



(19) **United States**

(12) **Patent Application Publication**

**WATANABE et al.**

(10) **Pub. No.: US 2001/0044874 A1**

(43) **Pub. Date: Nov. 22, 2001**

(54) **MEMORY SYSTEM CAPABLE OF SUPPORTING DIFFERENT MEMORY DEVICES AND A MEMORY DEVICE USED THEREFOR**

(22) Filed: **Feb. 11, 1997**

(30) **Foreign Application Priority Data**

Apr. 24, 1996 (JP)..... 8-102584

(76) Inventors: **NAOYA WATANABE, HYOGO (JP); AKIRA YAMAZAKI, HYOGO (JP)**

**Publication Classification**

Correspondence Address:  
**MCDERMOTT WILL & EMERY  
600 13TH STREET, N.W.  
WASHINGTON, DC 20005-3096 (US)**

(51) **Int. Cl.<sup>7</sup> ..... G06F 12/00**  
(52) **U.S. Cl. .... 711/105; 711/154**

(57) **ABSTRACT**

A memory includes an ROM portion storing information specific to the memory, and transfers the stored information to a memory controller via an output buffer and a sink link. The memory controller manages the characteristics of the memory, so that a memory system can be structured utilizing memories of different characteristics.

(\*) Notice: This is a publication of a continued prosecution application (CPA) filed under 37 CFR 1.53(d).

