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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 November 2007 (29.11.2007)

PCT

(10) International Publication Number
WO 2007/135144 A1

(51) International Patent Classification:
G06F 13/16 (2006.01)

(21) International Application Number:
PCT/EP2007/054929

(22) International Filing Date: 22 May 2007 (22.05.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/419,586 22 May 2006 (22.05.2006) US

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(71) Applicant *(for MG only)*: **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, North Harbour, Portsmouth Hampshire PO6 3AU (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **TREMAINE, Robert** [US/US]; 73 Stagecoach Pass, Stormville, NY 12582 (US).

(74) Agent: **WILLIAMS, Julian, David**; IBM United Kingdom Limited, Intellectual Property Law, Winchester Hampshire SO21 2JN (GB).

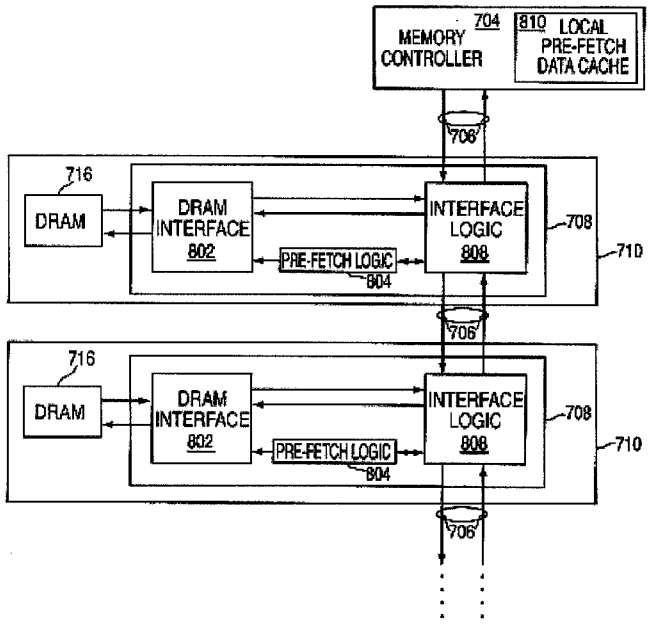
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP,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

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

(54) **Title:** SYSTEMS AND METHODS FOR PROVIDING REMOTE PRE-FETCH BUFFERS



(57) **Abstract:** Systems and methods for providing remote pre-fetch buffers. The systems include a computer memory system wit h a memory controller, one or more memory busses connected to the memory controller, and at least one memory subsystem in communication with the memory controller via the memory busses. The memory controller generates, receives and responds to memory access requests including unsolicited data transfers. The memory subsystem includes one or more memory devices and logic to initiate an unsolicited data transfer to the memory controller based on analysis performed at the memory subsystem of prior memory access requests received by the memory subsystem.

WO 2007/135144 A1

SYSTEMS AND METHODS FOR PROVIDING REMOTE PRE-FETCH BUFFERS

BACKGROUND OF THE INVENTION

This invention relates generally to computer memory, and more particularly to systems and methods for providing remote pre-fetch buffers.

Contemporary high performance computing main memory systems are generally composed of one or more dynamic random access memory (DRAM) devices, which are connected to one or more processors via one or more memory control elements. Overall computer system performance is affected by each of the key elements of the computer structure, including the performance/structure of the processor(s), any memory cache(s), the input/output (I/O) subsystem(s), the efficiency of the memory control function(s), the main memory device(s), and the type and structure of the memory interconnect interface(s).

Extensive research and development efforts are invested by the industry, on an ongoing basis, to create improved and/or innovative solutions to maximizing overall system performance and density by improving the memory system/subsystem design and/or structure. High-availability systems present further challenges as related to overall system reliability due to customer expectations that new computer systems will markedly surpass existing systems in regard to mean-time-between-failure (MTBF), in addition to offering additional functions, increased performance, increased storage, lower operating costs, etc. Other frequent customer requirements further exacerbate the memory system design challenges, and include such items as ease of upgrade and reduced system environmental impact (such as space, power and cooling).

FIG. 1 relates to U.S. Patent Number 5,513,135 to Dell et al., of common assignment herewith, and depicts an early synchronous memory module. The memory module depicted in FIG. 1 is a dual in-line memory module (DIMM). This module is composed of synchronous DRAMs 8, buffer devices 12, an optimized pinout, and an interconnect and capacitive decoupling method to facilitate high performance operation. The patent also

describes the use of clock re-drive on the module, using such devices as phase-locked loops (PLLs).

FIG. 2 relates to U.S. Patent Number 6,173,382 to Dell et al., of common assignment herewith, and depicts a computer system 10 which includes a synchronous memory module 20 that is directly (i.e. point-to-point) connected to a memory controller 14 via a bus 40, and which further includes logic circuitry 24 (such as an application specific integrated circuit, or "ASIC") that buffers, registers or otherwise acts on the address, data and control information that is received from the memory controller 14. The memory module 20 can be programmed to operate in a plurality of selectable or programmable modes by way of an independent bus, such as an inter-integrated circuit (I2C) control bus 34, either as part of the memory initialization process or during normal operation. When utilized in applications requiring more than a single memory module connected directly to a memory controller, the patent notes that the resulting stubs can be minimized through the use of field-effect transistor (FET) switches to electrically disconnect modules from the bus.

Relative to U.S. Patent Number 5,513,135, U.S. Patent Number 6,173,382 further demonstrates the capability of integrating all of the defined functions (address, command, data, presence detect, etc) into a single device. The integration of functions is a common industry practice that is enabled by technology improvements and, in this case, enables additional module density and/or functionality.

FIG. 3, from U.S. Patent Number 6,510,100 to Grundon et al., of common assignment herewith, depicts a simplified diagram and description of a memory system 10 that includes up to four registered DIMMs 40 on a traditional multi-drop stub bus. The subsystem includes a memory controller 20, an external clock buffer 30, registered DIMMs 40, an address bus 50, a control bus 60 and a data bus 70 with terminators 95 on the address bus 50 and the data bus 70. Although only a single memory channel is shown in FIG. 3, systems produced with these modules often included more than one discrete memory channel from the memory controller, with each of the memory channels operated singly (when a single channel was populated with modules) or in parallel (when two or more channels where populated with modules) to achieve the desired system functionality and/or performance.

FIG. 4, from U.S. Patent Number 6,587,912 to Bonella et al., depicts a synchronous memory module 210 and system structure in which the repeater hubs 320 include local re-drive of the address, command and data to the local memory devices 301 and 302 via buses 321 and 322; generation of a local clock (as described in other figures and the patent text); and the re-driving of the appropriate memory interface signals to the next module or component in the system via bus 300.

Contemporary computing systems may employ hub chip based memory systems, where memory devices (typically DRAM) are connected to memory hub devices. The memory hub devices are interconnected to the system memory controller(s) via a network of communication channels. To facilitate high frequency signaling, the channels are generally comprised of one or more unidirectional outputs (downstream) and a unidirectional input (upstream) point-to-point links. Processing and/or input/output (I/O) requests (including address, read/write indication and/or any other attributes) to access the system memory (referred to as access requests) are serviced by the system memory controller(s). The memory controller translates and regulates the access requests. The memory controller schedules and prioritizes the requests to the available memory banks for optimal system performance. The requests are specifically encoded according to the channel protocol (or associated interface protocol) and transmitted to the selected memory hub devices via the downstream link(s). Write requests include data associated with the write request, but not necessarily sent in the same packet. Read requests imply an expected data reply that will be transferred back to the memory controller via the one or more subsequent upstream packets. The targeted hub device(s) translate received requests and responsively control the attached memory devices to store write data from the hub, or provide read data to the hub.

The memory controller data read access latency can be mitigated by having a pre-fetch buffer associated with the memory controller. A pre-fetch buffer generally consists of a small associative cache, with one or more logical entries for storing data, associated address information, and other attributes. The memory controller can autonomously speculatively read and/or responsively read data from the memory devices for the purpose of storing the data into the pre-fetch buffer for lowest latency to anticipated future requests.

Hub based memory systems that employ pre-fetch buffers within a hub device to mitigate the latency associated with directly accessing the memory devices have been described (see, for example, U.S. Publication Number US2004/0260909 to Lee, et al.). However, this scheme incurs significant latency associated with the memory controller communicating with the hub device(s) to reference pre-fetch data stored within the hub. Having a pre-fetch data cache in the memory controller mitigates the access latency associated with implementing such buffers in the remote hub devices. However, both of these techniques require the use of critical output and input channel resources to transport data from the memory devices to the pre-fetch buffers, often during periods of high contention for these resources. Therefore, a need exists for a more efficient hub-based memory system to improve computing performance through lower memory latency and higher memory throughput.

BRIEF SUMMARY OF THE INVENTION

Embodiments include a computer memory system with a memory controller, one or more memory busses connected to the memory controller, and at least one memory subsystem in communication with the memory controller via the memory busses. The memory controller generates, receives and responds to memory access requests including unsolicited data transfers. The memory subsystem includes one or more memory devices and logic to initiate an unsolicited data transfer to the memory controller based on analysis performed at the memory subsystem of prior memory access requests received by the memory subsystem.

Embodiments also include a computer memory subsystem including pre-fetch logic and a hub device. The pre-fetch logic performs an analysis of prior memory access requests to one or more memory devices in communication with the memory subsystem. The pre-fetch logic also initiates an unsolicited data read to one or more of the memory devices based on results of the analysis. The hub device is in communication with a memory bus for initiating an unsolicited data transfer to a memory controller. The unsolicited data transfer includes data returned by the unsolicited data read.

Embodiments also include a memory controller for generating, receiving, and responding to memory access requests. The memory controller includes a local pre-fetch data cache and unsolicited data transfer logic for facilitating processing of unsolicited data transfers from a memory subsystem. The processing includes receiving a data packet and identifying the data packet as an unsolicited data packet. The unsolicited data packet includes data from an address range. The processing also includes verifying that the data at the address range has not been overwritten by a subsequent write command to the address range. The contents of the unsolicited data packet are stored in the local pre-fetch data cache in response to verifying that the data at the address range has not been overwritten by a subsequent write command to the address range.

Embodiments further include a method for receiving and responding to memory access requests including receiving a data packet at a memory controller. The data packet is identified as being an unsolicited data packet and includes an address range. It is verified that the data at the address range has not been overwritten by a subsequent write command to the address range. The contents of the unsolicited data packet are stored in a local pre-fetch data cache in response to verifying that the data at the address range has not been overwritten by a subsequent write command to the address range.

