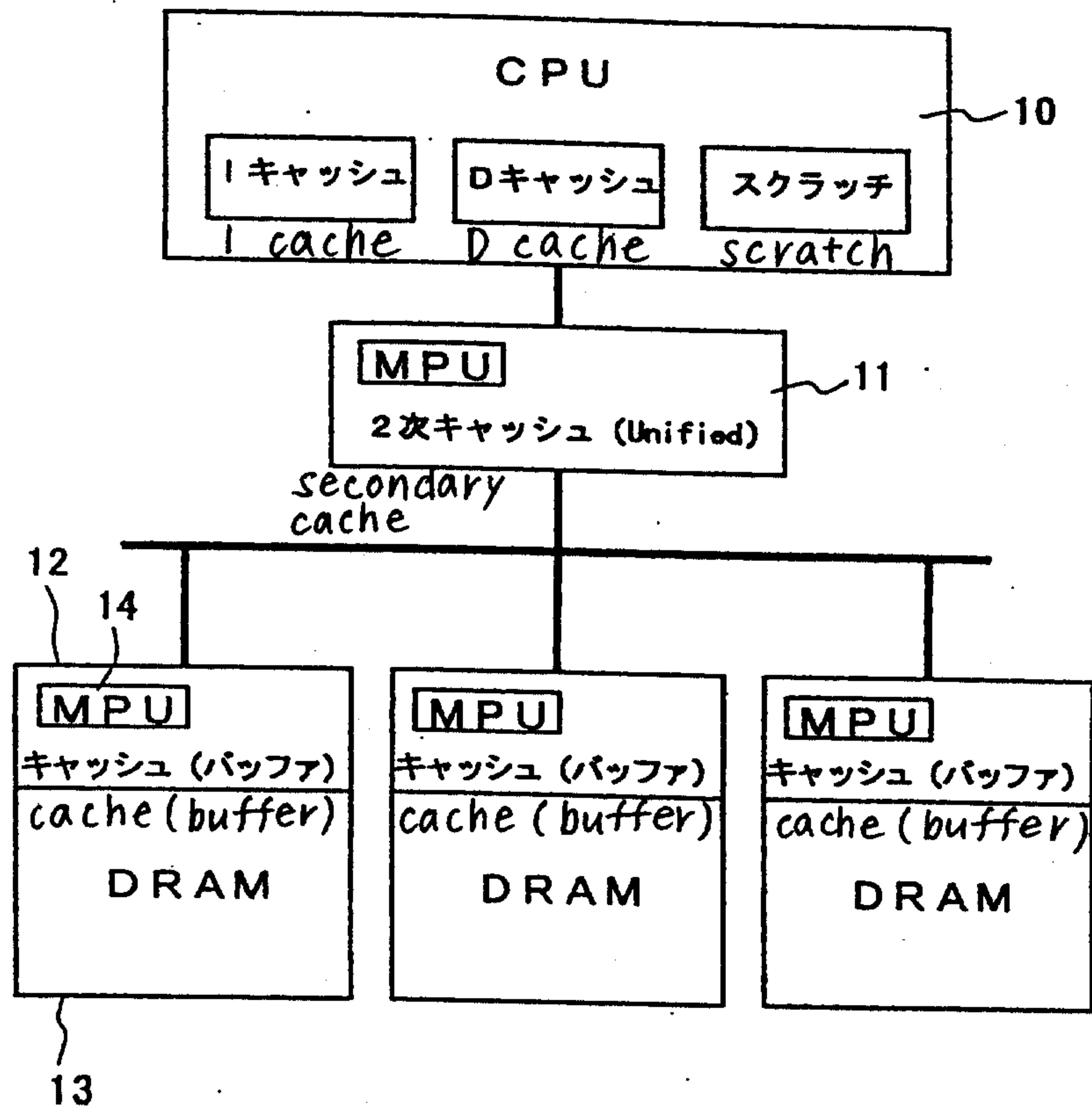
said information processing device, wherein the data is transferred from the CPU to the MPU in accordance with a control protocol such as a prefetch instruction or an intelligent prefetch instruction.