(21) Appl. No.: **08/798,950**

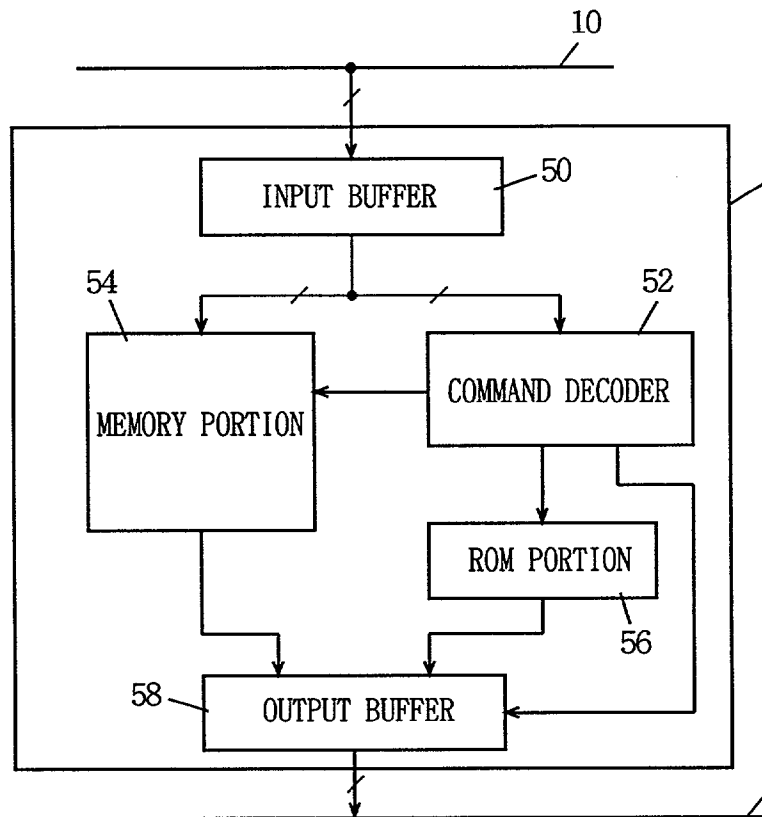


FIG. 1

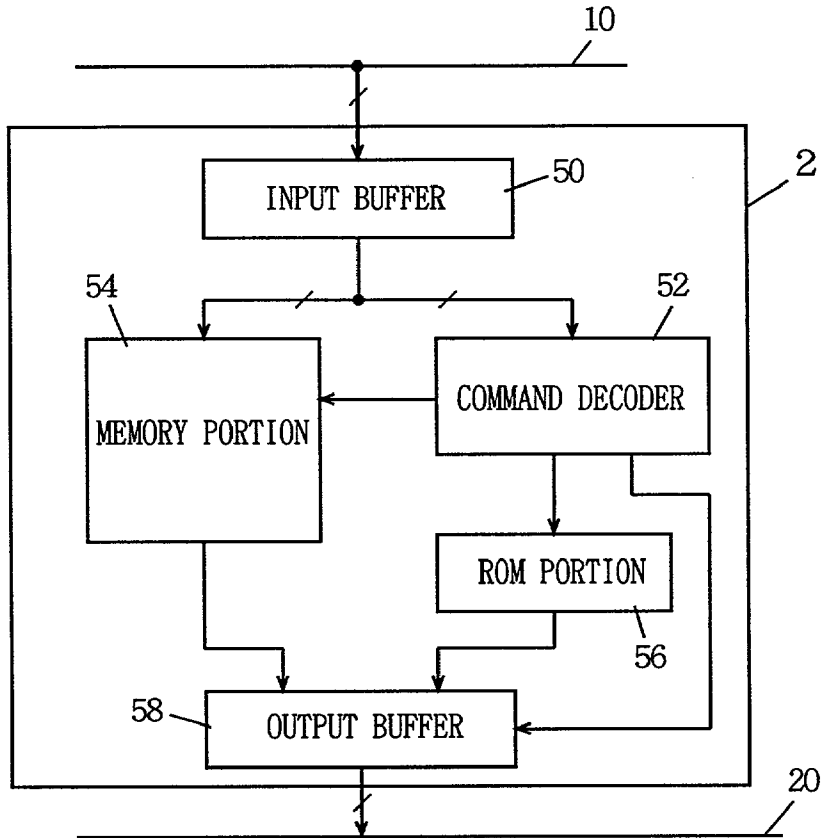


FIG. 2

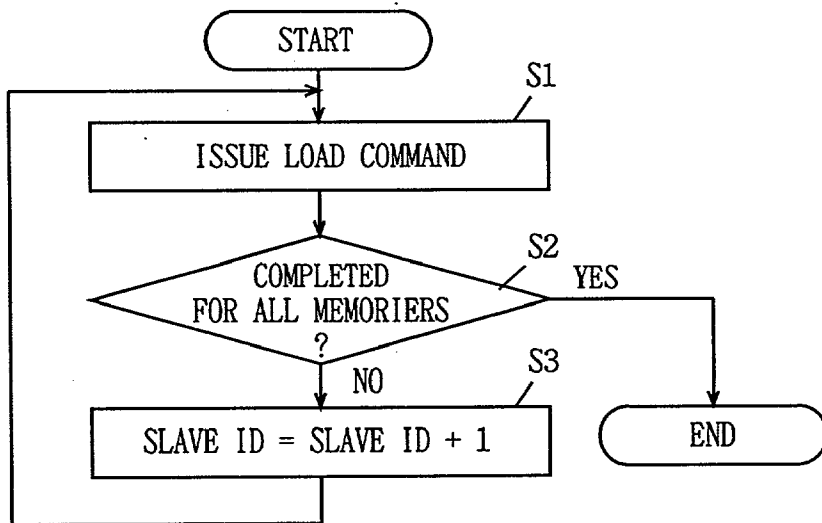




FIG. 5A

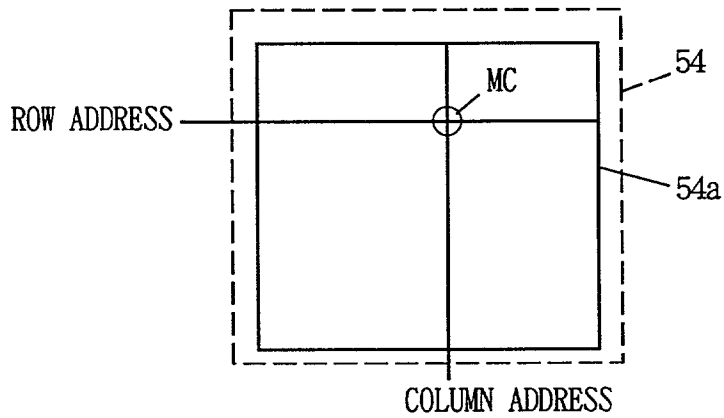


FIG. 5B

ROW ADDRESS : RAO - RA11 | RAO - RA12  
 COLUMN ADDRESS : CAO - CA10 | CAO - CA9

FIG. 6

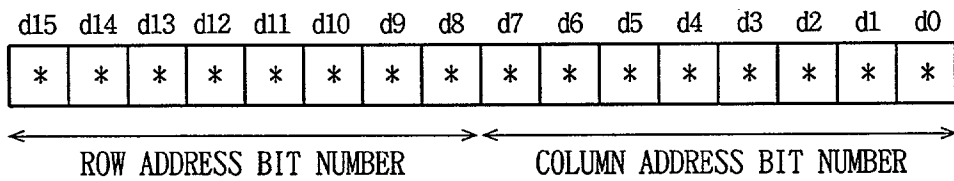


FIG. 7

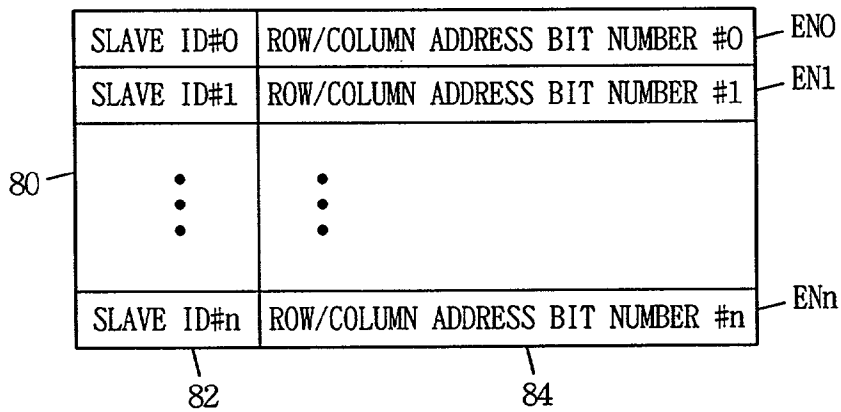


FIG. 8