Embodiments further include a method of driving unsolicited data packets in a memory system including analyzing prior memory access requests to one or more memory devices, the analyzing performed at a memory module. A data read command is initiated at an address range on one or more of the memory devices based on results of the analyzing. An unsolicited data packet is transmitted to a memory controller when an upstream bus is idle. The unsolicited data packet includes data returned by the data read command.

Other systems, methods, and/or computer program products according to embodiments will be or become apparent to one with skill in the art upon review of the following drawings and detailed description. It is intended that all such additional systems, methods, and/or computer program products be included within this description, be within the scope of the present invention, and be protected by the accompanying claims.

Viewed from a first aspect, the present invention comprises a computer memory system comprising: a memory controller for generating, receiving and responding to memory access requests including unsolicited data transfers; one or more memory busses connected to the memory controller; and at least one memory subsystem in communication with the memory controller via the one or memory busses, the memory subsystem including one or more memory devices and logic to initiate an unsolicited data transfer to the one or more memory controller based on analysis performed at the memory subsystem of prior memory access requests received by the memory subsystem.

Preferably, the present invention provides a computer memory system wherein the logic in the memory subsystem autonomously performs the analysis and initiates the unsolicited data transfer.

Preferably, the present invention provides a computer memory system wherein the unsolicited data transfer includes address information associated with the data transfer.

Preferably, the present invention provides a computer memory system wherein the unsolicited data transfer includes a tag identifying it as an unsolicited data transfer.

Preferably, the present invention provides a computer memory system wherein the unsolicited data transfer is initiated at a lower priority than a solicited data transfer.

Preferably, the present invention provides a computer memory system wherein the unsolicited data transfer is formatted as a packet that includes a tag identifying the packet as containing an unsolicited data transfer.

Preferably, the present invention provides a computer memory system wherein the memory controller includes logic to identify the unsolicited data transfer.

Preferably, the present invention provides a computer memory system further comprising a data cache for storing pre-fetch data in anticipation of future data reference, thereby

eliminating an access to the memory subsystem when the requested data is available in the data cache.

Preferably, the present invention provides s computer memory system wherein the data cache is a logical partition of one or more caches in a hierarchy between the memory controller and one or more requesting devices and the data cache is coherently managed according to policies established for a computing environment.

Preferably, the present invention provides a computer memory system wherein the data cache includes instructions to service a memory access request from the data cache without initiating an access request to the memory controller.

Preferably, the present invention provides a computer memory system wherein the data cache is integrated within the memory controller and independently managed by the memory controller for replacement and coherency with respect to request traffic.

Preferably, the present invention provides a computer memory system wherein the memory controller includes instructions to service a memory access request using data from the data cache in lieu of accessing the memory subsystem.

Preferably, the present invention provides a computer memory system further comprising a computer memory subsystem, the computer memory subsystem comprising: pre-fetch logic to perform an analysis of prior memory access requests to one or more memory devices in communication with the memory subsystem and to initiate an unsolicited data read to one or more of the memory devices based on results of the analysis; and a hub device in communication with a memory bus for initiating an unsolicited data transfer to a memory controller, the unsolicited data transfer including data returned by the unsolicited data read.

Preferably, the present invention provides a computer memory system wherein the unsolicited data transfer includes address information associated with the unsolicited data read.

Preferably, the present invention provides a computer memory system wherein the unsolicited data transfer includes a tag identifying the data transfer as an unsolicited data transfer.

Preferably, the present invention provides a computer memory system wherein the unsolicited data transfer is initiated at a lower priority than a solicited data transfer.

Preferably, the present invention provides a computer memory system wherein the unsolicited data transfer is formatted as a packet that includes a tag identifying the packet as containing an unsolicited data transfer.

Preferably, the present invention provides a computer memory system further comprises a memory controller for generating, receiving and responding to memory access requests, the memory controller comprising: a local pre-fetch data cache; and unsolicited data transfer logic for facilitating processing of unsolicited data transfers from a memory subsystem, the processing including: receiving a data packet; identifying the data packet as an unsolicited data packet, the data packet including data from an address range; verifying that the data at the address range has not been overwritten by a subsequent write command; and storing contents of the unsolicited data packet in the local pre-fetch data cache in response to verifying that the data at the address range has not been overwritten by a subsequent write command at the address range.

Viewed from a second aspect the present invention provides a computer memory subsystem comprising: pre-fetch logic to perform an analysis of prior memory access requests to one or more memory devices in communication with the memory subsystem and to initiate an unsolicited data read to one or more of the memory devices based on results of the analysis; and a hub device in communication with a memory bus for initiating an unsolicited data transfer to a memory controller, the unsolicited data transfer including data returned by the unsolicited data read.

Preferably, the present invention provides a computer memory subsystem wherein the unsolicited data transfer includes address information associated with the unsolicited data read.

Preferably, the present invention provides a computer memory subsystem wherein the unsolicited data transfer includes a tag identifying the data transfer as an unsolicited data transfer.

Preferably, the present invention provides a computer memory subsystem wherein the unsolicited data transfer is initiated at a lower priority than a solicited data transfer.

Preferably, the present invention provides a computer memory subsystem wherein the unsolicited data transfer is formatted as a packet that includes a tag identifying the packet as containing an unsolicited data transfer.

Viewed from a third aspect, the present invention provides a memory controller for generating, receiving and responding to memory access requests, the memory controller comprising: a local pre-fetch data cache; and unsolicited data transfer logic for facilitating processing of unsolicited data transfers from a memory subsystem, the processing including: receiving a data packet; identifying the data packet as an unsolicited data packet, the data packet including data from an address range; verifying that the data at the address range has not been overwritten by a subsequent write command; and storing contents of the unsolicited data packet in the local pre-fetch data cache in response to verifying that the data at the address range has not been overwritten by a subsequent write command at the address range.

Viewed from a fourth aspect, the present invention comprises a method for receiving and responding to memory access requests, the method comprising: receiving a data packet at a memory controller; identifying the data packet as an unsolicited data packet, the data packet including data from an address range; verifying that the data at the address range has not been overwritten by a subsequent write command; and storing contents of the unsolicited data packet in a local pre-fetch data cache in response to verifying that the data at the address range has not been overwritten by a subsequent write command at the address range.

Preferably, the present invention provides a method wherein the data packet includes tag indicating a packet type and the identifying includes determining if the tag indicates an unsolicited data packet.

Preferably, the present invention provides a method wherein the identifying includes verifying that the memory access request does not correlate to an outstanding memory access request.

Preferably, the present invention provides a method further comprising monitoring an upstream data channel for incoming data packets.

Viewed from a fifth aspect, a method of driving unsolicited data packets in a memory system, the method comprising: analyzing prior memory access requests to one or more memory devices, the analyzing performed at a memory module; initiating a data read command at an address range corresponding to one or more of the memory devices, the initiating responsive to results of the analyzing; and transmitting an unsolicited data packet including data returned by the data read command to a memory controller when an upstream bus is idle.

Preferably, the present invention provides a method further comprising merging pending read request data with the unsolicited data packet if the pending read request specifies the address range.

Preferably, the present invention provides a method further comprising discarding the unsolicited data packet if a pending write request specifies the address range.

Preferably, the present invention provides a method further comprising discarding the unsolicited data packet if the unsolicited data packet has aged beyond a limit and the upstream bus is not idle.

Preferably, the present invention provides a method further comprising monitoring activity on the upstream bus.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are described below in detail, by way of example only, with reference to the accompanying drawings in which:

FIG. 1 depicts an exemplary early synchronous memory module as is known in the art;

FIG. 2 depicts an exemplary computer system with a fully buffered synchronous memory module that is directly connected to a memory controller as is known in the art;

FIG. 3 depicts an exemplary memory system, shown with a single, traditional multi-drop stub bus as is known in the art;

FIG. 4 depicts a fully buffered synchronous memory module and system structure, where the fully buffered synchronous memory module includes a repeater function as is known in the art;

FIG. 5 depicts a block diagram of multiple independent daisy-chained memory interface channels that operate in parallel to support data access requests in accordance with a preferred embodiment of the present invention;

FIG. 6 depicts a block diagram of multiple independent daisy-chained memory interface channels that operate in parallel to support data access requests in accordance with a preferred embodiment of the present invention;

FIG. 7 depicts a memory system having one or more memory channels and memory controllers interconnected by one or more cascade interconnect buses that may be implemented by exemplary embodiments in accordance with a preferred embodiment of the present invention;

FIG. 8 is a block diagram of a memory system that may be implemented by exemplary embodiments in accordance with a preferred embodiment of the present invention;

FIG. 9 is a process flow for operations performed at a hub device following the pre-fetching of unsolicited data that may be implemented by exemplary embodiments in accordance with a preferred embodiment of the present invention;

FIG. 10 is a process flow for operations performed at a memory controller following the receipt of an unsolicited read packet from a hub chip that may be implemented by exemplary embodiments in accordance with a preferred embodiment of the present invention; and

FIG. 11 depicts a packet structure that may be implemented by exemplary embodiments in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

Exemplary embodiments include a computer memory system that includes one or more memory hub devices, with each hub device being connected to one or more memory devices. A memory controller directs requests to access the memory devices via channel(s) interconnecting the memory controller and the hub devices. Each hub device can pre-fetch data and send it to the memory controller as an unsolicited "read" reply. In exemplary embodiments, the sending of the pre-fetch data occurs on the channel during idle periods. The memory controller maintains a local set of "pre-fetch" data buffers (data cache) within the controller for storing the received pre-fetch data. Memory access requests are first compared to the local pre-fetch buffer contents, and are serviced directly from the local pre-fetch buffer when the address is valid, precluding the need to utilize the memory devices, hub devices and/or channels to access the data.

Exemplary embodiments include a computing system with a processor(s) and an I/O unit(s) (e.g., requestors) interconnected to a memory system that contains a memory controller and memory devices. The computer memory system includes a physical memory array with a plurality of memory devices for storing data and instructions. These memory devices may be connected directly to the memory controller and/or indirectly coupled to the memory controller through hub devices. In exemplary embodiments, the hub-based computer memory system has memory devices attached to a communication hub device that is

connected to a memory control device (e.g., a memory controller). The hub device may be connected to the memory controller through a multi-drop or point-to-point bus structure (which may further include a cascade connection to one or more additional hubs). Memory access requests are transmitted by the memory controller through the bus structure to the selected hub(s). In response to receiving the memory access requests, the hub device translates the memory access requests to control the memory devices to store write data from the hub device or to provide read data to the hub device. Read data is encoded into one or more communication packet(s) and transmitted through the bus(ses) to the memory controller.