【書類名】 図面

【図 1】
(Fig. 1)

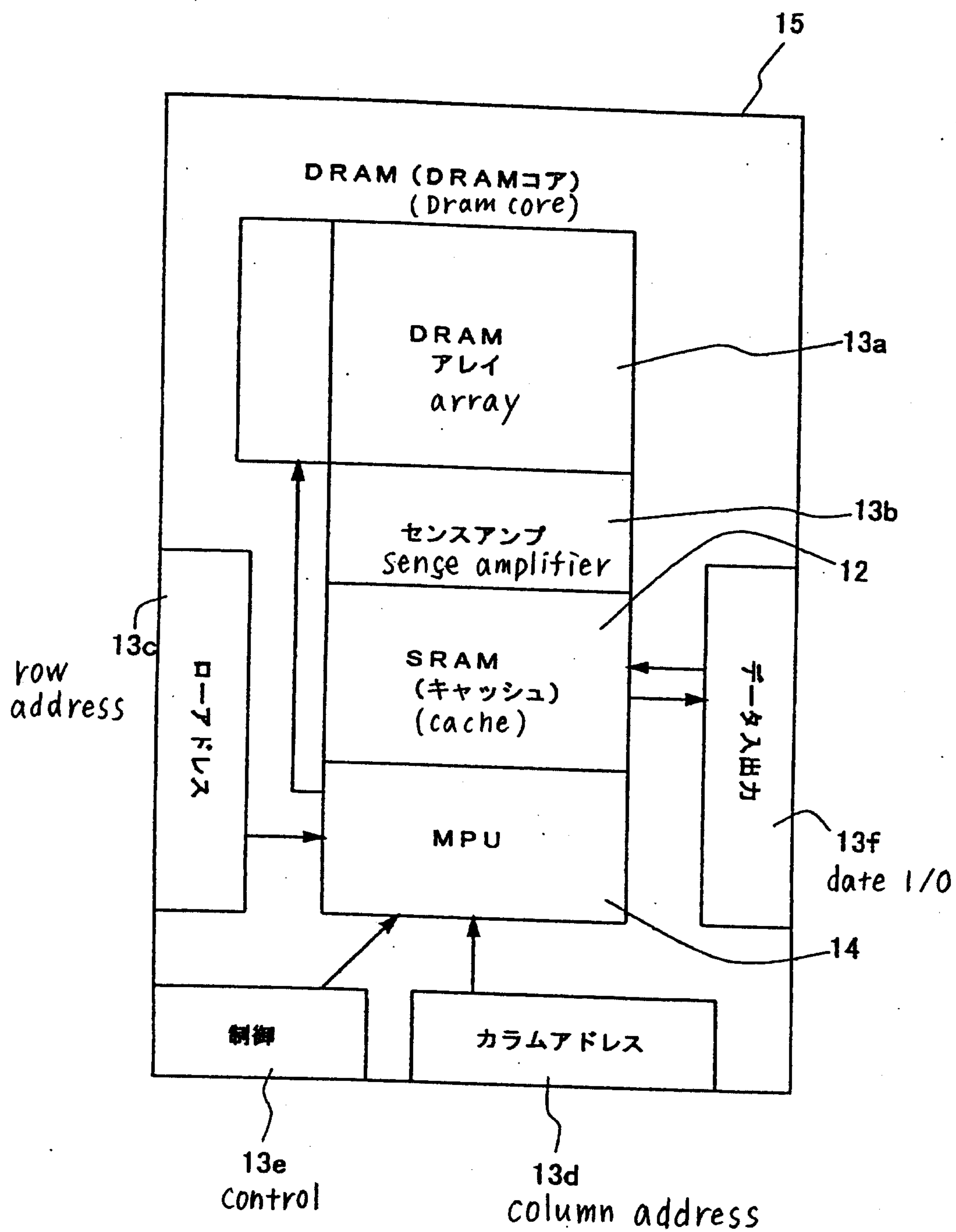
【図2】

(Fig. 2)