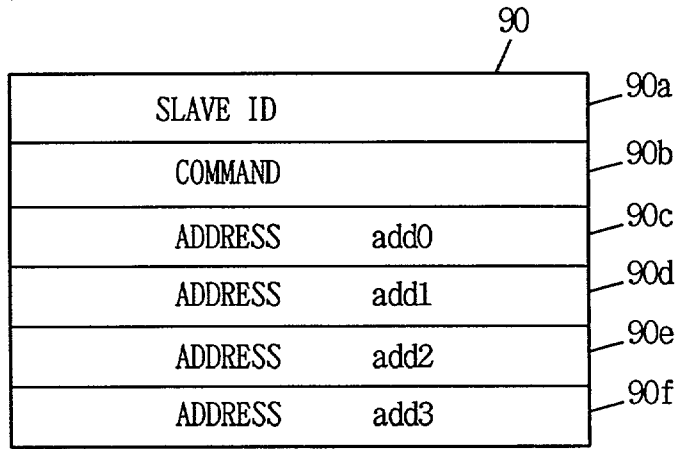


FIG. 9

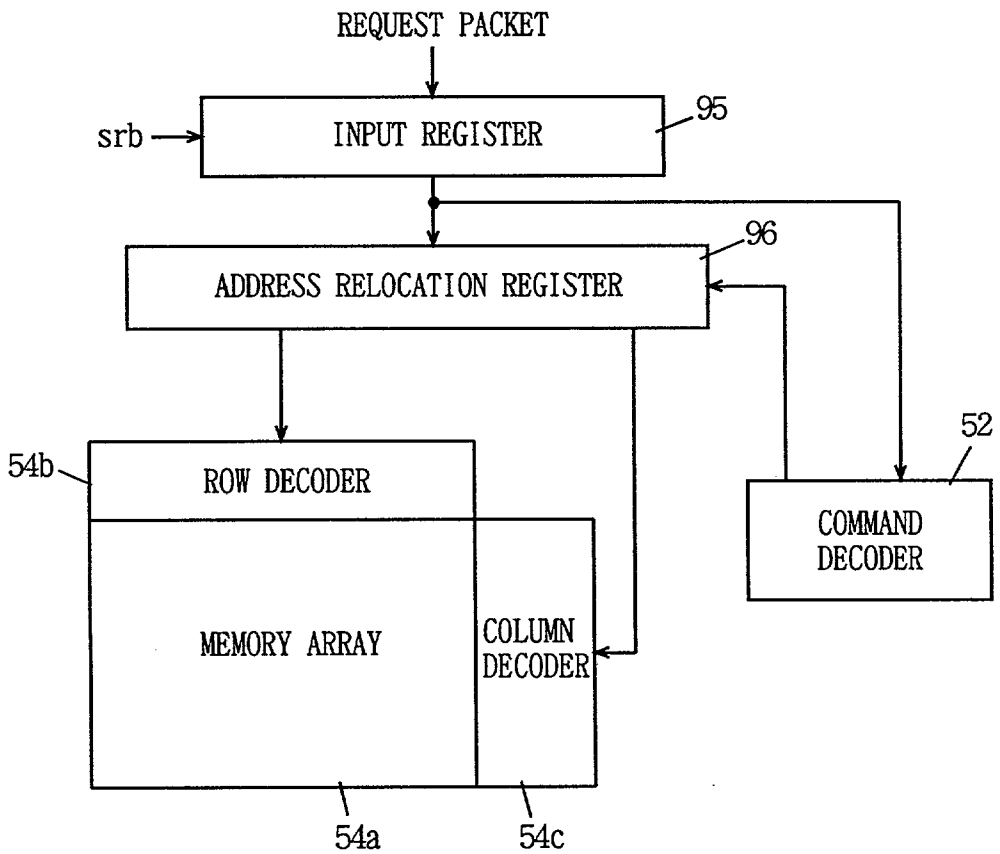


FIG. 10

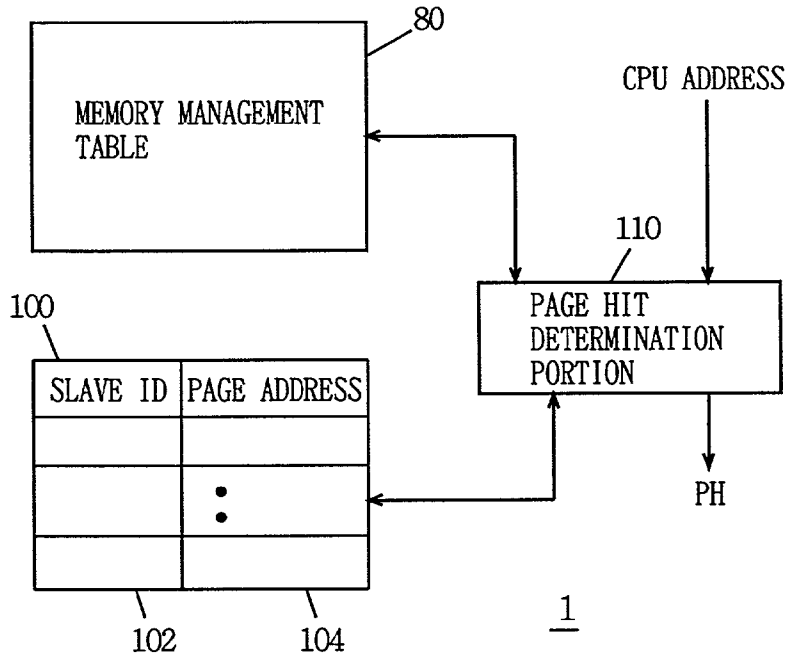
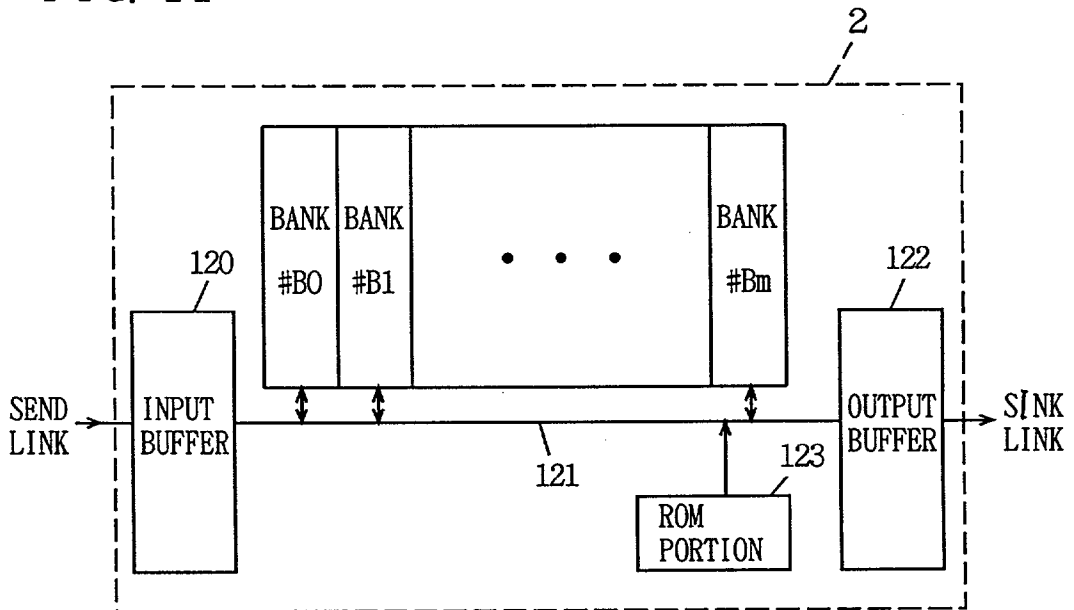


FIG. 11



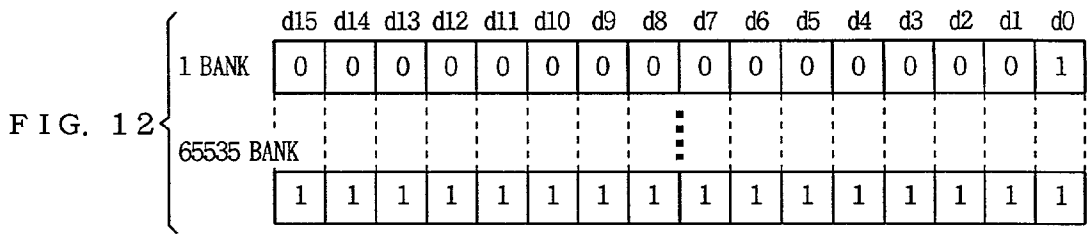


FIG. 13

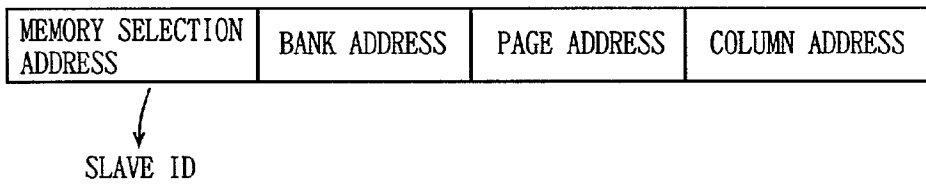


FIG. 14

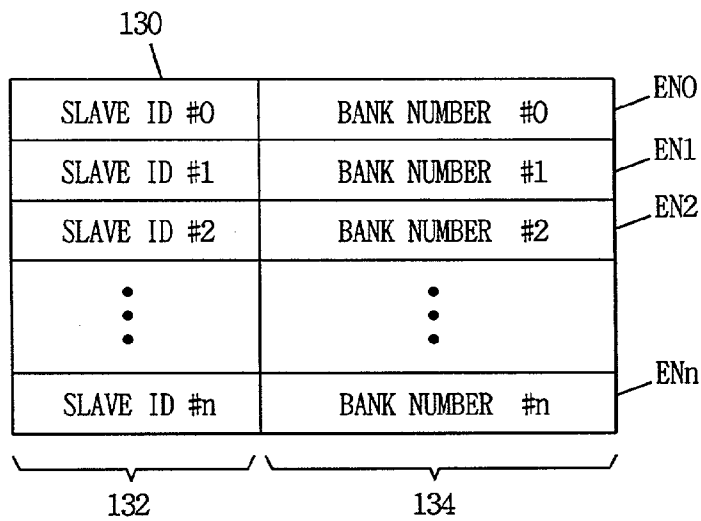


FIG. 15

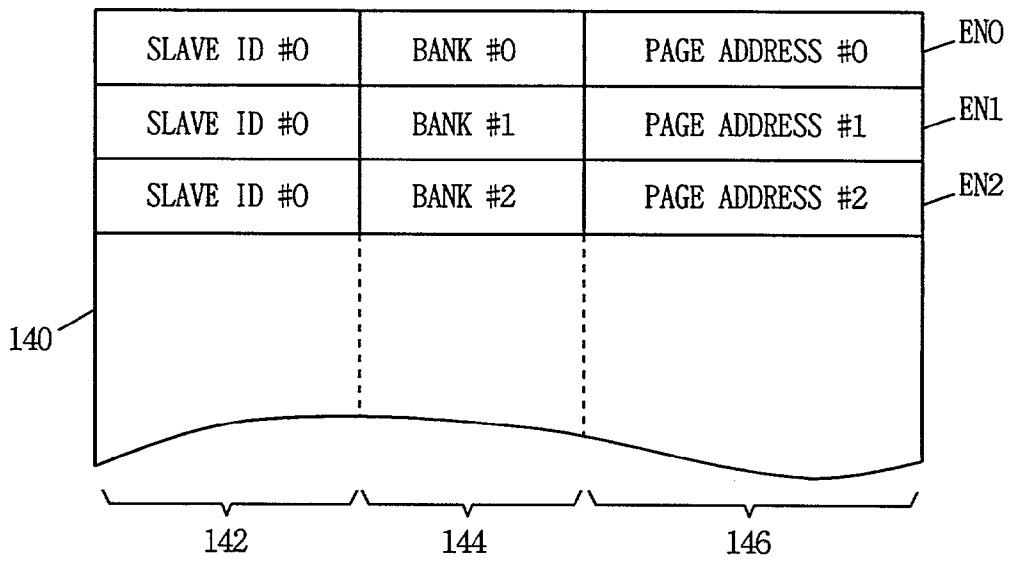
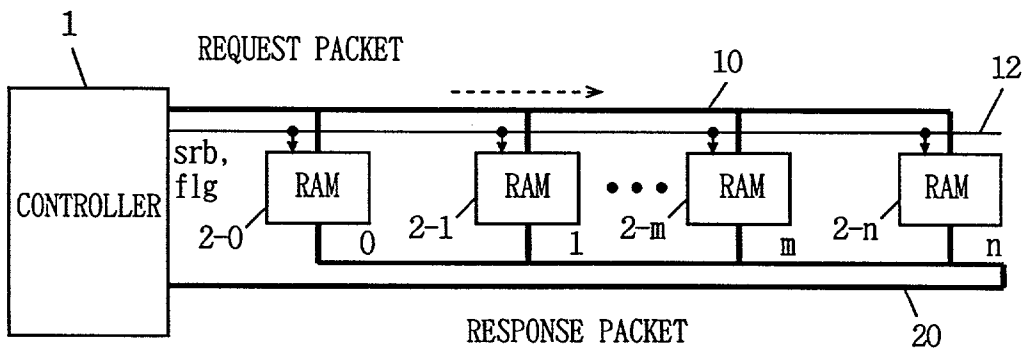