In exemplary embodiments, a pre-fetch data cache is contained within the memory controller, and/or may be logically partitioned from the cache hierarchy between the requestor(s) and the memory controller. The memory controller can autonomously (self-initiate accesses, without direction from the processor or another system element), speculatively (initiate an access based on a historical accesses and in response to a local algorithm) and/or responsively read data from the memory devices for the purpose of storing the data into the pre-fetch data cache for lowest latency to anticipated future requests. In any case, there is an implied data reply from the hub device with the data associated with the read request.

Exemplary embodiments include a new network packet and associated protocol for having an unsolicited data packet from the hub device received at the memory controller. An unsolicited data packet refers to a data packet that is not associated with a read request from the memory controller (i.e., is not a "solicited" data packet). An unsolicited data packet is used by the hub device to forward unsolicited pre-fetch data to the pre-fetch data cache in the memory controller.

FIG. 5 depicts a contemporary system composed of two processors 502 which share a common memory controller 504. The common memory controller is integrated into a "north bridge chip" 506. In the configuration depicted in FIG. 5, multiple independent multi-drop memory interface busses 508 are logically aggregated together to operate in unison to support a single independent access request at a higher bandwidth with data and error

detection/correction information distributed or "striped" across the parallel busses and associated devices. The memory controller 504 attaches to four narrow/high speed point-to-point memory busses 508, with each bus 508 connecting one of the several unique memory controller interface channels to a daisy-chained memory subsystem 510 (or memory module) which includes at least a hub device and one or more memory devices. Some systems further enable operations when a subset of the memory busses 508 are populated with memory subsystems 510. In this case, the one or more populated memory busses 508 may operate in unison to support a single access request.

FIG. 6 depicts a contemporary system composed of an integrated processor chip 602, which contains one or more processor elements and an integrated memory controller 504. The operation of the system depicted in FIG. 6 is fundamentally identical to that of FIG. 5, with the integrated processor chip 602 in FIG. 6 representing the two processors 502 and the north bridge chip 506 in FIG. 5.

In alternate exemplary embodiments, the memory controller(s) 504 may be integrated together with one or more processor chips and supporting logic, packaged in a discrete chip (commonly called a "northbridge" chip), included in a multi-chip carrier with the one or more processors and/or supporting logic, or packaged in various alternative forms that best match the application/environment. Any of these solutions may or may not employ one or more narrow/high speed links to connect to one or more hub chips and/or memory devices.

Exemplary embodiments apply to memory systems constructed of one or more memory subsystems, generally in the form of memory modules with hub logic chips, or buffer devices, which are connected to a processor complex by a cascade interconnect bus 706 as depicted in FIG. 7. In alternate exemplary embodiments, the memory modules 710 (also referred to as memory subsystems due to the inclusion of memory devices 716 and associated memory interface circuitry, in for example a hub logic chip 708) are connected to the processor complex 702 by one or more busses 706. The memory modules 710 depicted in FIG. 7 contain both a hub logic chip 708 that contains remote prefetch logic and a first interface to the memory controller 704 in the processor complex 702 (containing one or more processors) via a cascade-interconnect memory bus 706 (shown in the exemplary

embodiment as including an upstream bus and a downstream bus). The memory modules 710 also include one or more memory devices 716 that are connected to the hub logic chip 708 via a second interface. A third interface may also exist on the hub logic chip 708, especially in the case of a cascade interconnect bus 706, to re-drive information such as address, command and data information to one or more memory modules 710 located downstream from the first memory module 710, as well as to receive and re-drive data and/or error, status and other operational information intended for an upstream memory module 710 or the main memory controller 704.

The memory modules 710 may be implemented in a variety of technologies including a DIMM, a single in-line memory module (SIMM) and/or other memory module or card structures. In general, a DIMM refers to a small circuit board which is comprised primarily of random access memory (RAM) integrated circuits or die on one or both sides with signal and/or power pins on both sides of the board. This can be contrasted to a SIMM which is a small circuit board or substrate composed primarily of RAM integrated circuits or die on one or both sides and single row of pins along one long edge. The DIMM depicted in FIG. 1 includes 168 pins in the exemplary embodiment, whereas subsequent DIMMs have been constructed with pincounts ranging from 100 pins to over 300 pins.

In exemplary embodiments, the memory bus 706 is constructed using multi-drop connections to the memory modules 710 and/or using point-to-point connections. The downstream portion of the controller interface (or memory bus 706), referred to as the downstream bus, may include command, address, data and other operational, initialization or status information being sent to the memory modules 710. Each memory module 710 may simply forward the information to the subsequent module(s) 710 via bypass circuitry; receive, interpret and re-drive the information if it is determined to be targeting a downstream memory module 710; re-drive some or all of the information without first interpreting the information to determine the intended recipient; or perform a subset or combination of these options.

The upstream portion of the memory bus 706, referred to as the upstream bus, returns requested read data and/or error, status or other operational information, and this information

may be forwarded to the subsequent memory module(s) 710 via bypass circuitry; be received, interpreted and re-driven if it is determined to be targeting an upstream memory module 710 and/or memory controller 704 in the processor complex 702; be re-driven in part or in total without first interpreting the information to determine the intended recipient; or perform a subset or combination of these options.

Exemplary embodiments may be utilized by a single level memory system having only a single memory module 710, or memory subsystem, which is directly connected (such as via one or more point-to-point interconnections) to the main memory controller 704, with or without additional memory subsystem positions connected in a cascade interconnect or multi-drop structure. In additional exemplary embodiments, the main memory controller 704 includes more than one memory channel (see for e.g., bus 712 in FIG. 7), with each memory channel often comprised of one or more upstream and downstream buses 706 and 712 operating in a like manner; with the memory controller 704 operating in a parallel or independent manner for each memory bus 706 and 712; and/or with each memory bus 706 and 712 including the same or a different number of installed memory modules 710.

In exemplary embodiments, the memory system includes more than one memory module 710, with subsequent memory modules 710 communicating to the controller 704 by way of and/or independent of the first memory module 710. In exemplary embodiments, such as the one depicted in FIG. 7, the memory controller 704 is in communication with the first memory module 710 via a point-to-point bus, with the bus cascade-connected to a second (subsequent) memory module 710 via re-drive circuitry in the first memory module 710 (e.g., via circuitry in the hub logic chip 708). Third and fourth (and greater) memory modules 710 may be connected via a similar method, with the re-drive circuitry included in the memory module 710 located prior to that memory module 710 in the memory bus 706.

In alternate exemplary embodiments, the point-to-point bus includes a switch or bypass mechanism which results in the bus information being directed to one of two or more possible memory modules 710 during downstream communication (communication passing from the main memory controller 704 to a memory module 710), as well as directing upstream information (communication from a memory module 710 to the memory controller

704), often by way of one or more upstream memory modules 710. Further embodiments include the use of continuity modules, such as those recognized in the art, which, for example, can be placed between the memory controller and a first populated memory module, in a cascade interconnect memory system, such that any intermediate module positions between the memory controller and the first populated memory module include a means by which information passing between the memory controller and the first populated memory module position can be received even if the one or more intermediate module position(s) do not include a memory module. The continuity module(s) may be installed in any module position(s), subject to any bus restrictions, including the first position (closest to the main memory controller, the last position (prior to any included termination) or any intermediate position(s). The use of continuity modules may be especially beneficial in a multi-module cascade interconnect bus structure, where an intermediate memory module is removed and replaced by a continuity module, such that the system continues to operate after the removal of the intermediate memory module. In more common embodiments, the continuity module(s) would include either interconnect wires to transfer all required signals from the input(s) to the corresponding output(s), or be re-driven through a repeater device. The continuity module(s) might further include a non-volatile storage device (such as an EEPROM), but would not include main memory storage devices.

In exemplary embodiments, the memory system includes one or more memory modules 710 connected to the memory controller 704 via a cascade interconnect memory bus 706, however other memory structures may be implemented such as a point-to-point bus, a multi-drop memory bus or a shared bus. Depending on the signaling methods used, the target operating frequencies, space, power, cost, and other constraints, various alternate bus structures may be considered. A point-to-point bus may provide the optimal performance in systems produced with electrical interconnections, due to the reduced signal degradation that may occur as compared to bus structures having branched signal lines, switch devices, or stubs. However, when used in systems requiring communication with multiple devices or subsystems, this method will often result in significant added component cost and increased system power, and may reduce the potential memory density due to the need for intermediate buffering and/or re-drive.

Although not shown in FIG. 7, the memory modules 710 may also include a separate bus, such as a 'presence detect' bus, an I2C bus and/or an SMBus which is used for one or more purposes including the determination of the memory module attributes (generally after power-up), the reporting of fault or status information to the system, the configuration of the memory subsystem(s) 710 after power-up or during normal operation or other purposes. Depending on the bus characteristics, this bus might also provide a means by which the valid completion of operations could be reported by the memory module(s) 710 to the memory controller(s) 704, or the identification of failures occurring during the execution of the main memory controller requests.

Performances similar to those obtained from point-to-point bus structures can be obtained by adding switch devices. These and other solutions offer increased memory packaging density at lower power, while retaining many of the characteristics of a point-to-point bus. Multi-drop busses provide an alternate solution, albeit often limited to a lower operating frequency, but at a cost/performance point that may be advantageous for many applications. Optical bus solutions permit significantly increased frequency and bandwidth potential, either in point-to-point or multi-drop applications, but may incur cost and space impacts.

As used herein the term "buffer" or "buffer device" refers to a temporary storage unit (as in a computer), especially one that accepts information at one rate and delivers it another. In exemplary embodiments, a buffer is an electronic device that provides compatibility between two signals (e.g., changing voltage levels or current capability). The term "hub" is sometimes used interchangeably with the term "buffer." A hub is a device containing multiple ports that is connected to several other devices. A port is a portion of an interface that serves a congruent I/O functionality (e.g., a port may be utilized for sending and receiving data, address, and control information over one of the point-to-point links, or busses). A hub may be a central device that connects several systems, subsystems, or networks together. A passive hub may simply forward messages, while an active hub, or repeater, amplifies and refreshes the stream of data which otherwise would deteriorate over a distance. The term hub device 708, as used herein, refers to a hub chip that includes logic (hardware and/or software) for performing memory functions.