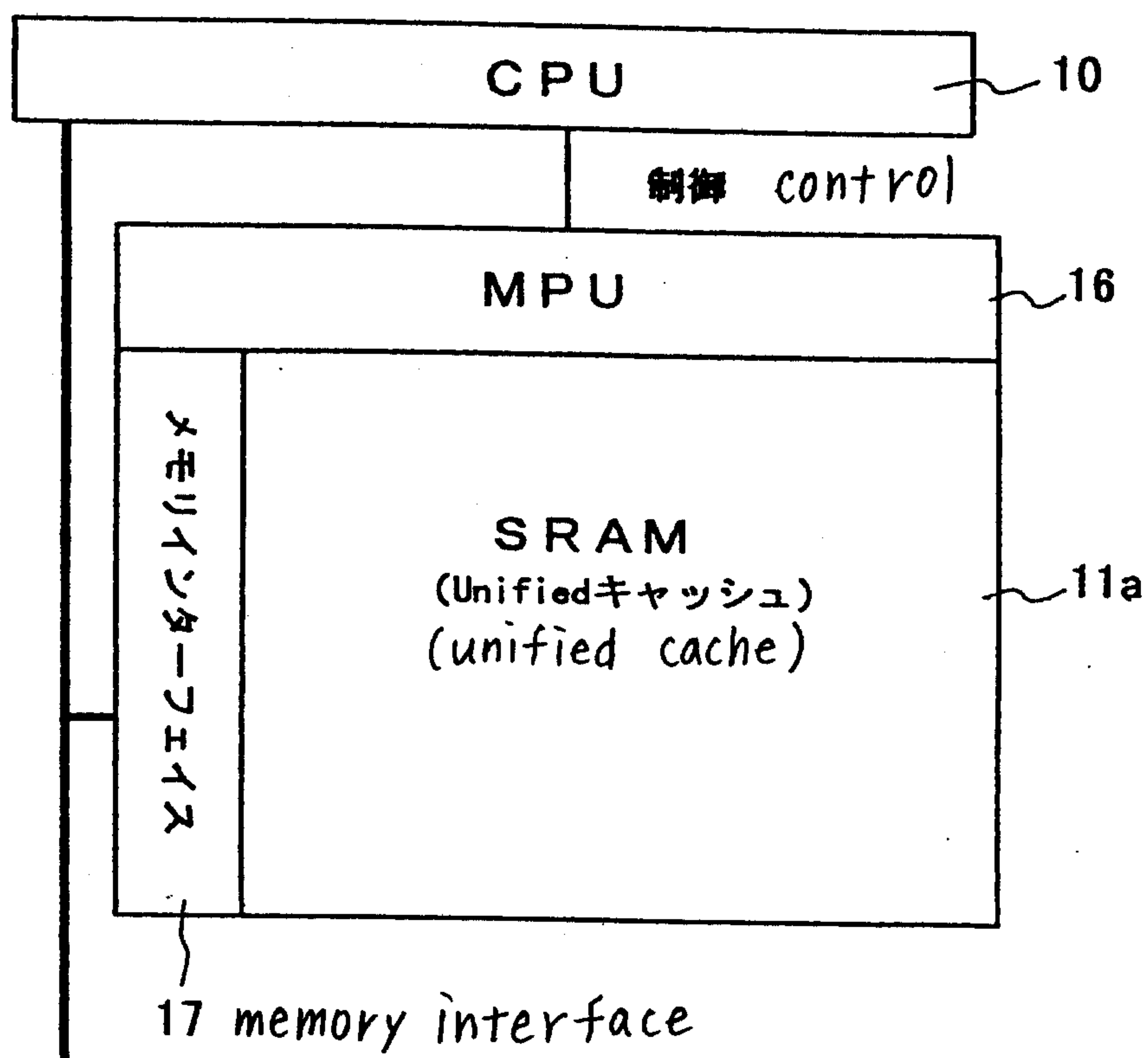
【図 3】

(Fig. 3)