FIG. 16 PRIOR ART





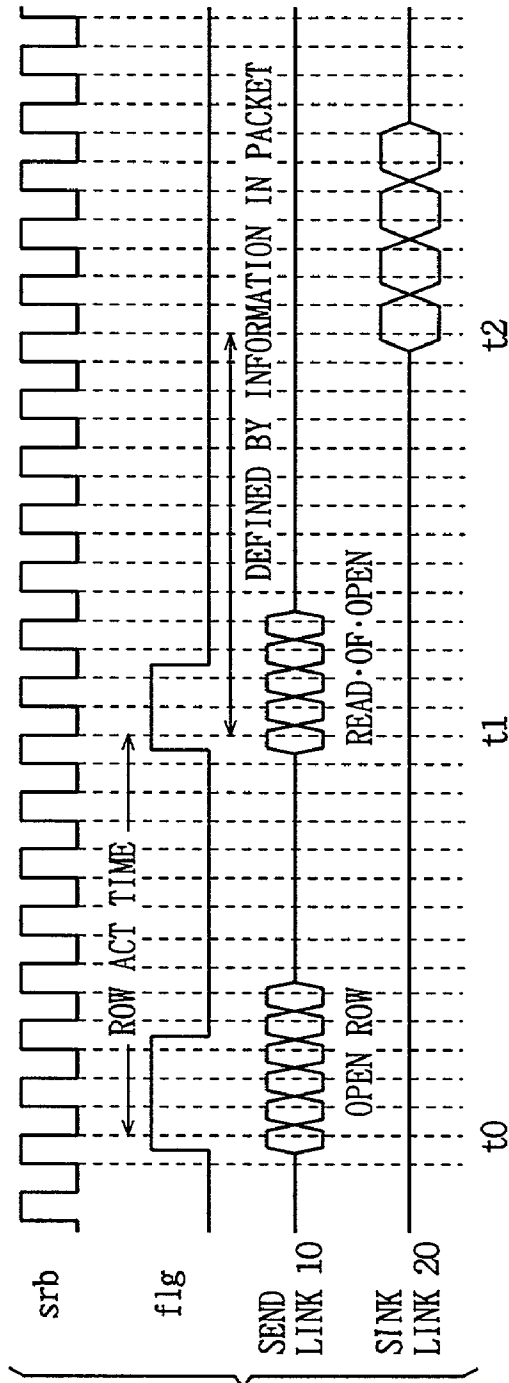


FIG. 17  
PRIOR ART

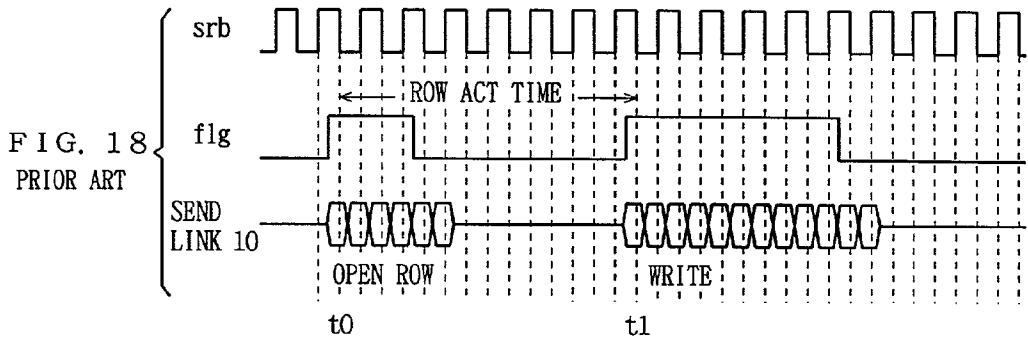


FIG. 19A PRIOR ART

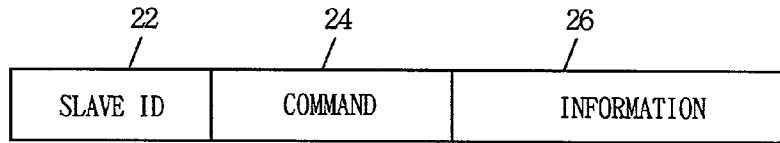
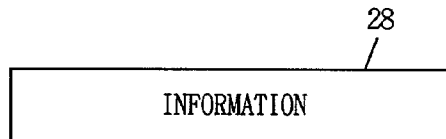


FIG. 19B PRIOR ART



## MEMORY SYSTEM CAPABLE OF SUPPORTING DIFFERENT MEMORY DEVICES AND A MEMORY DEVICE USED THEREFOR

### BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] The present invention relates to a memory system and a semiconductor memory device used therefor, and more particularly relates to a high speed memory system and a semiconductor memory device for the system achieving high speed transfer of a large amount of data.

[0003] 2. Description of the Background Art

[0004] The performance of a microprocessor has been improved, and the storage capacity of a Dynamic Random Access Memory (DRAM) as a memory device is increasing. However, a large amount of data (including instructions) requested by the microprocessor cannot be transferred at high speed from the DRAM to the microprocessor since the operation speed of the DRAM is slower than that of the microprocessor. Therefore, a high speed memory system has been proposed in which a memory controller/processor and a plurality of DRAMs are connected by a bus, and data are consecutively transferred in synchronization with a clock signal. As one example of the high speed memory system, a memory system employing a high speed memory interface referred to as "Sync Link" will be described in the following.

[0005] FIG. 16 is an illustration showing a structure of a general Sync Link memory system. In FIG. 16, the memory system includes: a controller 1; a send link 10 transmitting a request packet output from controller 1; memories (RAMs) 2-0 to 2-n located in parallel and connected to send link 10 in parallel with each other and executing a designated operation according to the request packet supplied via send link 10; a sink link 20 commonly coupled to memories 2-0 to 2-n transmitting a response packet read from a selected memory to controller 1; and a control bus line 12 transmitting a flag flg and a strobe srb which are operation timing signals from controller 1.

