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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 December 2005 (01.12.2005)

PCT

(10) International Publication Number
WO 2005/114427 A2

(51) International Patent Classification[7]: G06F 12/00

(21) International Application Number:
PCT/US2005/015694

(22) International Filing Date: 4 May 2005 (04.05.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/846,988 14 May 2004 (14.05.2004) US

(71) Applicant *(for all designated States except US)*: **MICRON TECHNOLOGY, INC.** [US/US]; 8000 South Federal Way, Boise, ID 83716-9632 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **JEDDELOH, Joseph, M.** [US/US]; 4302 Reiland Lane, Shoreview, MN 55126 (US).

(74) Agents: **ENG, Kimton, N.** et al.; Dorsey & Whitney LLP, 1420 Fifth Avenue, Suite 3400, Seattle, WA 98101 (US).

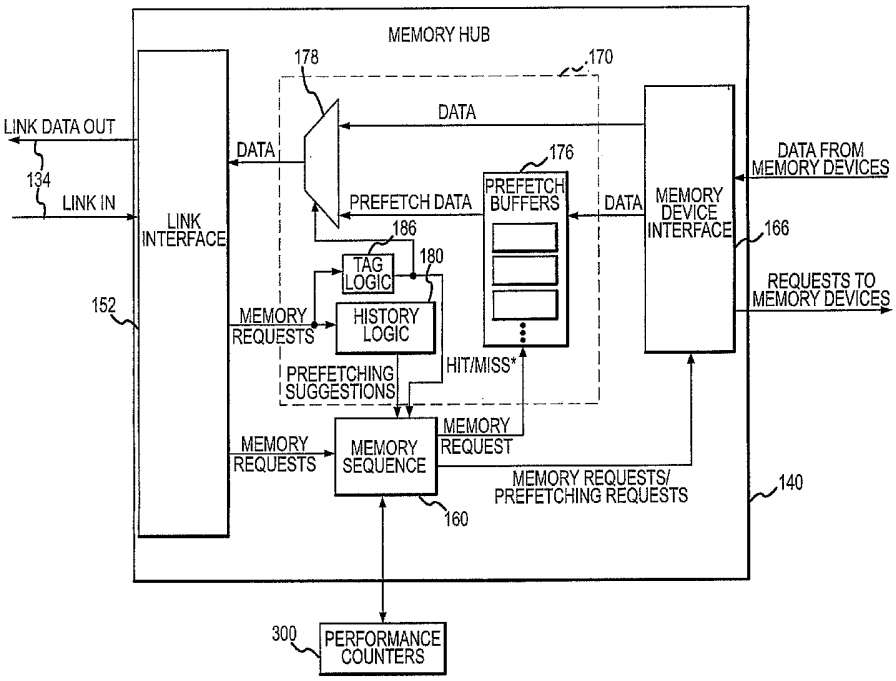
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK,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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: MEMORY HUB AND METHOD FOR MEMORY SEQUENCING



(57) Abstract: A memory module includes a memory hub coupled to several memory devices. The memory hub includes at least one performance counter that tracks one or more system metrics-for example, page hit rate, prefetch hits, and/or cache hit rate. The performance counter communicates with a memory sequencer that adjusts its operation based on the system metrics tracked by the performance counter.

WO 2005/114427 A2

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

MEMORY HUB AND METHOD FOR MEMORY SEQUENCING

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application claims the benefit of the filing date of U.S. Patent Application No. 10/846,988, MEMORY HUB AND METHOD FOR MEMORY SEQUENCING, filed May 14, 2004, which is incorporated herein by reference.

TECHNICAL FIELD

This invention relates to computer systems, and, more particularly, to a computer system having a memory hub coupling several memory devices to a processor or other memory access device.

BACKGROUND OF THE INVENTION

Computer systems use memory devices, such as dynamic random access memory ("DRAM") devices, to store data that are accessed by a processor. These memory devices are normally used as system memory in a computer system. In a typical computer system, the processor communicates with the system memory through a processor bus and a memory controller. The processor issues a memory request, which includes a memory command, such as a read command, and an address designating the location from which data or instructions are to be read. The memory controller uses the command and address to generate appropriate command signals as well as row and column addresses, which are applied to the system memory. In response to the commands and addresses, data are transferred between the system memory and the processor. The memory controller is often part of a system controller, which also includes bus bridge circuitry for coupling the processor bus to an expansion bus, such as a PCI bus.

Although the operating speed of memory devices has continuously increased, this increase in operating speed has not kept pace with increases in the operating speed of processors. Even slower has been the increase in operating speed of

memory controllers coupling processors to memory devices. The relatively slow speed of memory controllers and memory devices limits the data bandwidth between the processor and the memory devices.

In addition to the limited bandwidth between processors and memory devices, the performance of computer systems is also limited by latency problems that increase the time required to read data from system memory devices. More specifically, when a memory device read command is coupled to a system memory device, such as a synchronous DRAM ("SDRAM") device, the read data are output from the SDRAM device only after a delay of several clock periods. Therefore, although SDRAM devices can synchronously output burst data at a high data rate, the delay in initially providing the data can significantly slow the operating speed of a computer system using such SDRAM devices.

One approach to alleviating the memory latency problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory hub architecture, a system controller or memory controller is coupled to several memory modules, each of which includes a memory hub coupled to several memory devices. The memory hub efficiently routes memory requests and responses between the controller and the memory devices. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory device while another memory device is responding to a prior memory access. For example, the processor can output write data to one of the memory devices in the system while another memory device in the system is preparing to provide read data to the processor.

Although computer systems using memory hubs may provide superior performance, they nevertheless often fail to operate at optimum speed for several reasons. For example, even though memory hubs can provide computer systems with a greater memory bandwidth, they still suffer from latency problems of the type described above. More specifically, although the processor may communicate with one memory device while another memory device is preparing to transfer data, it is sometimes necessary to receive data from one memory device before the data from another memory device can be used. In the event data must be received from one memory device before

data received from another memory device can be used, the latency problem continues to slow the operating speed of such computer systems.

One technique that has been used to reduce latency in memory devices is to prefetch data, *i.e.*, read data from system memory before the data are requested by a program being executed. Generally the data that are to be prefetched are selected based on a pattern of previously fetched data. The pattern may be as simple as a sequence of addresses from which data are fetched so that data can be fetched from subsequent addresses in the sequence before the data are needed by the program being executed. The pattern, which is known as a "stride," may, of course, be more complex.