【図4】

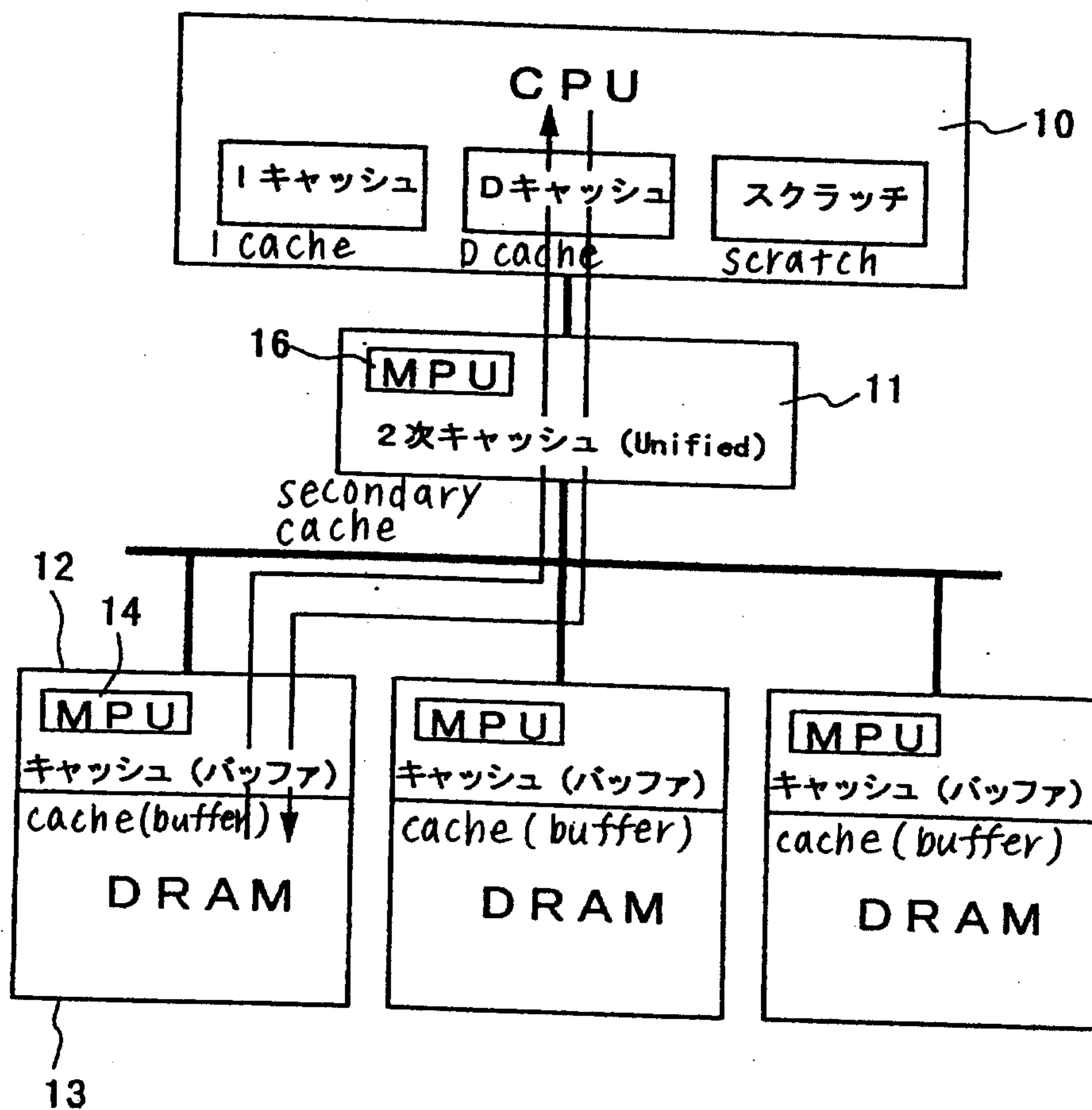
{ Fig. 4 }



DRAM (メインメモリ) へ
to dram (main memory)