Also as used herein, the term "bus" refers to one of the sets of conductors (e.g., wires, and printed circuit board traces or connections in an integrated circuit) connecting two or more functional units in a computer. The data bus, address bus and control signals, despite their names, constitute a single bus since each are often useless without the others. A bus may include a plurality of signal lines, each signal line having two or more connection points, that form a main transmission path that electrically connects two or more transceivers, transmitters and/or receivers. The term "bus" is contrasted with the term "channel" which is often used to describe the function of a "port" as related to a memory controller in a memory system, and which may include one or more busses or sets of busses. The term "channel" as used herein refers to a port on a memory controller. Note that this term is often used in conjunction with I/O or other peripheral equipment, however the term channel has been adopted by some to describe the interface between a processor or memory controller and one of one or more memory subsystem(s).

Further, as used herein, the term "daisy chain" refers to a bus wiring structure in which, for example, device A is wired to device B, device B is wired to device C, etc. The last device is typically wired to a resistor or terminator. All devices may receive identical signals or, in contrast to a simple bus, each device may modify one or more signals before passing them on. A "cascade" or cascade interconnect' as used herein refers to a succession of stages or units or a collection of interconnected networking devices, typically hubs, in which the hubs operate as a logical repeater, further permitting merging data to be concentrated into the existing data stream. Also as used herein, the term "point-to-point" bus and/or link refers to one or a plurality of signal lines that may each include one or more terminators. In a point-to-point bus and/or link, each signal line has two transceiver connection points, with each transceiver connection point coupled to transmitter circuitry, receiver circuitry or transceiver circuitry. A signal line refers to one or more electrical conductors or optical carriers, generally configured as a single carrier or as two or more carriers, in a twisted, parallel, or concentric arrangement, used to transport at least one logical signal.

Memory devices are generally defined as integrated circuits that are composed primarily of memory (storage) cells, such as DRAMs (Dynamic Random Access Memories), SRAMs (Static Random Access Memories), FeRAMs (Ferro-Electric RAMs), MRAMs (Magnetic

Random Access Memories), Flash Memory and other forms of random access and related memories that store information in the form of electrical, optical, magnetic, biological or other means. Dynamic memory device types may include asynchronous memory devices such as FPM DRAMs (Fast Page Mode Dynamic Random Access Memories), EDO (Extended Data Out) DRAMs, BEDO (Burst EDO) DRAMs, SDR (Single Data Rate) Synchronous DRAMs, DDR (Double Data Rate) Synchronous DRAMs or any of the expected follow-on devices such as DDR2, DDR3, DDR4 and related technologies such as Graphics RAMs, Video RAMs, LP RAM (Low Power DRAMs) which are often based on the fundamental functions, features and/or interfaces found on related DRAMs.

Memory devices may be utilized in the form of chips (die) and/or single or multi-chip packages of various types and configurations. In multi-chip packages, the memory devices may be packaged with other device types such as other memory devices, logic chips, analog devices and programmable devices, and may also include passive devices such as resistors, capacitors and inductors. These packages may include an integrated heat sink or other cooling enhancements, which may be further attached to the immediate carrier or another nearby carrier or heat removal system.

Module support devices (such as buffers, hubs, hub logic chips, registers, PLL's, DLL's, non-volatile memory, etc) may be comprised of multiple separate chips and/or components, may be combined as multiple separate chips onto one or more substrates, may be combined onto a single package or even integrated onto a single device – based on technology, power, space, cost and other tradeoffs. In addition, one or more of the various passive devices such as resistors, capacitors may be integrated into the support chip packages, or into the substrate, board or raw card itself, based on technology, power, space, cost and other tradeoffs. These packages may include an integrated heat sink or other cooling enhancements, which may be further attached to the immediate carrier or another nearby carrier or heat removal system.

FIG. 8 is a block diagram of a memory system that may be implemented by exemplary embodiments. The system depicted in FIG. 8 is similar to the system depicted in FIG. 7, with FIG. 8 depicting elements in the hub device 708 and memory controller 704 that may

be utilized to implement exemplary embodiments. The memory controller 704 includes a local pre-fetch data cache 810, although this cache may also be logically partitioned from the existing cache hierarchy between the requestors (e.g. processor(s)) and the memory controller. In exemplary embodiments, the local pre-fetch data cache 810 is managed according to policies established by the computing environment. In further embodiments, the local pre-fetch data cache is managed by the memory controller 704 according to the buffer replacement policy and coherency dependencies with respect to outstanding read and write requests.

The hub device 708 includes interface logic 808, pre-fetch logic 804 and a DRAM interface 802. The pre-fetch logic 804 is utilized to monitor accesses to the memory devices from the memory controller 704. Based on access patterns by the memory controller 704, the pre-fetch logic 804 determines if a pre-fetch should be performed at the hub device 708. In an exemplary embodiment, pre-fetch requests would include the next sequential address or stride address, based on the historical system access pattern – although other pre-fetch algorithms could be applied based on the system design and/or programming. As used herein, the term stride address implies that the next address will be a distance from the previous address that is consistent with historical address spacings and not necessarily an adjacent address. Any suitable pre-fetch algorithm may be implemented by the pre-fetch logic 804 on the hub device 708. In exemplary embodiments, if the pre-fetch logic 804 determines that data should be pre-fetched, then it directs the DRAM interface 802 to read the data. In alternate exemplary embodiments, the pre-fetch logic 804 directs the interface logic 808 to read the data and the interface logic requests the data from the DRAM interface 802.

The interface logic 808 includes standard interface logic functions in addition to logic that is utilized to transmit the unsolicited pre-fetch data to the memory controller 704 as described below in reference to FIG. 9. In alternate exemplary embodiments, the pre-fetch logic includes the instructions that are utilized to transmit the unsolicited pre-fetch data to the memory controller 704 as described below in reference to FIG. 9. In alternate exemplary embodiments, the memory devices 716 and/or hub devices 708 include a pre-fetch data

cache for storing the pre-fetch data (also referred to herein as the data returned by the unsolicited data read) locally before transmitting it to the memory controller 704.

FIG. 9 is a process flow for operations performed at the hub device 708 following the pre-fetching of unsolicited data that is implemented by exemplary embodiments. The processing flow may be implemented by adding additional functions to the interface logic 808 in the hub device 708 or by adding separate pre-fetch logic 804 to the hub device 708. As used herein, the term "logic" refers to computer instructions that are implemented by hardware and/or software. At block 902, the hub device 708 (e.g., the pre-fetch logic 804) monitors accesses to the memory devices 716 by the memory controller 704. Based on an algorithm in the hub device 708 (e.g., the pre-fetch logic 804) and in response to prior accesses by the memory controller 704, data may be pre-fetched by the hub device 708 at block 902. Again, pre-fetch algorithms are well known in the art, and any suitable algorithm may be utilized by exemplary embodiments. Once the read operation to the memory device(s) 716 is completed, one or more packets are prepared for transmission to the memory controller 704. At block 904, it is determined if an access is complete and a packet is ready for transmission to the memory controller 704. In exemplary embodiments, a memory access will result in a cache line transfer (e.g. 64 bytes of 128 bytes) and the transmission of one or more packets to the memory controller is not initiated until data equal to the size of the cache line has been pre-fetched from the memory device(s) 716; however, other transfer and packet sizes may be utilized, depending on the application, and the transfer may be comprised of one or more packets. Block 902 is performed and the hub pre-fetches data until at least the minimum information associated with the first of one or more intended packet(s) is available. A variety of transfer or packet sizes can be used, based on the system design objectives, with FIG. 11 relating to an exemplary embodiment. The tag that accompanies the transfer identifies the type of transfer it is, and therefore the number of bits/bytes/packets associated with the transfer.

If a packet is ready for transmission to the memory controller 704, as determined at block 904, then block 906 is performed. At block 906, the activity on the upstream bus is monitored by the hub device 708 (e.g., by the interface logic 808). At block 908, it is determined if the upstream bus is available to enable transmission of the unsolicited data

packet. Generally, the upstream bus is available if there is a period of inactivity on the upstream bus and there are no other higher priority packets to be sent upstream from the hub device 708. If the upstream bus is available, then block 916 is performed and the hub device 708 transmits the data packet to the memory controller 704, in conjunction with a tag which identifies the data (e.g. the address range and module identification). In this manner, the unsolicited data is sent, typically at a low priority, to the memory controller 704. Processing then continues at block 902. In conventional cascade interconnect memory systems, interface logic 808 is responsible for the merging of local information onto the cascade interconnect bus, although the memory controller schedules accesses such that the bus will be idle at the time the information is prepared for transmission to the memory controller, for the duration of the transfer. In the exemplary embodiment, interface logic 808 will initiate an upstream transfer to the controller at a time based on the application settings and transfer priority, and may temporarily delay or otherwise interrupt transfers in flight from downstream modules.

If the upstream bus is not available, as determined at block 908, then block 910 is performed and the hub device 708 monitors accesses directed to that hub device 708 on the downstream bus to determine if any accesses are directed to the same address range as the unsolicited data packet(s) currently available and waiting to be sent to the memory controller 704 on the upstream bus. If it is determined, at block 912, that a read operation is received that matches the address range of the available unsolicited packet, then block 914 is performed. At block 914, the tag associated with the unsolicited packet is modified to indicate the packet is a response packet and the priority is raised commensurate with the new packet identification. Then, block 916 is performed to merge the packet onto the upstream data bus once any current transfers from that hub device 708 are completed, with the merge also being consistent with the read priority algorithm.

If it is determined, at block 918, that a write operation has been received that matches the address range of the available unsolicited packet, then block 922 is performed. At block 922, the packet is discarded because the pre-fetch data is no longer valid, and processing continues at block 902 where the hub device 708 continues to monitor accesses for optimal pre-fetches. Upon completion of an unsolicited pre-fetch from the memory device(s) 716,

the timeout counter is set to a pre-defined time limit for retaining the packet(s) (e.g., twenty clock cycles, thirty clock cycles) and begins to count down to zero. Once the packet(s) have aged to the pre-defined time limit (or have reached a pre-defined age), as determined at block 920, the timeout counter raises a flag, and any packet(s) associated with that timeout period that are not currently in the process of being sent to the memory controller 704 are discarded at block 922. The pre-defined time limit is a programmable (e.g., modifiable) number that may differ between hub devices 708. Processing then continues at block 902. If the timeout period has not been reached, then processing continues at block 908 to determine if the upstream bus is available 908 to transmit the packet. Although the steps in block 908, 910, 912, 918 and 920 are shown in an exemplary process flow, other process flows may be utilized, including a parallel flow (where read, write and timeout events are monitored in parallel) or a combination of series and parallel process flows (such as monitoring the timeout block 920 in parallel with completing read block 912 followed by write block 918) without straying from the fundamental teachings.