Further, even though memory hubs can provide computer systems with a greater memory bandwidth, they still suffer from throughput problems. For example, before data can be read from a particular row of memory cells, that digit lines in the array are typically precharged by equilibrating the digit lines in the array. The particular row is then opened by coupling the memory cells in the row to a digit line in respective columns. A respective sense amplifier coupled between the digit lines in each column then responds to a change in voltage corresponding to the data stored in respective memory cell. Once the row has been opened, data can be coupled from each column of the open row by coupling the digit lines to a data read path. Opening a row, also referred to as a page, therefore consumes a finite amount of time and places a limit on the memory throughput.

Finally, the optimal decision of whether or not to prefetch data (and which data to prefetch), as well as whether or not to precharge or open a row, and whether or not to cache accessed data, may change over time and vary as a function of an application being executed by a processor that is coupled to the memory hub.

There is therefore a need for a computer architecture that provides the advantages of a memory hub architecture and also minimize the latency and/or throughput problems common in such systems, thereby providing memory devices with high bandwidth, high throughput, and low latency. Such a system would also desirably allow the operation of the memory hub to change over time.

SUMMARY OF THE INVENTION

According to one aspect of the invention, a memory module and method is provided including a plurality of memory devices and a memory hub. The memory hub contains a link interface, such as an optical input/output port, that receives memory requests for access to memory cells in at least one of the memory devices. The memory hub further contains a memory device interface coupled to the memory devices, the memory device interface being operable to couple memory requests to the memory devices for access to memory cells in at least one of the memory devices and to receive read data responsive to at least some of the memory requests. The memory hub further contains a performance counter coupled to the memory device interface, the performance counter operable to track at least one metric selected from the group consisting of page hit rate, prefetch hits, and cache hit rate. The memory hub further contains a memory sequencer coupled to the link interface and the memory device interface. The memory sequencer is operable to couple memory requests to the memory device interface responsive to memory requests received from the link interface. The memory sequencer is further operable to dynamically adjust operability responsive to the performance counter. Forexample, the performance counter may track page hit rate and the memory sequencer may change a number of open pages in the memory device or switch to an auto-precharge mode responsive to the tracked page hit rate. Alternatively, the performance counter may track a percentage of prefetch hits, and the memory sequencer may enable prefetching or disable prefetching or adjust the number of prefetch requests as a function of the tracked prefetch hit percentage. As a further example, the performance counter may track a cache hit rate, and the memory sequencer may disable the cache as a function of the tracked cache hit rate.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system according to one example of the invention in which a memory hub is included in each of a plurality of memory modules.

Figure 2 is a block diagram of a memory hub used in the computer system of Figure 1, which contains performance counters according to one example of the invention.

Figure 3 is a block diagram of a memory hub used in the computer system of Figure 1, which contains prefetch buffers according to one example of the invention.

DETAILED DESCRIPTION OF THE INVENTION

A computer system 100 according to one example of the invention is shown in Figure 1. The computer system 100 includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 includes a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is typically coupled to cache memory 108, which, as previously mentioned, is usually static random access memory ("SRAM"). Finally, the processor bus 106 is coupled to a system controller 110, which is also sometimes referred to as a "North Bridge" or "memory controller."

The system controller 110 serves as a communications path to the processor 104 for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or more input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

The system controller 110 is coupled to several memory modules 130a,b...n, which serve as system memory for the computer system 100. The memory modules 130 are preferably coupled to the system controller 110 through a high-speed link 134, which may be an optical or electrical communication path or some other type of communications path. In the event the high-speed link 134 is implemented as an optical communication path, the optical communication path may be in the form of one or more optical fibers, for example. In such case, the system controller 110 and the memory modules will include an optical input/output port or separate input and output ports coupled to the optical communication path. The memory modules 130 are shown coupled to the system controller 110 in a multi-drop arrangement in which the single high-speed link 134 is coupled to all of the memory modules 130. However, it will be understood that other topologies may also be used, such as a point-to-point coupling arrangement in which a separate high-speed link (not shown) is used to couple each of the memory modules 130 to the system controller 110. A switching topology may also be used in which the system controller 110 is selectively coupled to each of the memory modules 130 through a switch (not shown). Other topologies that may be used will be apparent to one skilled in the art.

Each of the memory modules 130 includes a memory hub 140 for controlling access to 32 memory devices 148, which, in the example illustrated in Figure 1, are synchronous dynamic random access memory ("SDRAM") devices. However, a fewer or greater number of memory devices 148 may be used, and memory devices other than SDRAM devices may, of course, also be used. In the example illustrated in Figure 1, the memory hubs 140 communicate over 4 independent memory channels 149 over the high-speed link 134. In this example, although not shown in Figure 1, 4 memory hub controllers 128 are provided, each to receive data from one memory channel 149. A fewer or greater number of memory channels 149 may be used, however, in other examples. The memory hub 140 is coupled to each of the system memory devices 148 through a bus system 150, which normally includes a control bus, an address bus and a data bus.

A memory hub 200 according to an embodiment of the present invention is shown in Figure 2. The memory hub 200 can be substituted for the memory hub 140 of Figure 1. The memory hub 200 is shown in Figure 2 as being coupled to four memory devices 240a-d, which, in the present example are conventional SDRAM devices. In an alternative embodiment, the memory hub 200 is coupled to four different banks of memory devices, rather than merely four different memory devices 240a-d, with each bank typically having a plurality of memory devices. However, for the purpose of providing an example, the present description will be with reference to the memory hub 200 coupled to the four memory devices 240a-d. It will be appreciated that the necessary modifications to the memory hub 200 to accommodate multiple banks of memory is within the knowledge of those ordinarily skilled in the art.