[0006] Strobe srb on control signal bus 12 defines an operation speed and an operation timing of controller 1 and memories 2-0 to 2-n, and flag flg indicates the start of a packet transmitted onto send link 10. Send link 10 transmits the request packet from controller 1 in one direction only, while sink link 20 transfers the response packet output from memories 2-0 to 2-n only in one direction toward controller 1. The request packet includes a slave ID (identifier) for identifying each of the memories 2-0 to 2-n, a command which instructs an operation to be executed, and information on address and write data, for example. The response packet transferred onto sink link 20 includes only read data in a normal operation.

[0007] As for the path along which the request packet is transferred from controller 1 to the memories and the response packet is transferred from the memories via sink link 20, the length of the packet transferred path for each of memories 2-0 to 2-n is made equal. Accordingly, sink link 20 includes a portion coupled to memories 2-0 to 2-n transferring the response packet output from a selected memory in the direction away from controller 1, and a portion transferring the response packet in the direction toward controller

1. The packet transfer path of the same distance allows controller 1 to take the same period of time for each of memories 2-0 to 2-n, from outputting the request packet to obtaining the response packet, and synchronized packet transfer is thus easily implemented.

[0008] It is noted that controller 1 may be a processor. In the following description, "memory controller" is used as a term referring to both of a controller controlling the access to memories 2-0 to 2-n and a processor having an operational processing function.

[0009] Send link 10 generally has a width of 8 or 9 bits, and sink link 20 has a bit width two times larger than that of send link 10.

[0010] FIG. 17 is a timing chart at the time of data reading of the memory system. Referring to FIG. 17, a data reading operation will be described.

[0011] At time t0, "open\*row" request is generated. Prior to sending of an open\*row packet at time t0, a flag flg is raised from "0" to "1". Transfer of the packet is instructed by the rise of flag flg. The open\*row packet includes a slave ID (identifier) designating one of memories 2-0 to 2-n, a command indicating the open\*row, and an address designating a row to be opened. In the case of the open\*row, an addressed row in a designated memory 2-i is selected. At this time, only a row select operation is performed, and data in a memory cell connected to the selected row is not output. Therefore, there is no output of a response packet on sink link 20.

[0012] At time t1, a "read\*of\*open" request is output. At time t1, flag flg is also raised from "0" to "1", and transfer of a packet is instructed. The "read\*of\*open" request instructs to select a necessary memory cell out of memory cells connected to the row selected by the open\*row command and to read data. In other words, the read\*of\*open corresponds to an ordinary "page hit" state. The request packet on send link 10 is taken into the addressed memory at both of the rise and fall edges of strobe srb. A time period required for the addressed row to be selected in the addressed memory (corresponding to RAS-CAS delay time tRCD of an ordinary DRAM) is needed between time t0 and time t1.

[0013] According to the read\*of\*open, from the addressed memory, corresponding data in the addressed memory cell is read. The data in the addressed memory cell is sent onto sink link 20 at time t2. The time between time t1 and time t2 is defined by information included in the request packet. The response packet (read data) onto sink link 20 is taken into controller 1 at one of rise and fall of strobe srb.

[0014] The bit width of send link 10 is one half that of sink link 20, while the sampling rate on send link 10 is two times higher than that of sink link 20. The data transfer rate is accordingly the same. The request packet and the response packet are transferred respectively on send link 10 and sink link 20, so that data can be consecutively transferred between the memory controller and the memory by sending the request packet to one memory while sending the response packet to memory controller 1 from another memory.

[0015] FIG. 18 is a timing chart representing an operation at the time of data writing in the memory system shown in

**FIG. 16.** At the time of data writing, transfer of request packet is also indicated by the rise of flag *flg* from "0" to "1" prior to time *t0*. A request packet instructing the open\*row is sent onto send link **10**. An addressed row is selected in an addressed memory by the open\*row.

**[0016]** After an elapse of row act time (*tRCD*), a request packet instructing a write operation is sent at time *t1*. The request packet instructing the write operation includes a slave ID for identifying a memory, write data, a command indicating writing of data, and the number of write data. When the write request packet is sent at time *t1*, data is written to an addressed memory cell (column) on the row selected by the open\*row in the addressed memory. In the case of the write\*request packet, a data packet is not sent onto the sink link since only the writing of data is executed and there is no sending of the response packet.

**[0017]** At the time of data writing access, only the sending of the request packet is performed using send link **10**. Therefore, the response packet can be sent using sink link **20** in parallel with the data writing operation, and high speed data transfer can thus be possible.

**[0018]** **FIGS. 19A and 19B** show structures of packets transmitted and received by the controller. **FIG. 19A** illustrates a request packet. The request packet includes an identifier area **22** storing a slave ID (identifier) for identifying the memory, a command area **24** storing a command instructing an operation to be executed, an information area **26** storing information about, for example, address, time to start a response, number of transfer data byte, and write data. **FIG. 19B** illustrates a structure of a response packet. A response packet **28** is only transmitted according to a request packet and includes only the information which is read data.

**[0019]** As is described above, when information is transmitted in the form of a packet, the size of the area included in each packet is defined. Therefore, the bit number of address information or the like could be constant. The memory controller has no knowledge about information specific to a memory constituting a memory system (size of address bit number of row/column address, storage capacity, and bank number), so that memories employed in the memory system should have the same structure, and a problem of lack of flexibility in structuring the memory system occurs. In other words, if a nonvolatile memory is used in addition to the dynamic random access memory (DRAM) as a memory in the memory system, the memory system cannot be structured when these address configurations are different, and a problem of the lack of flexibility of the system arises.

**[0020]** Further, when the memory system is utilized in a system which processes image data while executing an ordinary operational processing, a memory for storing the image data and a memory for storing data used in the operational processing are often used separately in the memory system. In this case, if respective characteristics of the memory for storing image data and the memory for storing data (instruction and data) used for the operational processing are different, the memory controller cannot acknowledge the characteristics of the individual memories constituting the memory system. As a result, a memory system cannot be structured utilizing memories of different types or characteristics. Accordingly, the use of the memory system is limited and generality of the system is adversely affected.

## SUMMARY OF THE INVENTION

**[0021]** An object of the present invention is to provide a memory system and a semiconductor memory device for the memory system capable of mixedly employing memories having different characteristics.

**[0022]** Another object of the present invention is to provide a semiconductor memory device which can be easily incorporated in a high speed memory system.

**[0023]** A semiconductor memory device according to a first aspect of the invention is provided with circuitry for storing specific information representing inherent characteristics, and an output circuit for transferring the specific information stored in a storing unit onto a bus according to a transfer instruction command supplied via the bus.

**[0024]** A memory system according to a second aspect of the invention includes a memory controller, and a plurality of semiconductor memory devices connected in parallel with each other to the memory controller via first and second buses. Each of the plurality of semiconductor memory devices is provided with a storing unit for storing specific information inherent to the semiconductor memory device, and output circuit for transmitting the specific information stored in the storing unit onto the second bus according to a transfer instruction command supplied via first bus.

**[0025]** Since the information inherent to respective semiconductor memory devices is transferred to the memory controller, the memory controller can manage the specific information for respective memories (semiconductor memory devices), and achieve an efficient address mapping, so that a memory system can easily be structured under the management of the memory controller even if the memories have different characteristics.

**[0026]** The foregoing and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0027]** **FIG. 1** schematically shows an entire structure of a memory according to the present invention.

**[0028]** **FIG. 2** is a flow chart representing an operation of a memory controller for the memory shown in **FIG. 1**.

**[0029]** **FIG. 3** schematically shows a structure of a memory management table of the memory controller according to the first embodiment of the invention.