FIG. 10 is an exemplary process flow for operations performed at the memory controller 704 following the receipt of one or more unsolicited read packets from one or more hub devices 708 resident in the memory system. At block 1002, the memory controller monitors all bus activity from the hub devices 708 on the memory modules 710 by interpreting the packets to determine if they contain header information identifying the packet as a solicited, status, unsolicited or other form of communication (e.g. one or more informational packets) from the one or more memory modules 710, or memory subsystems, resident in the memory system. At block 1004, it is determined if a valid packet has been received. In exemplary embodiments, the memory controller 704 looks at the first transfer(s) (or some number of bits) of the bus information (e.g., a header field) to determine if the information appears to include one or more valid packet(s). If the memory controller 704 does not identify the information on the bus 706 as comprising one or more valid data packet(s) at block 1004, then block 1002 is performed and the memory controller 704 continues to monitor activity on the upstream bus 706.

If the memory controller 704 identifies a transmission on the bus 706 as being some form of data packet(s) at block 1004, then block 1006 is performed to determine if the packet(s) are

related to a reply to an access or other operation initiated by the controller. If the tag information attached to the packet(s) is decoded as a response to a solicited read request at block 1006, then block 1010 is performed and the memory controller 704 compares the remaining tag information (e.g. address range and/or module identifier) to the pending read requests to verify the validity of the data. If the tag correlates to an outstanding read request, as determined at block 1010, then block 1012 is performed and the data is transferred to the requestor (e.g., to one or more processors) at the next available time, based on the priority assigned to the read request, and the pending read request is retired. Processing then continues at block 1002. If the tag does not correlate to an outstanding read request, as determined at block 1010, then block 1014 is performed and the packet(s) are discarded and an error flag is generated for servicing by the processor or alternate error recovery system. Block 1014 is performed when the packet(s) appears to be a valid response packet, but does not correlate to an outstanding read request (e.g., the address or module identifier does not match outstanding read requests). In this case, the memory controller 704 discards the packet and then attempts to figure out why the error occurred by performing error recovery (e.g., channel diagnostics). Upon resolution of the error condition, in parallel with the diagnostics or immediately upon discarding the invalid packet(s) (e.g. due to a decision to delay diagnostics until a later time), processing then continues at block 1002.

If the information on the bus 706 is identified as one or more valid packet(s) in block 1004, but does not correlate to an outstanding read or write request to that address range as determined at block 1006, then block 1008 is performed and the packet(s) are evaluated to determine if the packet(s) is an unsolicited data packet(s). If the packet(s) is determined to be an unsolicited read packet at block 1008, then block 1022 is performed and the memory controller 704 double checks and once again compares the tag information to determine if the request correlates to a recent read request. If the packet(s) is found to correlate to a read request at block 1022, then block 1028 is performed and the solicited read request is transmitted to the requestor in a manner that is functionally similar to the processing in block 1012. Processing then continues at block 1002.

If it is determined, at block 1022, that the packet(s) does not correlate to a read request, then block 1024 is performed. At block 1024 it is determined if the packet(s) relates to a recent

write operation to that address range, and if the tag confirms that the data is older than the previous write to that address range, then block 1030 is performed and the packet(s) is discarded because the data is out of date. Updating the local cache with this data would result in out of date (or incorrect) data being stored in the local cache and thus a lack of data coherency. Processing then continues at block 1002. If no outstanding read or write request is identified for the address range, then block 1026 is performed and the packet(s) is added to the local pre-fetch data cache 810, in conjunction with the address information, consistent with a local pre-fetch data cache control algorithm. Processing then continues at block 1002.

If the packet(s) is determined not to be one or more unsolicited read packet(s) at block 1008, then the received packet(s) are neither solicited nor unsolicited read packet(s) and the packet(s) may consist of other valid communication(s) from one or more hub devices 708 to the memory controller 704. At block 1016, the packet(s) is evaluated to determine if the information in the data packet(s) is associated with status, failure recovery or other actions, in which case block 1020 is performed and the memory controller 704 takes action consistent with the design or programming of the system. Alternatively, block 1018 is performed if the packet(s) is determined to be a valid packet(s), but the contents are indeterminate or otherwise invalid. In this case, the memory controller 704 discards the packet and then attempts to figure out why the error occurred by performing error recovery (e.g., channel diagnostics). Upon resolution of the error condition, in parallel with the diagnostics or immediately upon discarding the invalid packet(s) (e.g. due to a decision to delay diagnostics until a later time), processing then continues at block 1002

Upon responding to the received transmission, in parallel with the execution of a response to the current packet(s) or immediately after initiating operation in response to the current packet, the memory controller 704 continues to monitor the bus 706 for incoming solicited or unsolicited packets at block 1002. The processing depicted in FIGs. 9 and 10 represent a logical process flow. The actual physical implementation of the processes may be different (e.g., some process blocks performed in parallel) as long as the physical implementation achieves the same results as those depicted in FIGs. 9 and 10.

FIG. 11 depicts an exemplary upstream data packet, which could be sent by one or more of the memory modules 710, via the cascade interconnect memory bus 706, to the memory controller 704. In exemplary embodiments, the unsolicited data transfer(s) from the hub device 708 to the memory controller 704 is formatted as a data packet. The exemplary data packet depicted in FIG. 11 includes nineteen bit lanes and one header and eight data transfers, yielding a total of one hundred and seventy one bit positions. Other exemplary packets may have more or less bit lanes and more or less transfers, and may include one or more spare bit lanes, based on the application requirements and/or conditions. The packet header is shown under the channel lane numbers, and defines the location of the tag (bits 0–5), command (CMD) bits (6–7), status (STS) bits (8–9), ECC bits (12-17), and other lanes which may be used for functions such as interrupts, spare bit lanes, CRC, data and other information that might apply to a given application. In exemplary embodiments, the tag is utilized, as described above in reference to FIG. 10, by the memory controller 704 to identify reply packets, to identify unsolicited pre-fetch data packets and to identify other types of packets received at the memory controller 704. The subsequent transfers (identified as transfers 0 – 7) refer to the data transfers that follow the header, and in the exemplary embodiment include 18 bit lanes, identified as 0 through 17, and 8 transfers, identified as 0 through 7, comprising a total of 144 bit positions (e.g. 16 bytes of data plus the associated EDC check bits). An additional bit lane is shown (shown as column 18 of FIG. 11), which may be used for such purposes as a spare bit lane.

The locations, count and function of each of the defined bit positions may also vary from what is described. In one case, an upstream data transfer packet might consist exclusively of header and status information, comprising one or several transfers. This packet may be used to report completion of an operation such as a write command or refresh, report an error condition, report a status condition (such as module or device temperature) or provide other information that requires a limited number of bit positions/transfers. In another case, the full nine transfers may be included in the packet, which may provide data to the memory controller 704 associated with a read request initiated by the memory controller 704, data from a self-initiated read operation (initiated by the memory subsystem, or memory module 710), in conjunction with information such as ECC bits and/or CRC bits, status bits, tag bits and other information.

In many cases, one or more packet transfers will be required to provide the requested/required information to the memory controller 704 and/or processor(s) in the processor complex 702, with four or eight packets commonly used. The one or more packets may also include information associated with more than one requested operation. For example, a series of packets associated with a requested read to a second memory module 710 might be sent to the memory controller 704 by way of the upstream bus and a first memory module 710 located upstream from the second memory module 710 and prior to the memory controller 704. Upon receipt of the first of the one or more packets, the first memory module 710 might concatenate separate and independent tag and/or status information to the one or more packets traveling upstream, generally in locations pre-defined for this purpose, such that the resulting packets include information associated with two or more independent operations from one or more separate memory modules 710. In this manner, the utilization of the available bus bandwidth can be maximized by ensuring that all of the possible bit positions in the packet(s) are utilized. Examples of two independent operations might include the results of a read operation from a second memory module 710, to which status information is appended from a first memory module 710, such as a write operation completion status, a refresh completion status, a mode change status, environmental information such as temperature, or other information.

Although not shown in FIG. 11, exemplary downstream packets are structured in a similar manner, with multiple bit lanes and one or more transfers involved in the communication of command, address, data, status, initialization, priority flag(s) and other information from the memory controller 704 to the one or more memory modules 710 on the memory bus(es) 706. The downstream packet will often be included as a portion of a multi-packet transfer, with the bit lanes in each packet having the same or different functions based on the specific transfer and/or the packet number (such as the first packet of four packets or the eighth packet of eight total packets). This is also true for the upstream packet.

Memory devices, hubs, buffers, registers, clock devices, passives and other memory support devices and/or components may be attached to the memory subsystem, or memory module 710, via various methods including solder interconnects, conductive adhesives, socket

structures, pressure contacts and other methods which enable communication between the two or more devices via electrical, optical or alternate means.

The one or more memory subsystem(s), or memory modules 710, may be connected to the memory system, processor complex, computer system or other system environment via one or more methods such as soldered interconnects, connectors, pressure contacts, conductive adhesives, optical interconnects and other communication and power delivery methods. Connector systems may include mating connectors (male/female), conductive contacts and/or pins on one carrier mating with a male or female connector, optical connections, pressure contacts (often in conjunction with a retaining mechanism) and/or one or more of various other communication and power delivery methods. The interconnection(s) may be disposed along one or more edges of the memory assembly and/or placed a distance from an edge of the memory subsystem depending on such application requirements as ease-of-upgrade/repair, available space/volume, heat transfer, component size and shape and other related physical, electrical, optical, visual/physical access, etc.

As used herein, the term memory subsystem refers to, but is not limited to: one or more memory devices; one or more memory devices and associated interface and/or timing/control circuitry; and/or one or more memory devices in conjunction with a memory buffer, hub, and/or switch. The term memory subsystem may also refer to one or more memory devices, in addition to any associated interface and/or timing/control circuitry and/or a memory buffer, hub device or switch, assembled into a substrate, a card, a module or related assembly, which may also include a connector or similar means of electrically attaching the memory subsystem with other circuitry. The memory modules 710 described herein may also be referred to as memory subsystems because they include one or more memory devices 716 and a hub device 708.

Additional functions that may reside local to the memory subsystem, or memory module 710, include write and/or read buffers, one or more levels of memory cache, local pre-fetch logic, data encryption/decryption, compression/decompression, protocol translation, command prioritization logic, voltage and/or level translation, error detection and/or correction circuitry, data scrubbing, local power management circuitry and/or reporting,

operational and/or status registers, initialization circuitry, performance monitoring and/or control, one or more co-processors, search engine(s) and other functions that may have previously resided in other memory subsystems. By placing a function local to the memory subsystem, added performance may be obtained as related to the specific function, often while making use of unused circuits within the subsystem.