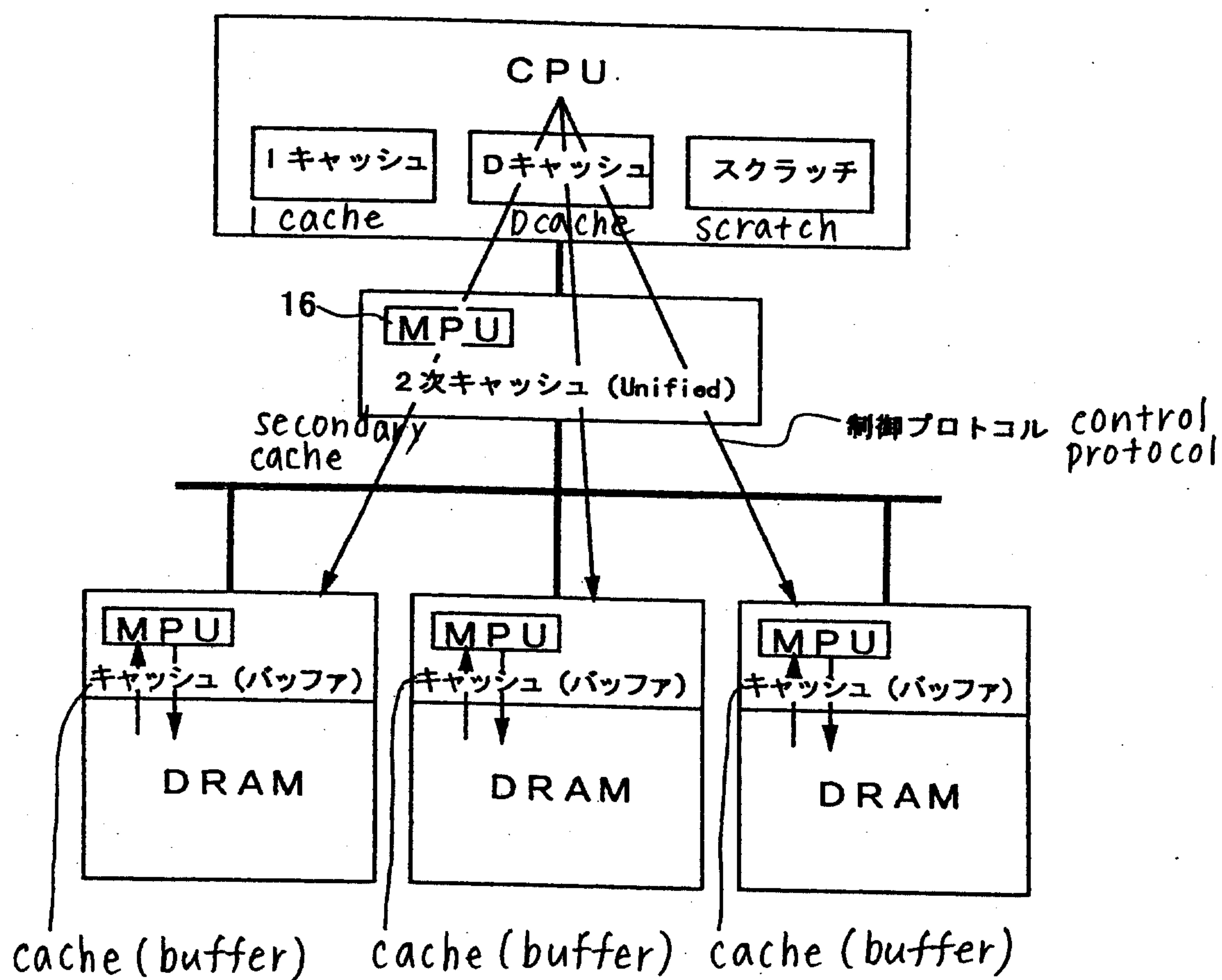
【図5】

(Fig. 5)