**[0030]** **FIG. 4A** shows an output bit structure of an output buffer in the memory shown in **FIG. 1**, and **FIG. 4B** shows a correspondence between the output bit of the output buffer and storage capacity.

**[0031]** **FIG. 5A** schematically shows a structure of an array portion of the memory in **FIG. 1**, and **FIG. 5B** shows structures of row and column addresses for designating a memory cell.

**[0032]** **FIG. 6** is an illustration showing the manner in which address bit number information is transmitted according to the second embodiment of the present invention.

[0033] FIG. 7 schematically shows a structure of a memory management table provided in the memory controller according to the second embodiment of the invention.

[0034] FIG. 8 schematically shows a structure of a request packet transmitted via a send link from the memory controller to the memory.

[0035] FIG. 9 is a schematic illustration of a structure of a main portion of the memory according to the second embodiment of the present invention.

[0036] FIG. 10 is an illustration showing how the memory management table in the memory controller is utilized, as one example, according to the second embodiment of the invention.

[0037] FIG. 11 schematically shows a structure of a main portion of the memory according to the third embodiment of the invention.

[0038] FIG. 12 is an illustration showing how the bank number information is transferred according to the third embodiment of the present invention.

[0039] FIG. 13 shows a correspondence between a CPU address and a memory address.

[0040] FIG. 14 schematically shows a structure of a memory management table in a memory controller according to the third embodiment of the present invention.

[0041] FIG. 15 is an illustration showing how the bank number information is utilized as one example according to the third embodiment of the invention.

[0042] FIG. 16 is a schematic representation of a structure of a high speed memory system which has been proposed conventionally.

[0043] FIG. 17 shows a sequence of request packet transfer at the time of data reading in the memory system shown in FIG. 16.

[0044] FIG. 18 is a timing chart showing the timing of transfer of a request packet for writing data in the memory system shown in FIG. 16.

[0045] FIGS. 19A and 19B respectively show structures of a request packet and a response packet.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0046] [First Embodiment]

[0047] FIG. 1 schematically shows an entire structure of a memory according to the present invention. In FIG. 1, a memory 2 includes: an input buffer 50 receiving a request packet from send link 10; a command decoder 52 receiving the request packet via input buffer 50, decoding a command included in the request packet, and generating a control signal according to the result of decoding; a memory portion 54 having a plurality of memory cells arrayed in rows and columns and accessing a memory cell addressed according to address information supplied via input buffer 50 under the control of command decoder 52; a ROM portion 56 storing specific information inherent in memory 2; and an output buffer 58 transmitting information of one of memory portion 54 and ROM portion 56 onto sink link 20 at prescribed timing under the control of command decoder 52. ROM

portion 56 reads the stored specific information and supply it to output buffer 58 under the control of command decoder 52.

[0048] Command decoder 52 is activated when a slave ID (identifier) included in the request packet supplied via send link 10 is the same as a slave ID stored in an identifier register (not shown), and decodes the command of the request packet supplied from input buffer 50. The identifier of memory 2 is the one allocated to each of the memories at an initialization sequence, as shown under "RAM" in FIG. 16. The slave ID is incremented one by one starting from zero, starting at the memory nearest to controller 1. The initialization sequence is executed through a path which is not shown in the figure. The initialization sequence of the slave ID includes the following steps.

[0049] An initialization command is supplied via send link 10 and all of the slave IDs of memory 2 are set at an initial value (62). Next, the controller transfers a slave ID setting command and the slave ID onto the send link, and outputs a slave ID input enable signal for a memory adjacent to the controller via a path not shown in FIG. 16. The memory 2-0 nearest to the controller (see FIG. 16) stores data supplied on the send link as its own slave ID when the slave ID enable signal is activated. After the storing of the slave ID, the memory transmits the slave ID input enable signal to an adjacent memory. The memory takes the slave ID supplied on the send link and stores it as its own identifier only when the slave ID input enable signal is supplied. After the storing of the slave ID by the last memory is completed, the memory of the final stage transfers an identifier store completion signal (slave ID input enable signal) to the memory controller. The memory controller thus recognizes that the slave IDs of all of the memories included in the memory system have been set.

[0050] After the initialization sequence, the memory controller reads the specific information included in each memory 2. The structure of the memory system is the same as that shown in FIG. 16. A sequence of reading specific information according to the first embodiment of the present invention is hereinafter described referring to the flow chart shown in FIG. 2.

[0051] After the initialization sequence is completed and all of the memories included in the memory system store respective slave IDs, memory controller (1) sends a load command with slave ID onto send link 10 (Step S1). Receiving the load command, command decoder 52 in a memory designated by slave ID operates to decode the load command, and gives ROM 56 a command to read the stored information. ROM portion 56 reads the stored information and supplies it to output buffer 58 under the control of command decoder 52. Output buffer 58 sends the specific information onto sink link 20 at a prescribed timing under the control of command decoder 52. The specific information sent onto sink link 20 is transmitted to the memory controller.

[0052] Memory controller then determines whether the load command has been issued to each of the memories (Step S2). The determination is made within the memory controller by comparing a slave ID being issued and a slave ID having the maximum value in the memory system. If the load command has not been issued for all of the memories, the memory controller increments slave ID by 1 (Step S3),

and returns to Step S1. When it is determined that the issuance of the load command is completed for all of the memories in Step S2, the load operation of the specific information is completed.

[0053] The memory controller controls information specific to the memory for each slave ID by issuing the load command for all of the memories included in the memory system, and the memory controller can thus flexibly control the memory system according to the specific information.

[0054] FIG. 3 shows a method of controlling in the memory controller when the specific (inherent) information is the one indicating the storage capacity of memory 2. As shown in FIG. 3, memory controller 1 includes a control unit 60 which controls transmission and reception of the request packet and the response packet, and a memory management table 70 for controlling information specific to each of the memories included in the memory system. Memory management table 70 includes an identifier area 72 storing slave IDs for respective memories included in the memory system, a storage capacity area 74 storing the storage capacity of each of the memories, and an address store area 76 storing CPU address space allocated to each of the memories. Slave ID for each of memories 2-0 to 2-n, the storage capacity, and the CPU address space are coupled with each other and constitute one entry, and stored in management table 70. Accordingly, memory management table 70 includes entries EN0-ENn respectively corresponding to memories 2-0 to 2-n provided in the memory system.

[0055] Memory 2-0 identified by slave ID#0 has storage capacity M#0 and CPU address space #0 is allocated to memory 2-0. By referring to memory management table 70, control unit 60 can identify a memory corresponding to an address area to which the processor requests accessing. The control unit 60 thus can easily transfer a requested packet to a memory storing information to which the processor (CPU) requests accessing.

[0056] By the use of memory management table 70 shown in FIG. 3 and the control of specific information for each of the memories, the CPU address space can be allocated according to each storage capacity even if memories 2-0 to 2-n have different storage capacities. A memory system can thus be structured easily when memories of different capacities are utilized.

[0057] A condition required for memories 2-0 to 2-n is that each of them has an input buffer (input interface) coupled to send link 10 and an output buffer (output interface) coupled to sink link 20. As long as the interface condition is fulfilled, a memory system can be structured employing memories of optional characteristics, and the memory controller can flexibly execute allocation of addresses and so on according to the characteristic of each memory (specific information).

[0058] Writing of the storage information as well as the storage capacity information to ROM portion 56 has already been completed before the shipment of memory 2. ROM portion 56 can be comprised of an EPROM (erasable programmable read only memory) as well as an EEPROM referred to as a flash memory as long as it can store information in nonvolatile manner. A masked ROM to which the storage capacity information is written by a specific mask may be utilized.