Memory subsystem support device(s) may be directly attached to the same substrate or assembly onto which the memory device(s) 716 are attached, or may be mounted to a separate interposer or substrate also produced using one or more of various plastic, silicon, ceramic or other materials which include electrical, optical or other communication paths to functionally interconnect the support device(s) to the memory device(s) 716 and/or to other elements of the memory or computer system.

Information transfers (e.g. packets) along a bus, channel, link or other naming convention applied to an interconnection method may be completed using one or more of many signaling options. These signaling options may include such methods as single-ended, differential, optical or other approaches, with electrical signaling further including such methods as voltage or current signaling using either single or multi-level approaches. Signals may also be modulated using such methods as time or frequency, non-return to zero, phase shift keying, amplitude modulation and others. Voltage levels are expected to continue to decrease, with 1.5V, 1.2V, 1V and lower signal voltages expected consistent with (but often independent of) the reduced power supply voltages required for the operation of the associated integrated circuits themselves.

One or more clocking methods may be utilized within the memory subsystem and the memory system itself, including global clocking, source-synchronous clocking, encoded clocking or combinations of these and other methods. The clock signaling may be identical to that of the signal lines themselves, or may utilize one of the listed or alternate methods that is more conducive to the planned clock frequency(ies), and the number of clocks planned within the various subsystems. A single clock may be associated with all communication to and from the memory, as well as all clocked functions within the memory subsystem, or multiple clocks may be sourced using one or more methods such as those

described earlier. When multiple clocks are used, the functions within the memory subsystem may be associated with a clock that is uniquely sourced to the subsystem, or may be based on a clock that is derived from the clock related to the information being transferred to and from the memory subsystem (such as that associated with an encoded clock). Alternately, a unique clock may be used for the information transferred to the memory subsystem, and a separate clock for information sourced from one (or more) of the memory subsystems. The clocks themselves may operate at the same or frequency multiple of the communication or functional frequency, and may be edge-aligned, center-aligned or placed in an alternate timing position relative to the data, command or address information.

Information passing to the memory subsystem(s) will generally be composed of address, command and data, as well as other signals generally associated with requesting or reporting status or error conditions, resetting the memory, completing memory or logic initialization and other functional, configuration or related information. Information passing from the memory subsystem(s) may include any or all of the information passing to the memory subsystem(s), however generally will not include address and command information. This information may be communicated using communication methods that may be consistent with normal memory device interface specifications (generally parallel in nature), the information may be encoded into a 'packet' structure, which may be consistent with future memory interfaces or simply developed to increase communication bandwidth and/or enable the subsystem to operate independently of the memory technology by converting the received information into the format required by the receiving device(s).

Initialization of the memory subsystem may be completed via one or more methods, based on the available interface busses, the desired initialization speed, available space, cost/complexity objectives, subsystem interconnect structures, the use of alternate processors (such as a service processor) which may be used for this and other purposes, etc. In one embodiment, the high speed bus may be used to complete the initialization of the memory subsystem(s), generally by first completing a training process to establish reliable communication, then by interrogation of the attribute or 'presence detect' data associated the various components and/or characteristics associated with that subsystem, and ultimately by programming the appropriate devices with information associated with the intended

operation within that system. In a cascaded system, communication with the first memory subsystem would generally be established, followed by subsequent (downstream) subsystems in the sequence consistent with their position along the cascade interconnect bus.

A second initialization method would include one in which the high speed bus is operated at one frequency during the initialization process, then at a second (and generally higher) frequency during the normal operation. In this embodiment, it may be possible to initiate communication with all of the memory subsystems on the cascade interconnect bus prior to completing the interrogation and/or programming of each subsystem, due to the increased timing margins associated with the lower frequency operation.

A third initialization method might include operation of the cascade interconnect bus at the normal operational frequency(ies), while increasing the number of cycles associated with each address, command and/or data transfer. In one embodiment, a packet containing all or a portion of the address, command and/or data information might be transferred in one clock cycle during normal operation, but the same amount and/or type of information might be transferred over two, three or more cycles during initialization. This initialization process would therefore be using a form of 'slow' commands, rather than 'normal' commands, and this mode might be automatically entered at some point after power-up and/or re-start by each of the subsystems and the memory controller by way of POR (power-on-reset) logic included in each of these subsystems.

A fourth initialization method might utilize a distinct bus, such as a presence detect bus (such as the one defined in U.S. Patent Number 5,513,135 to Dell et al., of common assignment herewith), an I2C bus (such as defined in published JEDEC standards such as the 168 Pin DIMM family in publication 21-C revision 7R8) and/or the SMBUS, which has been widely utilized and documented in computer systems using such memory modules. This bus might be connected to one or more modules within a memory system in a daisy chain/cascade interconnect, multi-drop or alternate structure, providing an independent means of interrogating memory subsystems, programming each of the one or more memory subsystems to operate within the overall system environment, and adjusting the operational

characteristics at other times during the normal system operation based on performance, thermal, configuration or other changes desired or detected in the system environment.

Other methods for initialization can also be used, in conjunction with or independent of those listed. The use of a separate bus, such as described in the fourth embodiment above, also offers the advantage of providing an independent means for both initialization and uses other than initialization, such as described in U.S. Patent Number 6,381,685 to Dell et al., of common assignment herewith, including changes to the subsystem operational characteristics on-the-fly and for the reporting of and response to operational subsystem information such as utilization, temperature data, failure information or other purposes.

With improvements in lithography, better process controls, the use of materials with lower resistance, increased field sizes and other semiconductor processing improvements, increased device circuit density (often in conjunction with increased die sizes) will help facilitate increased function on integrated devices as well as the integration of functions previously implemented on separate devices. This integration will serve to improve overall performance of the intended function, as well as promote increased storage density, reduced power, reduced space requirements, lower cost and other manufacturer and customer benefits. This integration is a natural evolutionary process, and may result in the need for structural changes to the fundamental building blocks associated with systems.

The integrity of the communication path, the data storage contents and all functional operations associated with each element of a memory system or subsystem can be assured, to a high degree, with the use of one or more fault detection and/or correction methods. Any or all of the various elements may include error detection and/or correction methods such as CRC (Cyclic Redundancy Code), EDC (Error Detection and Correction), parity or other encoding/decoding methods suited for this purpose. Further reliability enhancements may include operation re-try (to overcome intermittent faults such as those associated with the transfer of information), the use of one or more alternate or replacement communication paths to replace failing paths and/or lines, complement-re-complement techniques or alternate methods used in computer, communication and related systems.

The use of bus termination, on busses as simple as point-to-point links or as complex as multi-drop structures, is becoming more common consistent with increased performance demands. A wide variety of termination methods can be identified and/or considered, and include the use of such devices as resistors, capacitors, inductors or any combination thereof, with these devices connected between the signal line and a power supply voltage or ground, a termination voltage or another signal. The termination device(s) may be part of a passive or active termination structure, and may reside in one or more positions along one or more of the signal lines, and/or as part of the transmitter and/or receiving device(s). The terminator may be selected to match the impedance of the transmission line, or selected via an alternate approach to maximize the useable frequency, operating margins and related attributes within the cost, space, power and other constraints.

Technical effects and benefits of exemplary embodiments include the ability to implement a more efficient hub based memory system by to improving computing performance through lower memory latency and higher memory throughput. Exemplary embodiments include pre-fetch logic executing on the hub devices and the transmission of the pre-fetch data to the memory controller as unsolicited data packets with a tag identifying the packets as such. The memory controller then stores the unsolicited read data packet in its local cache, which results in the cache data being close to the processor for faster access (when compared with having to request it from the hub device). In exemplary embodiments, the data is transmitted to the memory controller as a low priority data packet when the bus is idle, thereby allowing for better utilization of the bus and for preventing the memory controller from having to utilize the downstream bus to request the pre-fetched data.

As described above, the embodiments of the invention may be embodied in the form of computer-implemented processes and apparatuses for practicing those processes. Embodiments of the invention may also be embodied in the form of computer program code containing instructions embodied in tangible media, such as floppy diskettes, CD-ROMs, hard drives, or any other computer-readable storage medium, wherein, when the computer program code is loaded into and executed by a computer, the computer becomes an apparatus for practicing the invention. The present invention can also be embodied in the form of computer program code, for example, whether stored in a storage medium, loaded

into and/or executed by a computer, or transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via electromagnetic radiation, wherein, when the computer program code is loaded into and executed by a computer, the computer becomes an apparatus for practicing the invention. When implemented on a general-purpose microprocessor, the computer program code segments configure the microprocessor to create specific logic circuits.

While the invention has been described with reference to exemplary embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted for elements thereof without departing from the scope of the invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the invention without departing from the essential scope thereof. Therefore, it is intended that the invention not be limited to the particular embodiment disclosed as the best mode contemplated for carrying out this invention, but that the invention will include all embodiments falling within the scope of the appended claims. Moreover, the use of the terms first, second, etc. do not denote any order or importance, but rather the terms first, second, etc. are used to distinguish one element from another.

CLAIMS

1. A computer memory system comprising:

a memory controller for generating, receiving and responding to memory access requests including unsolicited data transfers;

one or more memory busses connected to the memory controller; and

at least one memory subsystem in communication with the memory controller via the one or memory busses, the memory subsystem including one or more memory devices and logic to initiate an unsolicited data transfer to the one or more memory controller based on analysis performed at the memory subsystem of prior memory access requests received by the memory subsystem.

2. A computer memory system as claimed in claim 1 wherein the logic in the memory subsystem autonomously performs the analysis and initiates the unsolicited data transfer.

3. A computer memory system as claimed in claim 1 wherein the unsolicited data transfer includes address information associated with the data transfer.

4. A computer memory system as claimed in claim 1 wherein the unsolicited data transfer includes a tag identifying it as an unsolicited data transfer.

5. A computer memory system as claimed in claim 1 wherein the unsolicited data transfer is initiated at a lower priority than a solicited data transfer.

6. A computer memory system as claimed in claim 1 wherein the unsolicited data transfer is formatted as a packet that includes a tag identifying the packet as containing an unsolicited data transfer.

7. A computer memory system as claimed in claim 1 wherein the memory controller includes logic to identify the unsolicited data transfer.

8. A computer memory system as claimed in claim 1 further comprising a data cache for storing pre-fetch data in anticipation of future data reference, thereby eliminating an access to the memory subsystem when the requested data is available in the data cache.