Further included in the memory hub 200 are link interfaces 210a-d and 212a-d for coupling the memory module on which the memory hub 200 is located to a first high speed data link 220 and a second high speed data link 222, respectively. As previously discussed with respect to Figure 1, the high speed data links 220, 222 can be implemented using an optical or electrical communication path or some other type of communication path. The link interfaces 210a-d, 212a-d are conventional, and include circuitry used for transferring data, command, and address information to and from the high speed data links 220, 222. As well known, such circuitry includes transmitter and receiver logic known in the art. It will be appreciated that those ordinarily skilled in the art have sufficient understanding to modify the link interfaces 210a-d, 212a-d to be used with specific types of communication paths, and that such modifications to the link interfaces 210a-d, 212a-d can be made without departing from the scope of the present invention. For example, in the event the high-speed data link 220, 222 is implemented using an optical communications path, the link interfaces 210a-d, 212a-d will include an optical input/output port that can convert optical signals coupled through the optical communications path into electrical signals.

The link interfaces 210a-d, 212a-d are coupled to the a switch 260 through a plurality of bus and signal lines, represented by busses 214. The busses 214 are conventional, and include a write data bus and a read data bus, although a single bi-

directional data bus may alternatively be provided to couple data in both directions through the link interfaces 210a-d, 212a-d. It will be appreciated by those ordinarily skilled in the art that the busses 214 are provided by way of example, and that the busses 214 may include fewer or greater signal lines, such as further including a request line and a snoop line, which can be used for maintaining cache coherency.

The link interfaces 210a-d, 212a-d include circuitry that allow the memory hub 200 to be connected in the system memory in a variety of configurations. For example, the multi-drop arrangement, as shown in Figure 1, can be implemented by coupling each memory module to the memory hub controller 128 through either the link interfaces 210a-d or 212a-d. Alternatively, a point-to-point, or daisy chain configuration can be implemented by coupling the memory modules in series. For example, the link interfaces 210a-d can be used to couple a first memory module and the link interfaces 212a-d can be used to couple a second memory module. The memory module coupled to a processor, or system controller, will be coupled thereto through one set of the link interfaces and further coupled to another memory module through the other set of link interfaces. In one embodiment of the present invention, the memory hub 200 of a memory module is coupled to the processor in a point-to-point arrangement in which there are no other devices coupled to the connection between the processor 104 and the memory hub 200. This type of interconnection provides better signal coupling between the processor 104 and the memory hub 200 for several reasons, including relatively low capacitance, relatively few line discontinuities to reflect signals and relatively short signal paths.

The switch 260 is further coupled to four memory interfaces 270a-d which are, in turn, coupled to the system memory devices 240a-d, respectively. By providing a separate and independent memory interface 270a-d for each system memory device 240a-d, respectively, the memory hub 200 avoids bus or memory bank conflicts that typically occur with single channel memory architectures. The switch 260 is coupled to each memory interface through a plurality of bus and signal lines, represented by busses 274. The busses 274 include a write data bus, a read data bus, and a request line. However, it will be understood that a single bi-directional data bus

may alternatively be used instead of a separate write data bus and read data bus. Moreover, the busses 274 can include a greater or lesser number of signal lines than those previously described.

In an embodiment of the present invention, each memory interface 270a-d is specially adapted to the system memory devices 240a-d to which it is coupled. More specifically, each memory interface 270a-d is specially adapted to provide and receive the specific signals received and generated, respectively, by the system memory device 240a-d to which it is coupled. Also, the memory interfaces 270a-d are capable of operating with system memory devices 240a-d operating at different clock frequencies. As a result, the memory interfaces 270a-d isolate the processor 104 from changes that may occur at the interface between the memory hub 230 and memory devices 240a-d coupled to the memory hub 200, and it provides a more controlled environment to which the memory devices 240a-d may interface.

The switch 260 coupling the link interfaces 210a-d, 212a-d and the memory interfaces 270a-d can be any of a variety of conventional or hereinafter developed switches. For example, the switch 260 may be a cross-bar switch that can simultaneously couple link interfaces 210a-d, 212a-d and the memory interfaces 270a-d to each other in a variety of arrangements. The switch 260 can also be a set of multiplexers that do not provide the same level of connectivity as a cross-bar switch but nevertheless can couple the some or all of the link interfaces 210a-d, 212a-d to each of the memory interfaces 270a-d. The switch 260 may also includes arbitration logic (not shown) to determine which memory accesses should receive priority over other memory accesses. Bus arbitration performing this function is well known to one skilled in the art.

With further reference to Figure 2, each of the memory interfaces 270a-d includes a respective memory controller 280, a respective write buffer 282, and a respective cache memory unit 284. The memory controller 280 performs the same functions as a conventional memory controller by providing control, address and data signals to the system memory device 240a-d to which it is coupled and receiving data signals from the system memory device 240a-d to which it is coupled. The write buffer

282 and the cache memory unit 284 include the normal components of a buffer and cache memory, including a tag memory, a data memory, a comparator, and the like, as is well known in the art. The memory devices used in the write buffer 282 and the cache memory unit 284 may be either DRAM devices, static random access memory ("SRAM") devices, other types of memory devices, or a combination of all three. Furthermore, any or all of these memory devices as well as the other components used in the cache memory unit 284 may be either embedded or stand-alone devices.

The write buffer 282 in each memory interface 270a-d is used to store write requests while a read request is being serviced. In a such a system, the processor 104 can issue a write request to a system memory device 240a-d even if the memory device to which the write request is directed is busy servicing a prior write or read request. Using this approach, memory requests can be serviced out of order since an earlier write request can be stored in the write buffer 282 while a subsequent read request is being serviced. The ability to buffer write requests to allow a read request to be serviced can greatly reduce memory read latency since read requests can be given first priority regardless of their chronological order. For example, a series of write requests interspersed with read requests can be stored in the write buffer 282 to allow the read requests to be serviced in a pipelined manner followed by servicing the stored write requests in a pipelined manner. As a result, lengthy settling times between coupling write request to the memory devices 270a-d and subsequently coupling read request to the memory devices 270a-d for alternating write and read requests can be avoided.

The use of the cache memory unit 284 in each memory interface 270a-d allows the processor 104 to receive data responsive to a read command directed to a respective system memory device 240a-d without waiting for the memory device 240a-d to provide such data in the event that the data was recently read from or written to that memory device 240a-d. The cache memory unit 284 thus reduces the read latency of the system memory devices 240a-d to maximize the memory bandwidth of the computer system. Similarly, the processor 104 can store write data in the cache memory unit 284 and then perform other functions while the memory controller 280 in the same memory

interface 270a-d transfers the write data from the cache memory unit 284 to the system memory device 240a-d to which it is coupled.