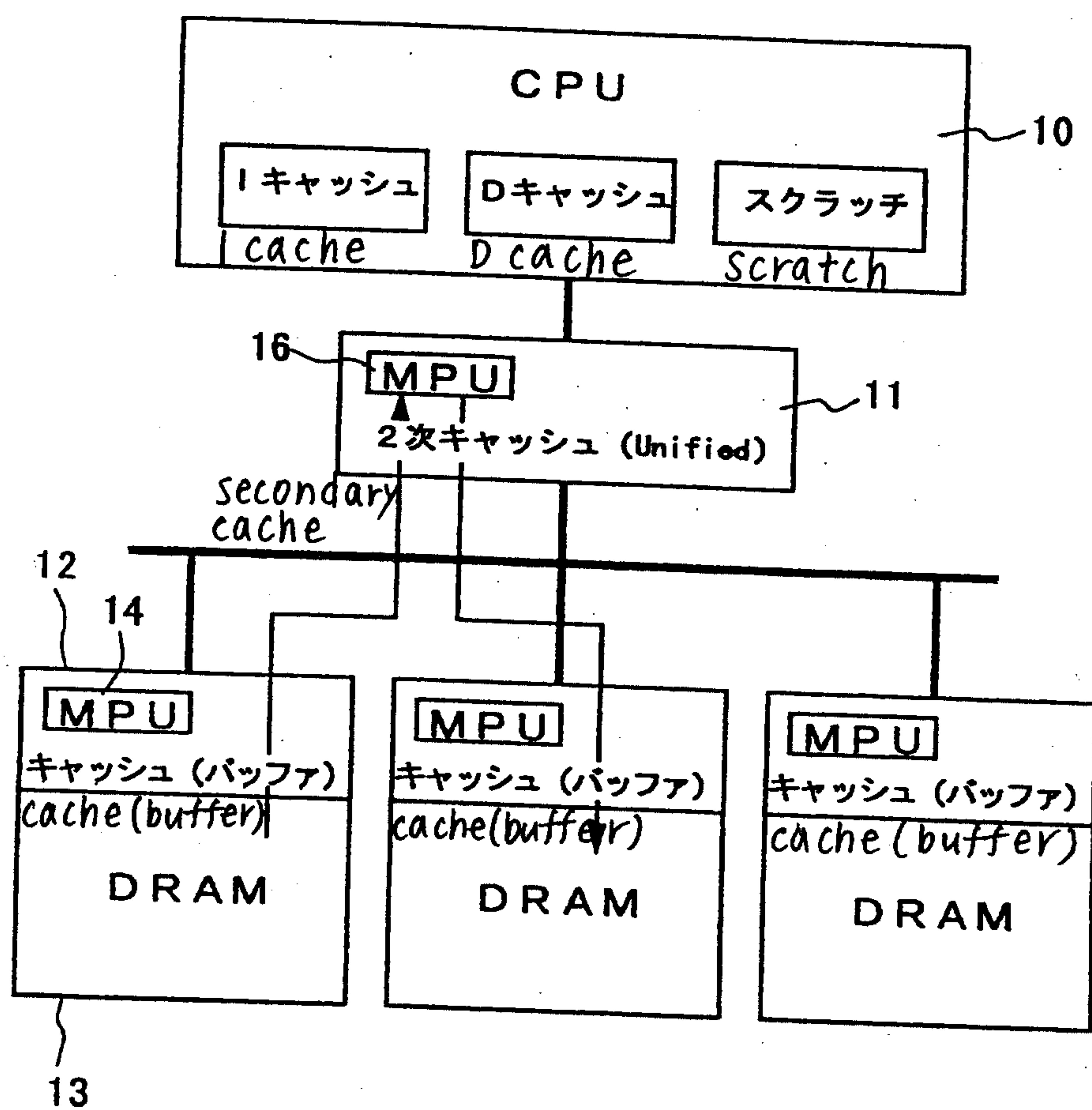
【図6】

{ Fig. 6 }