9. A computer memory system as claimed in claim 8, wherein the data cache is a logical partition of one or more caches in a hierarchy between the memory controller and one or more requesting devices and the data cache is coherently managed according to policies established for a computing environment.

10. A computer memory system as claimed in claim 9, wherein the data cache includes instructions to service a memory access request from the data cache without initiating an access request to the memory controller.

11. A computer memory system as claimed in claim 8, wherein the data cache is integrated within the memory controller and independently managed by the memory controller for replacement and coherency with respect to request traffic.

12. A computer memory system as claimed in claim 11, wherein the memory controller includes instructions to service a memory access request using data from the data cache in lieu of accessing the memory subsystem.

13. A computer memory system as claimed in claim 1 further comprises a computer memory subsystem, the computer memory subsystem comprising:

pre-fetch logic to perform an analysis of prior memory access requests to one or more memory devices in communication with the memory subsystem and to initiate an unsolicited data read to one or more of the memory devices based on results of the analysis; and

a hub device in communication with a memory bus for initiating an unsolicited data transfer to a memory controller, the unsolicited data transfer including data returned by the unsolicited data read.

14. A computer memory system as claimed in claim 13 wherein the unsolicited data transfer includes address information associated with the unsolicited data read.

15. A computer memory system as claimed in claim 13 wherein the unsolicited data transfer includes a tag identifying the data transfer as an unsolicited data transfer.

16. A computer memory system as claimed in claim 13 wherein the unsolicited data transfer is initiated at a lower priority than a solicited data transfer.

17. A computer memory system as claimed in claim 13 wherein the unsolicited data transfer is formatted as a packet that includes a tag identifying the packet as containing an unsolicited data transfer.

18. A computer memory system as claimed in claim 1 further comprises a memory controller for generating, receiving and responding to memory access requests, the memory controller comprising:

a local pre-fetch data cache; and

unsolicited data transfer logic for facilitating processing of unsolicited data transfers from a memory subsystem, the processing including:

receiving a data packet;

identifying the data packet as an unsolicited data packet, the data packet including data from an address range;

verifying that the data at the address range has not been overwritten by a subsequent write command; and

storing contents of the unsolicited data packet in the local pre-fetch data cache in response to verifying that the data at the address range has not been overwritten by a subsequent write command at the address range.

19. A computer memory subsystem comprising:

pre-fetch logic to perform an analysis of prior memory access requests to one or more memory devices in communication with the memory subsystem and to initiate an

unsolicited data read to one or more of the memory devices based on results of the analysis; and

a hub device in communication with a memory bus for initiating an unsolicited data transfer to a memory controller, the unsolicited data transfer including data returned by the unsolicited data read.

20. A computer memory subsystem as claimed in claim 19 wherein the unsolicited data transfer includes address information associated with the unsolicited data read.

21. A computer memory subsystem as claimed in claim 19 wherein the unsolicited data transfer includes a tag identifying the data transfer as an unsolicited data transfer.

22. A computer memory subsystem as claimed in claim 19 wherein the unsolicited data transfer is initiated at a lower priority than a solicited data transfer.

23. A computer memory subsystem as claimed in claim 19 wherein the unsolicited data transfer is formatted as a packet that includes a tag identifying the packet as containing an unsolicited data transfer.

24. A memory controller for generating, receiving and responding to memory access requests, the memory controller comprising:

a local pre-fetch data cache; and

unsolicited data transfer logic for facilitating processing of unsolicited data transfers from a memory subsystem, the processing including:

receiving a data packet;

identifying the data packet as an unsolicited data packet, the data packet including data from an address range;

verifying that the data at the address range has not been overwritten by a subsequent write command; and

storing contents of the unsolicited data packet in the local pre-fetch data cache in response to verifying that the data at the address range has not been overwritten by a subsequent write command at the address range.

25. A method for receiving and responding to memory access requests, the method comprising:

receiving a data packet at a memory controller;

identifying the data packet as an unsolicited data packet, the data packet including data from an address range;

verifying that the data at the address range has not been overwritten by a subsequent write command; and

storing contents of the unsolicited data packet in a local pre-fetch data cache in response to verifying that the data at the address range has not been overwritten by a subsequent write command at the address range.

26. A method as claimed in claim 25 wherein the data packet includes tag indicating a packet type and the identifying includes determining if the tag indicates an unsolicited data packet.

27. A method as claimed in claim 25 wherein the identifying includes verifying that the memory access request does not correlate to an outstanding memory access request.

28. A method as claimed in claim 25 further comprising monitoring an upstream data channel for incoming data packets.

29. A method of driving unsolicited data packets in a memory system, the method comprising:

analyzing prior memory access requests to one or more memory devices, the analyzing performed at a memory module;

initiating a data read command at an address range corresponding to one or more of the memory devices, the initiating responsive to results of the analyzing; and

transmitting an unsolicited data packet including data returned by the data read command to a memory controller when an upstream bus is idle.

30. A method as claimed in claim 29 further comprising merging pending read request data with the unsolicited data packet if the pending read request specifies the address range.

31. A method as claimed in claim 29 further comprising discarding the unsolicited data packet if a pending write request specifies the address range.

32. A method as claimed in claim 29 further comprising discarding the unsolicited data packet if the unsolicited data packet has aged beyond a limit and the upstream bus is not idle.

33. A method as claimed in claim 29 further comprising monitoring activity on the upstream bus.

34. A computer program product loadable into the internal memory of a digital computer, comprising software code portions for performing, when said product is run on a computer, to carry out the invention as claimed in claims 25 to 33.

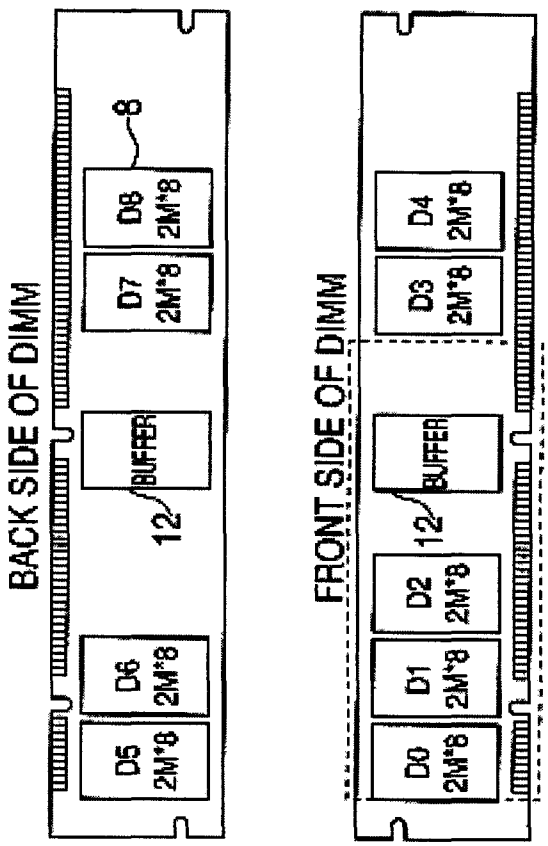


FIG. 1
(PRIOR ART)

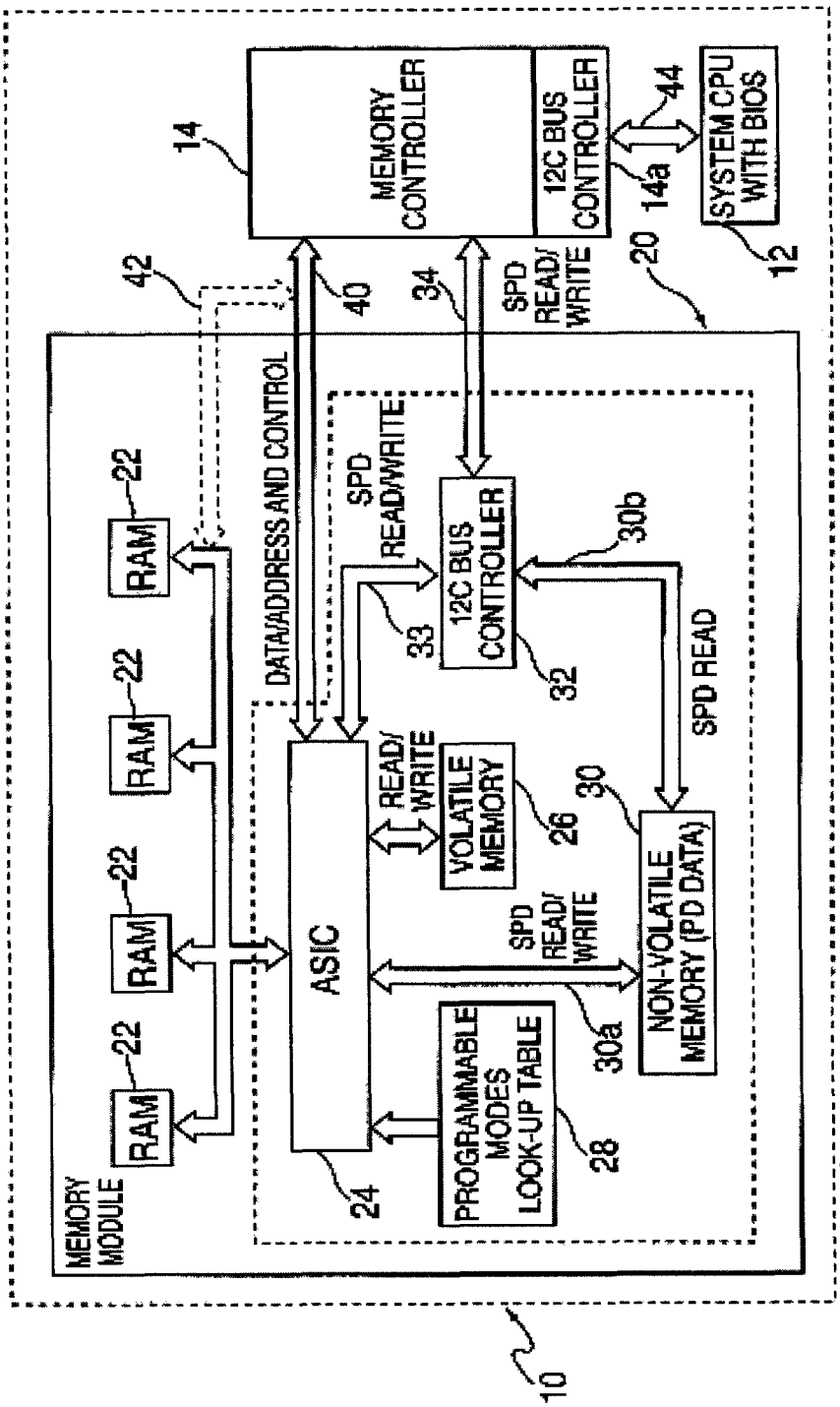


FIG. 2
(PRIOR ART)