Further included in the memory hub 200 is a built in self-test (BIST) and diagnostic engine 290 coupled to the switch 260 through a diagnostic bus 292. The diagnostic engine 290 is further coupled to a maintenance bus 296, such as a System Management Bus (SMBus) or a maintenance bus according to the Joint Test Action Group (JTAG) and IEEE 1149.1 standards. Both the SMBus and JTAG standards are well known by those ordinarily skilled in the art. Generally, the maintenance bus 296 provides a user access to the diagnostic engine 290 in order to perform memory channel and link diagnostics. For example, the user can couple a separate PC host via the maintenance bus 296 to conduct diagnostic testing or monitor memory system operation. By using the maintenance bus 296 to access diagnostic test results, issues related to the use of test probes, as previously discussed, can be avoided. It will be appreciated that the maintenance bus 296 can be modified from conventional bus standards without departing from the scope of the present invention. It will be further appreciated that the diagnostic engine 290 should accommodate the standards of the maintenance bus 296, where such a standard maintenance bus is employed. For example, the diagnostic engine should have an maintenance bus interface compliant with the JTAG bus standard where such a maintenance bus is used.

Further included in the memory hub 200 is a DMA engine 286 coupled to the switch 260 through a bus 288. The DMA engine 286 enables the memory hub 200 to move blocks of data from one location in the system memory to another location in the system memory without intervention from the processor 104. The bus 288 includes a plurality of conventional bus lines and signal lines, such as address, control, data busses, and the like, for handling data transfers in the system memory. Conventional DMA operations well known by those ordinarily skilled in the art can be implemented by the DMA engine 286. A more detailed description of a suitable DMA engine can be found in commonly assigned, co-pending U.S. Patent Application No. 10/625,132, entitled APPARATUS AND METHOD FOR DIRECT MEMORY ACCESS IN A HUB-BASED MEMORY SYSTEM, filed on July 22, 2003, which is

incorporated herein by reference. As described in more detail in the aforementioned patent application, the DMA engine 286 is able to read a link list in the system memory to execute the DMA memory operations without processor intervention, thus, freeing the processor 104 and the bandwidth limited system bus from executing the memory operations. The DMA engine 286 can also include circuitry to accommodate DMA operations on multiple channels, for example, for each of the system memory devices 240a-d. Such multiple channel DMA engines are well known in the art and can be implemented using conventional technologies.

The diagnostic engine 290 and the DMA engine 286 are preferably embedded circuits in the memory hub 200. However, including separate a diagnostic engine and a separate DMA device coupled to the memory hub 200 is also within the scope of the present invention.

Embodiments of the present invention provide performance monitoring components in communication with one or more of the memory controllers 280. The performance monitoring components allow the memory controllers 280 to dynamically adjust methods used to send and receive data from the memory units 240. In the example illustrated in Figure 2, at least one performance counter 300 is provided in communication with the memory controllers 280, as is described further below.

The performance counters 300 track one or more metrics associated with memory access and/or performance of memory hub 200, including for example, page hit rate, number or percentage of prefetch hits, and cache hit rate or percentage, in one example of the invention.

As described above, one approach to reducing latency in memory devices is to prefetch data. One example of the memory hub 140 of Figure 1 having prefetch buffers is shown in Figure 3 and described further in commonly assigned, co-pending U.S. Patent Application No. 10/601,252, entitled MEMORY HUB AND ACCESS METHOD HAVING INTERNAL PREFETCH BUFFERS, filed on June 20, 2003, which is incorporated herein by reference. As described in the aforementioned patent application, the memory hub 140 includes a link interface 152 that is coupled to the high-speed link 134. The link interface 152 may include a variety of conventional

interface circuitry such as, for example, a first-in, first-out buffer (not shown), for receiving and storing memory requests as they are received through the high-speed link 134. The memory requests can then be stored in the link interface until they can be processed by the memory hub 140.

A memory request received by the link interface 152 is processed by first transferring the request to a memory sequencer 160, which is included in one or more of memory controllers 270a-d in Figure 2, and is in communication with one or more performance counters 300. The memory sequencer 160 converts the memory requests from the format output from the system controller 110 (Figure 1) into a memory request having a format that can be used by the memory devices 148. These re-formatted request signals will normally include memory command signals, which are derived from memory commands contained in the memory request received by the memory hub 140, and row and column address signals, which are derived from an address contained in the memory request received by the memory hub 140. In the event the memory request is a write memory request, the re-formatted request signals will normally include write data signals which are derived from write data contained in the memory request received by the memory hub 140. For example, where the memory devices 148 are conventional DRAM devices, the memory sequencer 160 will output row address signals, a row address strobe ("RAS") signal, an active low write/active high read signal ("W*/R"), column address signals and a column address strobe ("CAS") signal. The re-formatted memory requests are preferably output from the sequencer 160 in the order they will be used by the memory devices 148.

The memory sequencer 160 applies the re-formatted memory requests to a memory device interface 166. The memory device interface 166, like the link interface 152, may include a FIFO buffer (not shown), for receiving and storing one or more memory requests as they are received from the link interface 152.

In the event the memory device interface 166 stores several memory requests until they can be processed by the memory devices 148, the memory device interface 166 may re-order the memory requests so that they are applied to the memory devices 148 in some other order. For example, the memory requests may be stored in

the interface 166 in a manner that causes one type of request, e.g., read requests, to be processed before other types of requests, e.g., write requests.

As previously explained, one of the disadvantages of using memory hubs is the increased latency they can sometimes create. As also previously explained, prefetch approaches that are traditionally used to reduce memory read latency are not well suited to a memory system using memory hubs. In contrast, the memory hub 140 shown in Figure 3 provides relatively low memory read latency by including a prefetch system 170 in the memory hub 140 that correctly anticipates which data will be needed during execution of a program, and then prefetches those data and stores them in one or more buffers that are part of the prefetch system 170. The prefetch system 170 includes several prefetch buffers 176, the number of which can be made variable depending upon operating conditions, as explained in greater detail below and in the aforementioned patent application. Briefly, the prefetch buffers 176 receive prefetched data from the memory device interface 166. The data are stored in the prefetch buffers 176 so that they will be available for a subsequent memory access. The data are then coupled through a multiplexer 178 to the link interface 152.