[0059] FIGS. 4A and 4B show, as one example, the way how the storage capacity information is output. Output

buffer 58 is supposed to have sixteen data output nodes (terminals) d0-d15. It is not necessary for ROM portion 56 to have an output structure of 16 bit width. ROM portion 56 may be of 8 bit width, or 8 bit data from ROM portion 56 may be converted to 16 bit data in output buffer 58 (it is sufficient for 8 bit data to be successively read out from ROM portion 56). As shown in FIG. 4A, output buffer 58 is supposed to output 16 bit data d0-d15 to sink link 20. In this case, suppose that storage capacity of 1M bits is represented by the least significant bit d0 of 1 and residual bits d1-d15 of "0", as shown in FIG. 4B, maximum storage capacity of 63G bits can be represented on the basis of 1M bits. Accordingly, for example, if the storage capacity of memory 2 is 16M bits, information showing the storage capacity is as follows.

[0060]  $d<15:0>=(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)$

[0061] The memory controller can thus allocate CPU address space according to the storage capacity information. The allocation of the address space is appropriately adjusted depending on the bit number that one address includes. When one address stores 16 bits of data, 16M bit storage capacity corresponds to an addresses space of addresses 0 to  $2^{20}-1$ .

[0062] As described above, even if a memory system is structured employing memories of different storage capacities, CPU address space can easily be allocated to each of the memories in a memory controller by utilizing, as specific information, information about the storage capacity of the memory. Therefore, the memory system can be structured utilizing memories having a plurality of different storage capacities.

[0063] [Second Embodiment]

[0064] FIG. 5A schematically shows a structure of a memory array 54a included in memory portion 54 in FIG. 1. In FIG. 5A, memory array 54a includes a plurality of memory cells MCs arrayed in a matrix of rows and columns. In memory array 54a, memory cell MC is designated by a row address and a column address. Row address designates a row in memory array 54a, and column address designates a column in memory array 54a.

[0065] As shown in FIG. 5B, in the memories having the same storage capacity, the bit number of row address and that of column address are different. Memory array 54a is usually divided into a plurality of blocks, and locations of the blocks selected at the same time are often different. In FIG. 5B, address arrangements are exemplary shown in which one memory is provided with row addresses of address bits RA0-RA11 and column addresses of address bits CA0-CA10, and the other memory is provided with row addresses of address bits RA0-RA12 and column addresses of address bits CA0-CA9. When the bit number of row addresses and column addresses are different from memory to memory, the memory controller is informed of the address bit number.

[0066] FIG. 6 illustrates a structure when specific information inherent in the memory is given to the memory controller, according to the second embodiment of the present invention. FIG. 6 also shows, as one example, that the output buffer sends 16 bits d15-d0 onto sink link 20 as shown in FIG. 4A. In this structure, as shown in FIG. 6,

suppose that the upper byte (8 bits) d15-d8 is used as an area storing information about the bit number of row address, and the lower byte d7-d0 is used as an area storing information about the bit number of column address. In this case, the bit number of row/column addresses can be represented respectively in the range of 1 to 255 bits. For example, if row address is 10 bits and column address is 8 bits, the information can be represented as follows.

[0067] D<15:0>=(0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0)

[0068] The information on the address bit number of the memory is stored in a memory management table 80 as shown in FIG. 7. Memory management table 80 includes an identifier store area 82 which stores slave ID for identifying respective memories, and an address bit number store area 84 which stores information on row/column address bit number of respective memories. Entries EN0-ENn are provided corresponding to respective memories. Accordingly, information on row/column address bit number corresponding to respective identifiers are stored.

[0069] FIG. 8 shows a structure of a request packet. In FIG. 8, a request packet 90 includes a field 90a storing slave ID for designating a memory included in the memory system, a field 90b storing a command indicating a process to be executed, and fields 90c-90f storing addresses add0-add3. Fields 90a-90f are respectively transferred sequentially in synchronization with the rise and fall of strobe srb. The bit number of fields 90a-90f is constant (determined according to the bit width of the send link). In this case, addresses are distributed in the memory as follows.

[0070] FIG. 9 illustrates a structure of a main portion of the memory. In FIG. 9, the memory includes: an input register 95 successively storing the request packet in synchronization with strobe srb; command decoder 52 decoding a command and generating a necessary control signal when the memory is addressed, according to slave ID and the command supplied from input register 95; an address relocation register 96 successively storing the address supplied from input register 95 under the control of command decoder 52; a row decoder 54b decoding the row address supplied from address relocation register 96 and selecting a row in memory array 54a; and a column decoder 54c decoding the column address from address relocation register 96 and selecting a column in memory array 54a.

[0071] Address relocation register 96 is activated by command decoder 52, and stores addresses supplied from input register 95 successively from, for example, upper bits. A prescribed bit number in address relocation register 96 (row address bit number of the memory) is provided to row decoder 54b, and residual address bits are supplied to column decoder 54c. Row decoder 54b and column decoder 54c are activated/deactivated by the control signal from command decoder 52 (the path is not shown).

[0072] Accordingly, the use of address relocation register 96 allows addresses included in the request packet transmitted from the memory controller to be distributed easily to suitable row/column addresses in the memory.

[0073] At the time of the page hit (read•of•open) operation, only the column address is provided. In this case, command decoder 52 causes address relocation register 96

to store an address supplied from input register 95 only in the portion corresponding to the column address.

[0074] Using the structure shown in FIG. 9, a memory system can be structured employing memories of different address bit numbers.

[0075] FIG. 10 illustrates a structure of a main portion of a memory controller. In FIG. 10, memory controller 1 includes memory management table 80, a page memory 100 storing addresses of pages (rows) selected respectively in the memories of the memory system, and a page hit determination portion 110 receiving an address from a processor (CPU) and determining whether the selected page is addressed or not referring to memory management table 80 and page memory 100. The memory is supposed to be a dynamic random access memory (DRAM).

[0076] Page memory 100 includes an identifier area 102 storing slave ID specifying each of the memories, and a page address area 104 storing a page address (row address) selected in each memory. Page hit determination portion 110 receives an address (CPU address) from the processor and determines which memory has been designated according to the CPU address (the second embodiment may be used), and retrieves information about row/column address bit number of the addressed memory referring to memory management table 80.

[0077] Page hit determination portion 110 extracts a page address (row address) from the CPU address according to the retrieved information on the row address bit number, retrieves corresponding page address from page memory 100, determines whether or not the retrieved page address is coincident with the page address included in the CPU address, and activates/deactivates a page hit instruction signal PH according to the result of the determination. When the page hit is determined, a control unit (not shown) extracts a column address from the CPU address according to information on the bit number of the column address, and generates a request packet.

[0078] By storing in memory management table 80, information on the bit number of the row/column address in respective memories, page hit/miss can be easily determined even if the address structures of the memories included in the memory system are different. Based on the result of the determination, an address necessary for generating a packet can easily be produced from the CPU address.

[0079] Accordingly, even if the memories have different address configurations, page mode access (open-row-read/write) can be performed according to the address configurations, and the memory access can thus be achieved efficiently.

[0080] [Third Embodiment]

[0081] FIG. 11 shows a structure of a main portion of a memory according to the third embodiment of the invention. In FIG. 11, memory 2 includes an input buffer 120 receiving a request packet on the send link, banks #B0-#Bm commonly coupled to an internal bus 121 and independently driven to active/inactive state, and an output buffer 122 coupled to internal data bus 121 and outputs information supplied via internal data bus 121 onto sink link as a response packet.