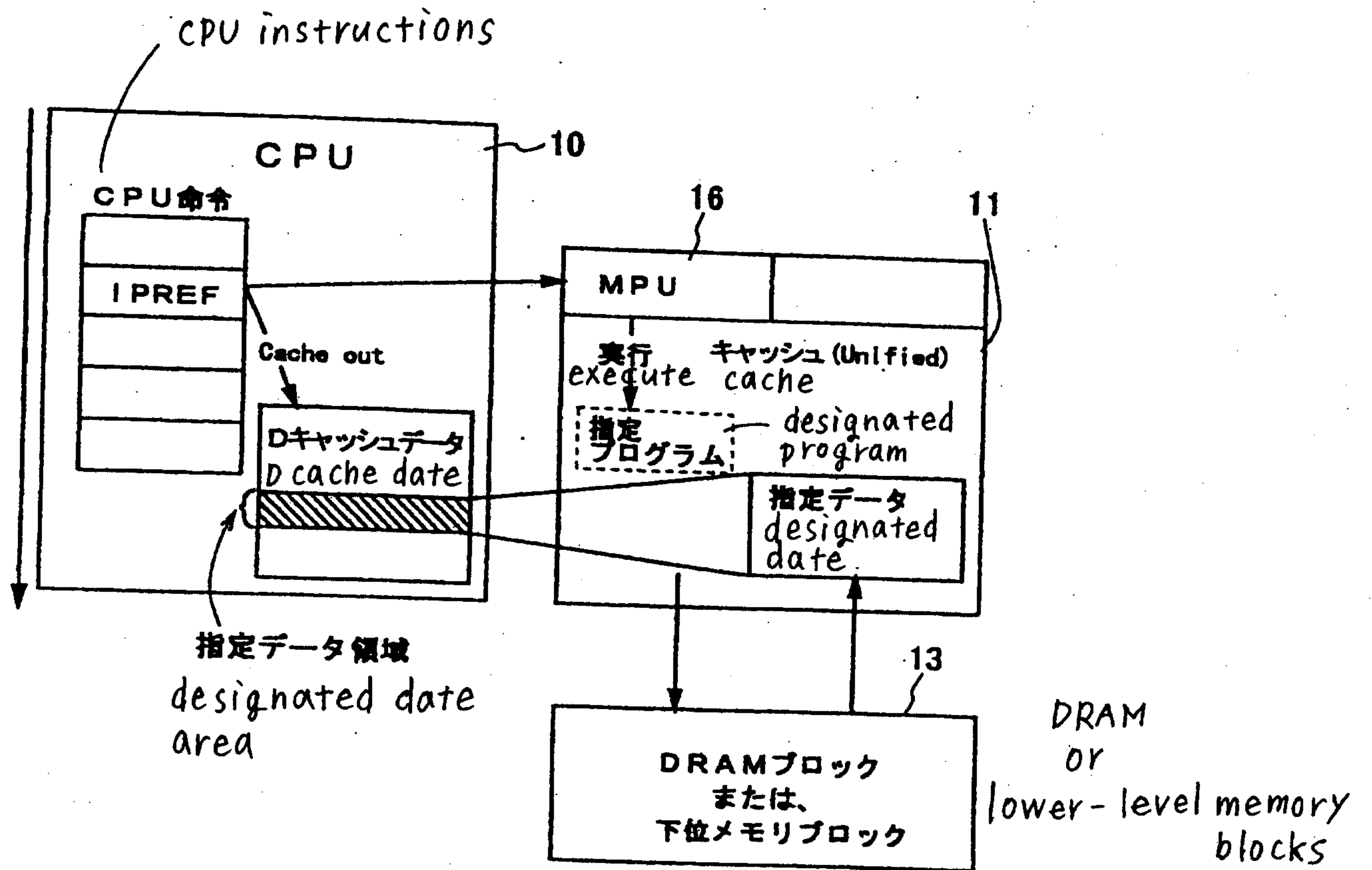
【図 7】

(Fig. 7)



【図 8】

(Fig. 8)



【図9】

[Fig. 9]