The prefetch system 170 also includes history logic 180 that receives the memory requests from the link interface 152. The history logic 180 analyzes the memory request using conventional algorithms to detect a pattern or stride from which future memory requests can be predicted. Although data may be prefetched from any address in the memory devices 148, the data are preferably prefetched only from rows in the memory devices 148 that are currently active or "open" so that the prefetching will not require a row of memory cells in the memory devices 148 to be precharged.In one example, one or more performance counter 300 tracks the number or percentage of page hits. The memory sequencer 160 adjusts the number of active or "open" pages based on information supplied by one or more performance counters 300, illustrated in Figure 2. In one example of the invention, the number of open pages is reduced by the memory sequencer 160 when the page hit count and/or page hit percentage tracked by at least one performance counter 300 falls below a threshold value. In an analogous manner, in one example, the number of open pages is increased when the page hit count or page hit

percentage exceeds a threshold value. Of course, other methods of adjusting the number of open pages are used in other examples of the invention.

The memory sequencer 160 may also selectively enable or disable prefetching depending on information supplied by one or more of the performace counters 300, such as page hit rate, percentage of prefetch hits, and the like. However, prefetching may also be enabled all of the time. In one example, the memory sequencer 300 disables prefetching when the number of prefetch hits and/or the page hit rate decreases below a threshold value. Alternatively, the sequencer 160 may enable or disable prefetching based on the percentage of memory requests that result in reading the requested data from the prefetch buffers 176 rather than from the memory devices 148.

When a memory module 130 containing a memory hub 140 receives a read memory request, it first determines whether or not the data or instruction called for by the request is stored in the prefetch buffers 176. This determination is made by coupling the memory request to tag logic 186. The tag logic 186 receives prefetch addresses from the history logic 180 corresponding to each prefetch suggestion. Alternatively, the tag logic 186 could receive prefetch addresses from the memory sequencer 160 corresponding to each prefetch request coupled to the memory device interface 166. Other means could also be used to allow the tag logic 186 to determine if data called for by a memory read request are stored in the prefetch buffer 176. In any case, the tag logic 186 stores the prefetch addresses to provide a record of the data that have been stored in the prefetch buffers 176. Using conventional techniques, the tag logic 186 compares the address in each memory request received from the link interface 152 with the prefetch addresses stored in the tag logic 186 to determine if the data called for by the memory request are stored in the prefetch buffers 176.

If the Tag Logic 186 determines that the data called for by a memory request are not stored in the prefetch buffers 176, it couples a low HIT/MISS* signal to the memory sequencer 160. If the Tag Logic 186 determines the data called for by a memory request are stored in the prefetch buffers 176, it couples a high HIT/MISS* signal to the memory sequencer 160. In one example, the incidences of high and/or lwo

HIT/MISS* signals are counted by one or more performance counters 300 to track the number of hits over the number of overall memory requests.

In one example, the performance counters 300, illustrated in Figure 3, track page hit rate over time. The page hit rate is then communicated to the memory sequencer 160 to adjust the number of open pages and/or to switch to an automatic precharge mode, where a requested line will automatically be precharged. In another example, the percentage of prefetch hits are tracked by the performance counters 300 to adjust whether prefetching is enabled and/or the number of prefetch requests to issue. In one example, at least one performance counter 300 tracks the number of cache hits, that is requests to caches 284a-d, where the requested data is located in the cache. If the cache hit rate is too low, the cache can be disabled, for example.

In one example., programmable thresholds are used to establish whether to use auto-precharge mode, number of open pages for page mode, number of prefetch requests and cacheability. In one example, the duration of monitoring by one or more performance counters 300 is programmable. The memory bus can be monitored for seconds, hours, or days, in various examples, to obtain the results or reset the counters.From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.

CLAIMS

1. A memory module, comprising:

a plurality of memory devices; and

a memory hub, comprising:

a link interface receiving memory requests for access to memory cells in at least one of the memory devices;

a memory device interface coupled to the memory devices, the memory device interface being operable to couple memory requests to the memory devices for access to memory cells in at least one of the memory devices and to receive read data responsive to at least some of the memory requests;

a performance counter coupled to the memory device interface, the performance counter operable to track at least one performance metric; and

a memory sequencer coupled to the link interface and the memory device interface, the memory sequencer being operable to couple memory requests to the memory device interface responsive to memory requests received from the link interface, the memory sequencer further being operable to dynamically adjust operability responsive to the performance metric tracked by the performance counter.

2. The memory module of claim 1 wherein the link interface comprises an optical input/output port.

3. The memory module of claim 1 wherein the performance metric tracked by the performance counter comprises at least one performance metric selected from the group consisting of page hit rate, prefetch hits, and cache hit rate.

4. The memory module of claim 3 wherein the performance counter tracks page hit rate and the memory sequencer is operable to change a number of open pages in the memory device.

5. The memory module of claim 3 wherein the performance counter tracks page hit rate and the memory sequencer is operable to switch to auto-precharge mode.

6. The memory module of claim 3 wherein the performance counter tracks a percentage of prefetch hits and the memory sequencer is operable to enable prefetching or disable prefetching.

7. The memory module of claim 3 wherein the performance counter tracks a percentage of prefetch hits and the memory sequencer is operable to determine a number of prefetch requests.

8. The memory module of claim 3 wherein the performance counter tracks a cache hit rate, and the memory sequencer is operable to disable the cache.

9. The memory module of claim 1 wherein the memory devices comprise dynamic random access memory devices.

10. A memory hub, comprising:

a link interface receiving memory requests for access to memory cells in at least one of the memory devices;

a memory device interface coupled to the memory devices, the memory device interface being operable to couple memory requests to the memory devices for access to memory cells in at least one of the memory devices and to receive read data responsive to at least some of the memory requests;

a performance counter coupled to the memory device interface, the performance counter operable to track at least one performance metric; and

a memory sequencer coupled to the link interface and the memory device interface, the memory sequencer being operable to couple memory requests to the memory device interface responsive to memory requests received from the link interface, the memory

sequencer further being operable to dynamically adjust operability responsive to the performance metric tracked by the performance counter.