[0082] Banks #B0-#Bm are selectively driven to active/inactive state independently of each other according to a bank address under the control of a command decoder (not shown). The number of banks #B0-#Bm included in memory 2 is appropriately determined. Memory 2 includes a ROM portion 123 which stores the number of banks #B0-#Bm. ROM portion 123 corresponds to ROM portion 56 in FIG. 1. Information on the bank number stored in ROM portion 123 is transferred onto the sink link via output buffer 122, and transmitted to the memory controller.

[0083] FIG. 12 shows a format of the information on the bank number output from output buffer 122. In FIG. 12, output buffer 122 transmits 16 bits d15-d0 onto the sink link. In this case, the number of the banks can be designated in the range of 1 to 65535 as shown in FIG. 12.

[0084] FIG. 13 shows a correspondence between a CPU address and an address output by the memory controller to the memory. As shown in FIG. 13, the CPU address includes a memory selection address designating a memory included in the memory system, a bank address specifying a bank included in the memory, a page address specifying a page (row) included in the bank, and a column address specifying a column on the page. The bit numbers of the bank addresses are different if the numbers of the banks are different. In this case, the bit number of the page addresses varies, accordingly. The memory selection address is correlated to slave ID.

[0085] FIG. 14 shows, as one example, a structure of a memory management table 130 included in the memory controller. As shown in FIG. 14, memory management table 130 included in memory controller 1 includes an identifier area 132 storing slave ID (slave ID#0-slave ID#n) specifying each of the memories, and a bank number information store area 134 storing information on the bank numbers #0-#n in each of the memories. Slave ID specifying a memory and the bank number information indicating the number of the banks included in the memory are stored in one entry EN (EN0-ENn). The memory controller controls the number of the banks information of each memory by referring to memory management table 130.

[0086] The memory executes an operation of memory cell selection according to the bank address, the page address, and the column address supplied from the memory controller. As the structure of the memory, the structure of FIG. 9 may be utilized, and the addresses supplied from input register 95 are divided into the bank address, the page address, and the column address by address relocation register 96.

[0087] FIG. 15 shows a structure of a page memory. In FIG. 15, a page memory 140 includes an identifier area 142 storing slave ID, a bank store area 144 specifying a bank provided corresponding to each slave ID, and a page address area 146 storing a page address indicating a selected page respective banks. The memory controller secures an area for storing a page address of each bank in page memory 140 according to information on the bank number stored in memory management table 130. In FIG. 15, banks #0, #1 and #2 relating to slave ID #0 designating memory 2-0, as well as page addresses #0, #1 and #2 stored for respective banks are representatively shown.

[0088] As a structure of the page hit determination portion, a structure similar to that shown in FIG. 10 may be

utilized. Page memory 140 shown in FIG. 15 is used in place of page memory 100. In this case, the operation of the page hit determination portion is slightly different. Page hit determination portion 110 identifies the bank address bit from the CPU address according to information on the bank number stored in memory management table 130, reads corresponding page address from page memory 140 using the memory selection address and the bank address as reference addresses, and extracts a page address from the CPU address. Page hit/miss is determined by deciding whether the page address of the CPU address coincides or does not coincide with the page address read from page memory 140.

[0089] When the request packet is sent, it is not necessary for the memory controller to distinguish the bank address from the page address. The reason is that the bank address and the page address are separated in memory 2.

[0090] As described above, even if the bank numbers of the memories included in the memory system are different, the memory controller can determine the page hit since the storage area is provided for storing information about the number of banks in the memory, and the bank number information is transferred to the memory controller, controlling the bank numbers of respective memories accurately. An efficient access can thus be achieved.

[0091] If the page memories 100 and 140 are constituted by a content addressable memory (CAM), the page hit determination portion need not discriminate the bank address from the page address. What is necessary for the page hit determination portion is simply to extract addresses including a memory selection address, a bank address, and a page address from CPU addresses. However, the location of the column address should be the same even if the numbers of the banks are different.

[0092] As for the memory, the memory may be the one in which random access is possible even if the page mode access cannot be executed, such as a dynamic random access memory, or a memory device of nonvolatile type (for example, a flash memory). A necessary condition of the memory is that it has an input port (input register or input buffer) coupled to a send link and an output port (output buffer or output register) coupled to a sink link.

[0093] As for the memory system, the present invention can be applied to a memory system if the memory system is structured such that the information is transferred according to a command from the memory controller, even if the system is not structured to be a bus structure including a send link and a sink link.

[0094] According to the present invention, a memory can be incorporated into a memory system if the memory is provided with input/output interface of the memory system, owing to storage means provided for storing information specific to the memory. Therefore, even if the memories of different standards are employed, a memory system can be structured. In other words, a memory system can be structured by selecting a memory suitable for the application of the memory system.

[0095] The memory controller can easily achieve allocation of CPU address space, as well as page hit/miss determination according to the characteristics of memories by



utilizing information on address bit number, storage capacity, and bank number as information specific to the memories.

[0096] Although the present invention has been described and illustrated in detail, it is clearly understood that the same is by way of illustration and example only and is not to be taken by way of limitation, the spirit and scope of the present invention being limited only by the terms of the appended claims.

What is claimed is:

1. A semiconductor memory device operating according to a command supplied via a first bus, comprising:

store means for storing specific information inherent in said semiconductor memory device; and

output means for sending the specific information stored in said store means onto a second bus according to a transfer command supplied via said first bus.

2. The semiconductor memory device according to claim 1, wherein

said specific information indicates a storage capacity of said semiconductor memory device.

3. The semiconductor memory device according to claim 1, further comprising a memory array having a plurality of memory cells arrayed in rows and columns and addressed by a multi-bit row address and a multi-bit column address, wherein

said specific information indicates bit number of said multi-bit row address and said multi-bit column address.

4. The semiconductor memory device according to claim 1, further comprising a plurality of banks operating independently of each other, wherein

said specific information indicates number of said banks.

5. The semiconductor memory device according to claim 1, wherein

said second bus is provided separately from said first bus transferring said command.

6. A memory system comprising:

a memory controller; and

a plurality of semiconductor memory devices connected in parallel with each other to said memory controller via first and second buses respectively transferring information in one direction, and operating according to a command transmitted via said first bus from said memory controller, wherein

each of said semiconductor memory devices includes

store means for storing specific information inherent in said semiconductor memory device, and

output means for transferring the specific information stored in said store means to said memory controller via said second bus, according to a transfer command supplied from said memory controller via said first bus.

7. The memory system according to claim 6, wherein said specific information indicates a storage capacity of corresponding semiconductor memory device.

8. The memory system according to claim 6, wherein

each of said semiconductor memory devices includes a memory array having a plurality of memory cells respectively addressed by a multibit row address and multibit column address, and

said specific information indicates bit number of said multibit row address and said multibit column address of said corresponding semiconductor memory device.

9. The memory system according to claim 6, wherein each of said plurality of semiconductor memory devices includes at least one bank, and when the semiconductor memory device has a plurality of banks, each of said plurality of banks can be driven to active and inactive states independently of each other, and

said specific information indicates number of banks included in said corresponding semiconductor memory device.

10. The memory system according to claim 6, wherein

said bus includes a first link transferring said command, and a second link provided separately from said first link and transferring specific information output from said output means to said memory controller.

\* \* \* \* \*