11. The memory hub of claim 10 wherein the link interface comprises an optical input/output port.

12. The memory hub of claim 10 wherein the performance metric tracked by the performance counter comprises at least one performance metric selected from the group consisting of page hit rate, prefetch hits, and cache hit rate.

13. The memory hub of claim 12 wherein the performance counter tracks page hit rate and the memory sequencer is operable to change a number of open pages in the memory device.

14. The memory hub of claim 12 wherein the performance counter tracks page hit rate and the memory sequencer is operable to switch to auto-precharge mode.

15. The memory hub of claim 12 wherein the performance counter tracks a percentage of prefetch hits and the memory sequencer is operable to enable prefetching or disable prefetching.

16. The memory hub of claim 12 wherein the performance counter tracks a percentage of prefetch hits and the memory sequencer is operable to determine a number of prefetch requests.

17. The memory hub of claim 12 wherein the performance counter tracks a cache hit rate, and the memory sequencer is operable to disable the cache.

18. A computer system, comprising:

a central processing unit ("CPU");

a system controller coupled to the CPU, the system controller having an input port and an output port;

an input device coupled to the CPU through the system controller;

an output device coupled to the CPU through the system controller;

a storage device coupled to the CPU through the system controller;

a plurality of memory modules, each of the memory modules comprising:

a plurality of memory devices; and

a memory hub, comprising:

a link interface receiving memory requests for access to memory cells in at least one of the memory devices;

a memory device interface coupled to the memory devices, the memory device interface being operable to couple memory requests to the memory devices for access to memory cells in at least one of the memory devices and to receive read data responsive to at least some of the memory requests;

a performance counter coupled to the memory device interface, the performance counter operable to track at least one performance metric; and

a memory sequencer coupled to the link interface and the memory device interface, the memory sequencer being operable to couple memory requests to the memory device interface responsive to memory requests received from the link interface, the memory sequencer further being operable to dynamically adjust operability responsive to the performance metric tracked by the performance counter.

19. The computer system of claim 18 wherein the link interface comprises an optical input/output port.

20. The computer system of claim 18 wherein the performance metric tracked by the performance counter comprises at least one performance metric selected from the group consisting of page hit rate, prefetch hits, and cache hit rate.

21. The computer system of claim 20 wherein the performance counter tracks page hit rate and the memory sequencer is operable to change a number of open pages in the memory device.

22. The computer system of claim 20 wherein the performance counter tracks page hit rate and the memory sequencer is operable to switch to auto-precharge mode.

23. The computer system of claim 20 wherein the performance counter tracks a percentage of prefetch hits and the memory sequencer is operable to enable prefetching or disable prefetching.

24. The computer system of claim 20 wherein the performance counter tracks a percentage of prefetch hits and the memory sequencer is operable to determine a number of prefetch requests.

25. The computer system of claim 20 wherein the performance counter tracks a cache hit rate, and the memory sequencer is operable to disable the cache.

26. The computer system of claim 18 wherein the memory devices comprise dynamic random access memory devices.

27. A method of reading data from a memory module, comprising:

receiving memory requests for access to a memory device mounted on the memory module;

coupling the memory requests to the memory device responsive to the received memory request, at least some of the memory requests being memory requests to read data;

receiving read data responsive to the read memory requests;

tracking at least one performance metric; and

adjusting operability of a memory sequencer based on the tracked performance metric.

28. The method of claim 27 wherein the act of tracking at least one performance metric comprising tracking at least one performance metric selected from the group consisting of page hit rate, prefetch hits, and cache hit rate.

29. The method of claim 28 wherein the act of adjusting operability of a memory sequencer based on the tracked metric comprises adjusting operability of a memory sequencer if the tracked metric exceeds a threshold level.

30. The method of claim 29, further comprising programming the threshold level into a performance counter that performs the tracking.

31. The method of claim 28 wherein the act of adjusting operability of a memory sequencer based on the tracked metric comprises adjusting operability of a memory sequencer if the tracked metric is less than a threshold level

32. The method of claim 31, further comprising programming the threshold level into a performance counter that performs the tracking.

33. The method of claim 28 wherein the act of receiving memory requests for access to a memory device mounted on the memory module comprises receiving optical signals corresponding to the memory requests.

34. The method of claim 28 wherein the tracked performance metric comprises page hit rate and the act of adjusting operability of a memory sequencer based on the tracked performance metric comprises adjusting operability of the memory sequencer to change a number of open pages in the memory device.

35. The method of claim 28 wherein the tracked performance metric comprises page hit rate and the act of adjusting operability of a memory sequencer based on

the tracked performance metric comprises adjusting operability of the memory sequencer to switch to auto-precharge mode.

36. The method of claim 28 wherein the tracked performance metric comprises a percentage of prefetch hits and the act of adjusting operability of a memory sequencer based on the tracked performance metric comprises adjusting operability of the memory sequencer to enable prefetching or disable prefetching.

37. The method of claim 28 wherein the tracked performance metric comprises a percentage of prefetch hits and the act of adjusting operability of a memory sequencer based on the tracked performance metric comprises adjusting operability of the memory sequencer to determine a number of prefetch requests.

38. The method of claim 28 wherein the tracked performance metric comprises a cache hit rate, and the act of adjusting operability of a memory sequencer based on the tracked performance metric comprises adjusting operability of the memory sequencer to disable the cache.

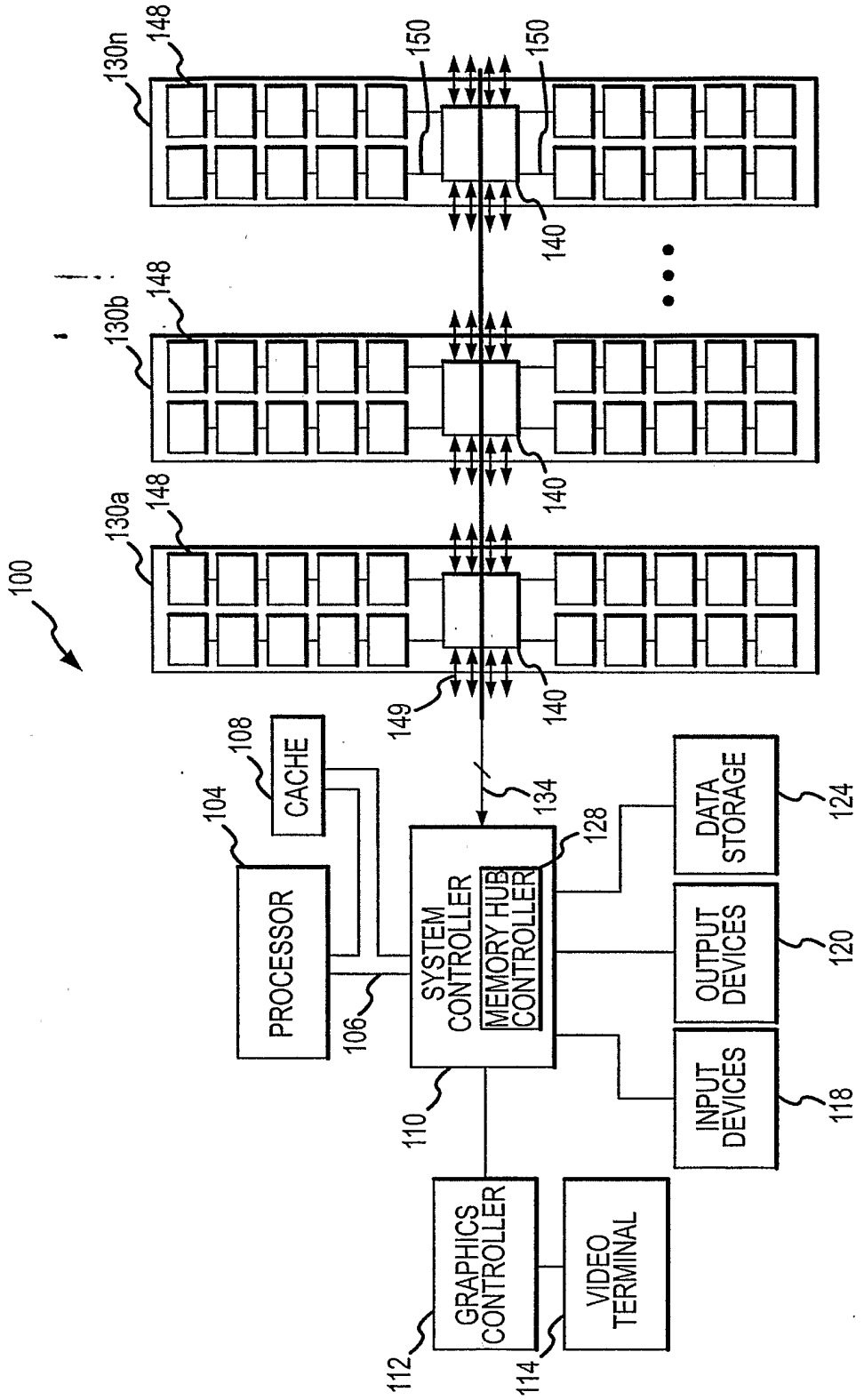

100
130n
130b
130a
148
150
140
149
134
128
108
CACHE
104
PROCESSOR
106
SYSTEM CONTROLLER
MEMORY HUB CONTROLLER
110
DATA STORAGE
124
OUTPUT DEVICES
120
INPUT DEVICES
118
GRAPHICS CONTROLLER
112
VIDEO TERMINAL
114

FIG.1

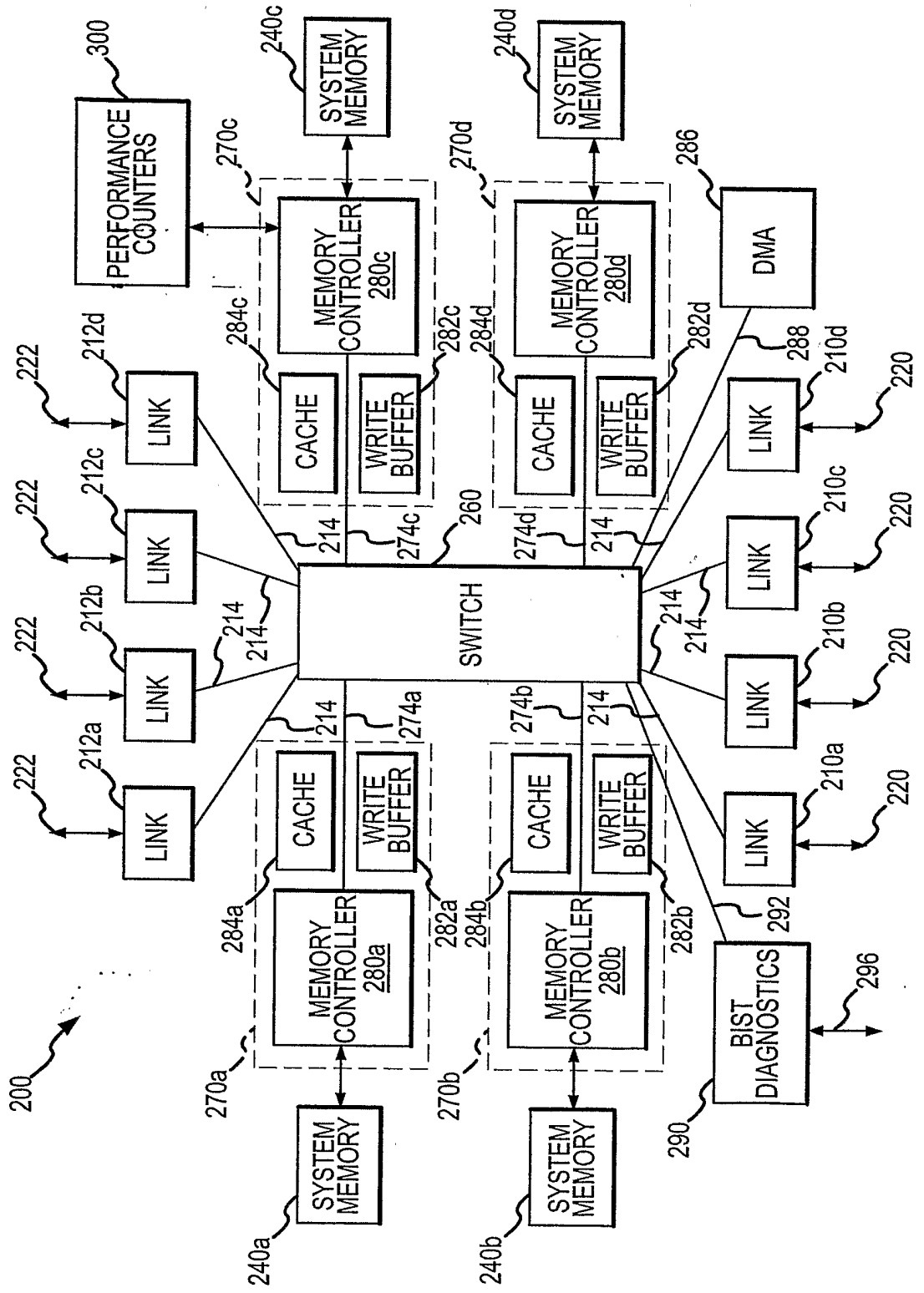

200
300
PERFORMANCE COUNTERS
240c
SYSTEM MEMORY
240d
SYSTEM MEMORY
270c
270d
MEMORY CONTROLLER 280c
MEMORY CONTROLLER 280d
284c
284d
CACHE
WRITE BUFFER
282c
282d
286
DMA
288
212d
212c
212b
212a
222
LINK
210d
210c
210b
210a
220
214
274c
274d
274a
274b
260
SWITCH
270a
270b
MEMORY CONTROLLER 280a
MEMORY CONTROLLER 280b
284a
284b
282a
282b
240a
240b
SYSTEM MEMORY
290
BIST DIAGNOSTICS
292
296

FIG.2

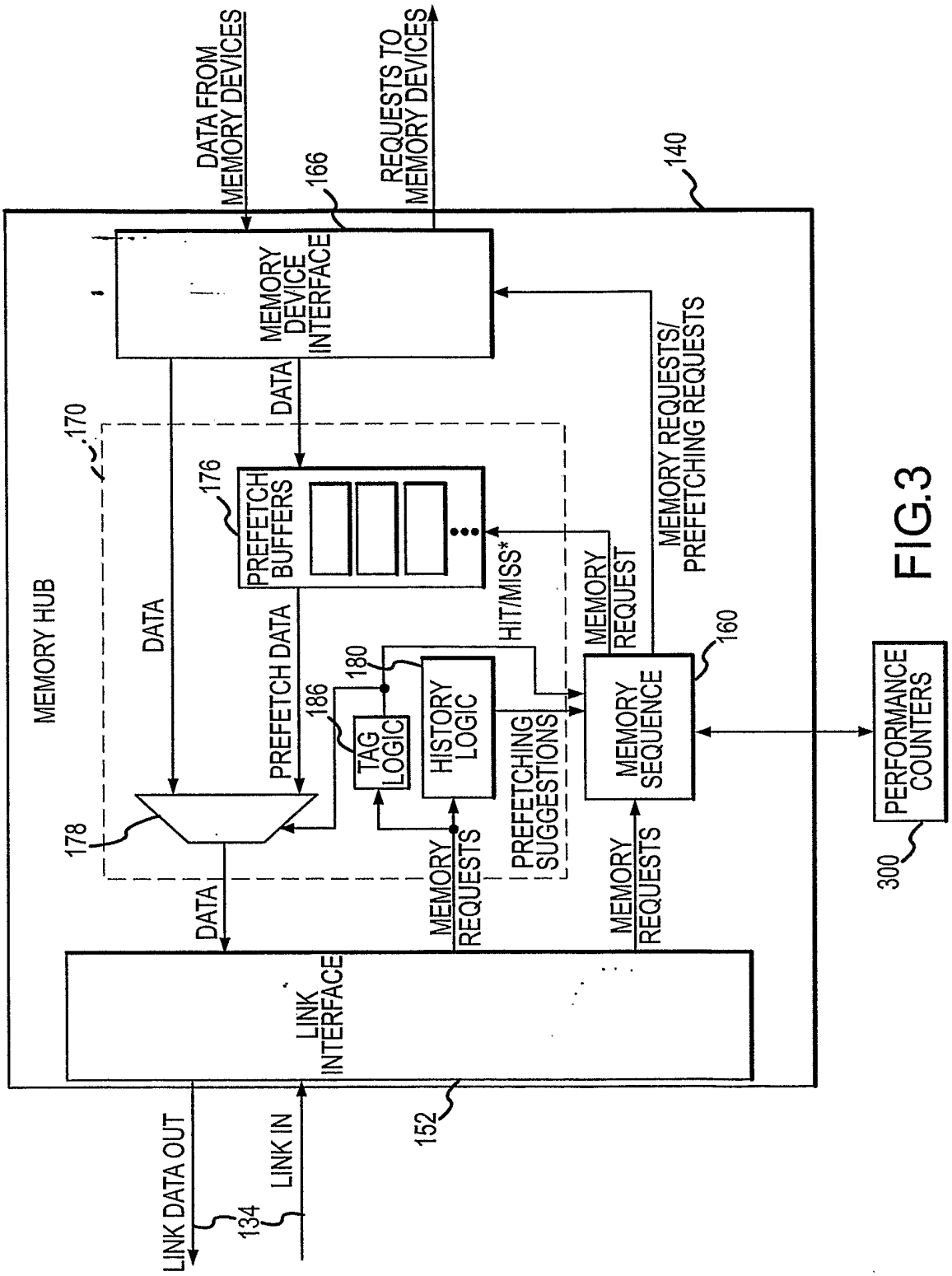

DATA FROM MEMORY DEVICES
REQUESTS TO MEMORY DEVICES
166
140
MEMORY DEVICE INTERFACE
DATA
MEMORY REQUESTS/ PREFETCHING REQUESTS
170
176
PREFETCH BUFFERS
MEMORY HUB
DATA
PREFETCH DATA
HIT/MISS*
MEMORY REQUEST
160
180
186
TAG LOGIC
HISTORY LOGIC
MEMORY SEQUENCE
PERFORMANCE COUNTERS
178
PREFETCHING SUGGESTIONS
MEMORY REQUESTS
MEMORY REQUESTS
300
DATA
LINK INTERFACE
152
LINK DATA OUT
LINK IN
134

FIG.3