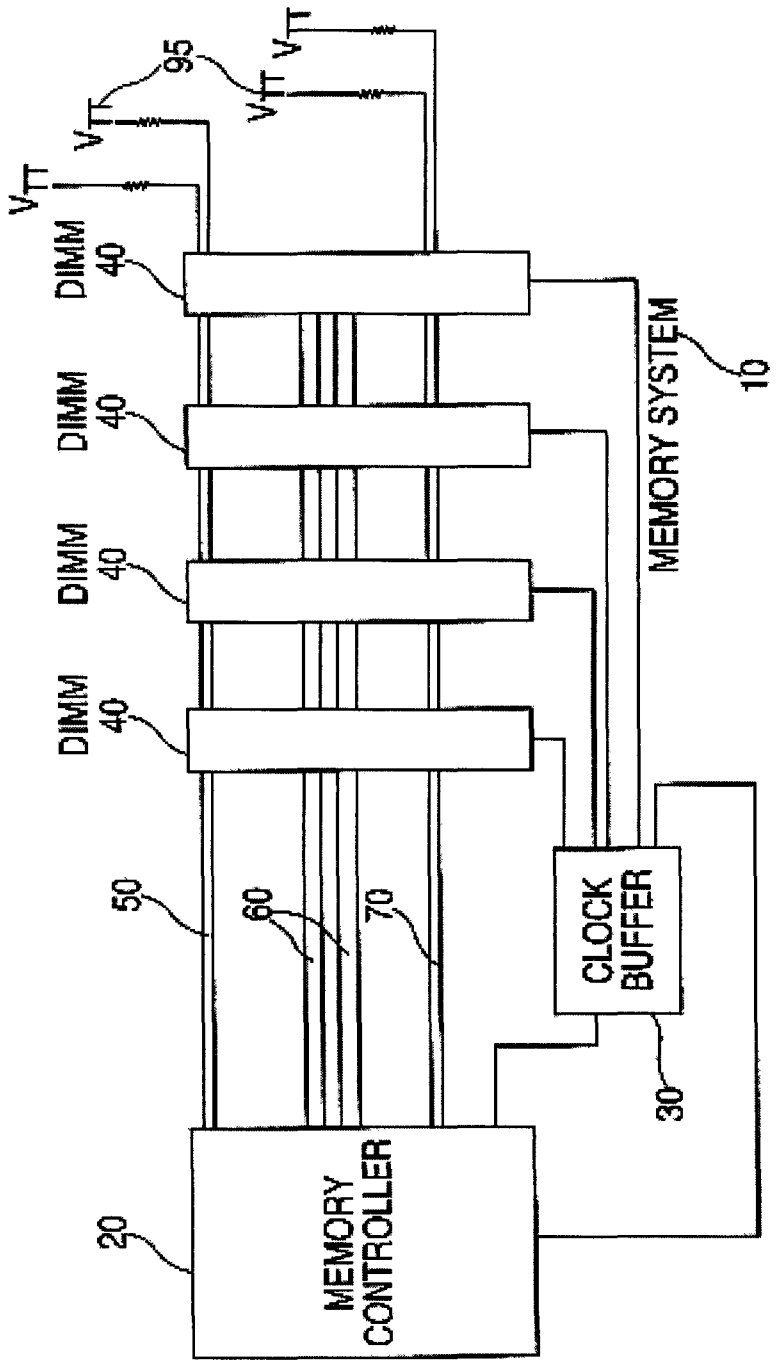


FIG. 3
(PRIOR ART)

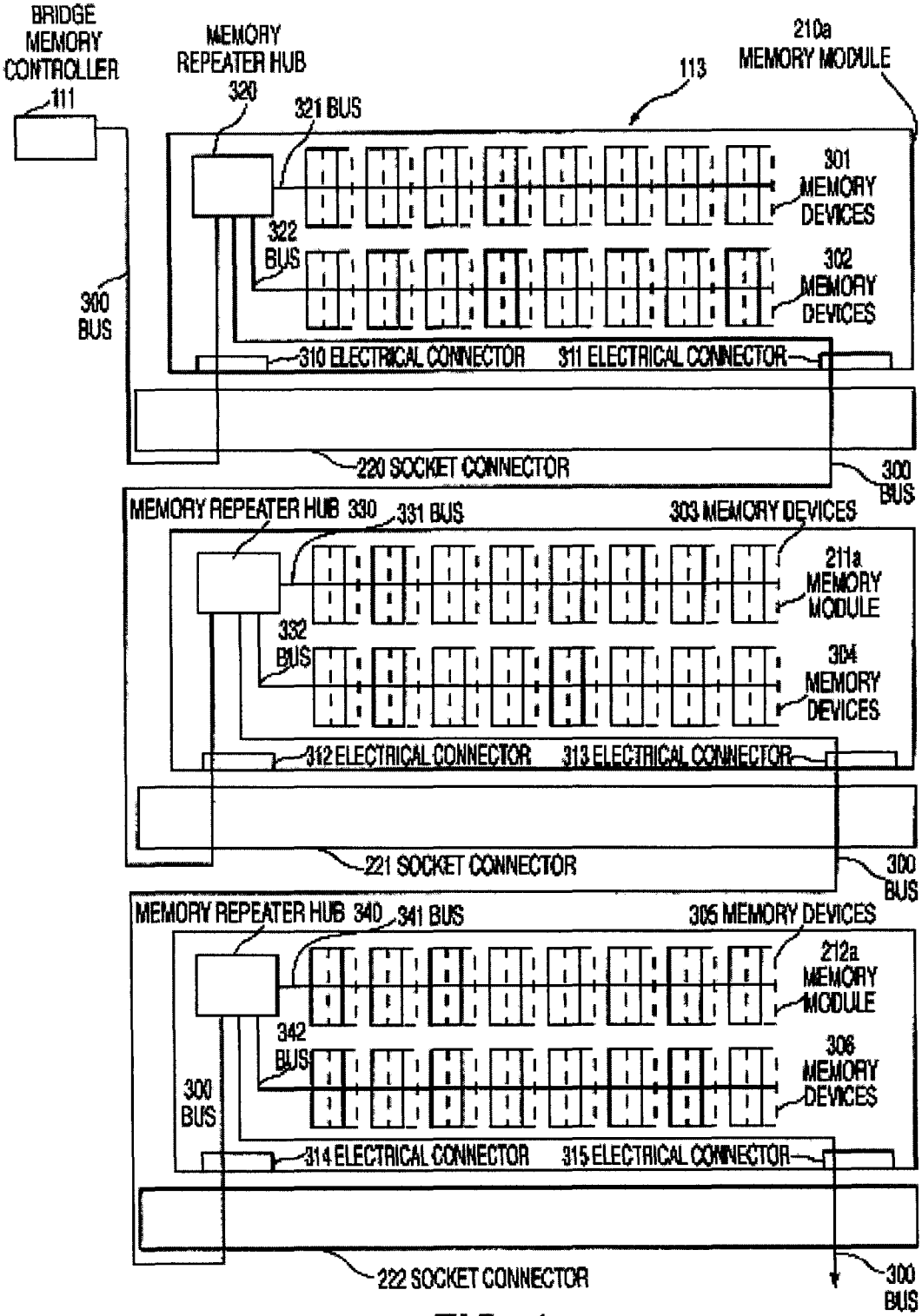


FIG. 4
(PRIOR ART)

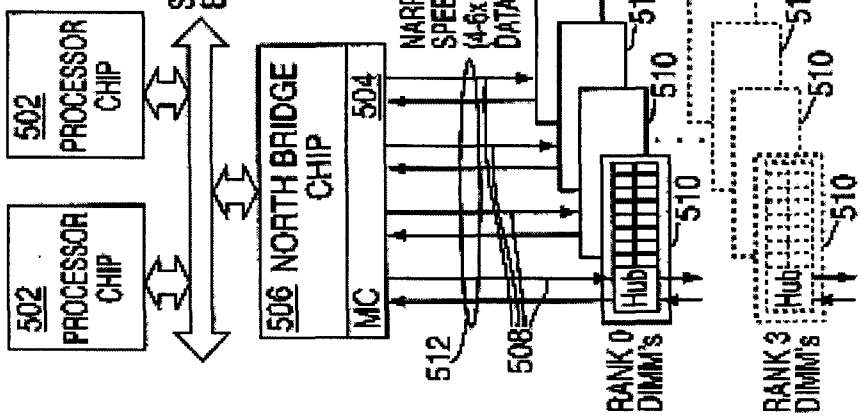

502
PROCESSOR CHIP
502
PROCESSOR CHIP
SMP BUS
506 NORTH BRIDGE CHIP
504
MC
NARROW HIGH SPEED LINKS (4-6x DRAM DATA RATE)
512
508
510
Hub
RANK 0 DIMM's
RANK 3 DIMM's

FIG. 5

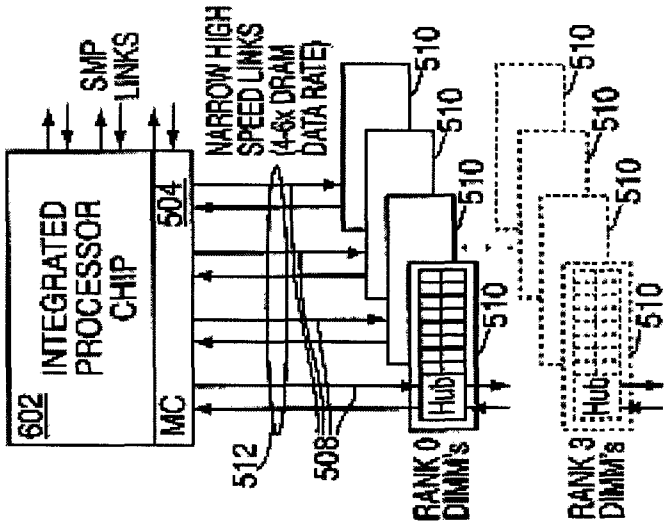

602 INTEGRATED PROCESSOR CHIP
504
MC
SMP LINKS
NARROW HIGH SPEED LINKS (4-6x DRAM DATA RATE)
512
508
510
Hub
RANK 0 DIMM's
RANK 3 DIMM's

FIG. 6

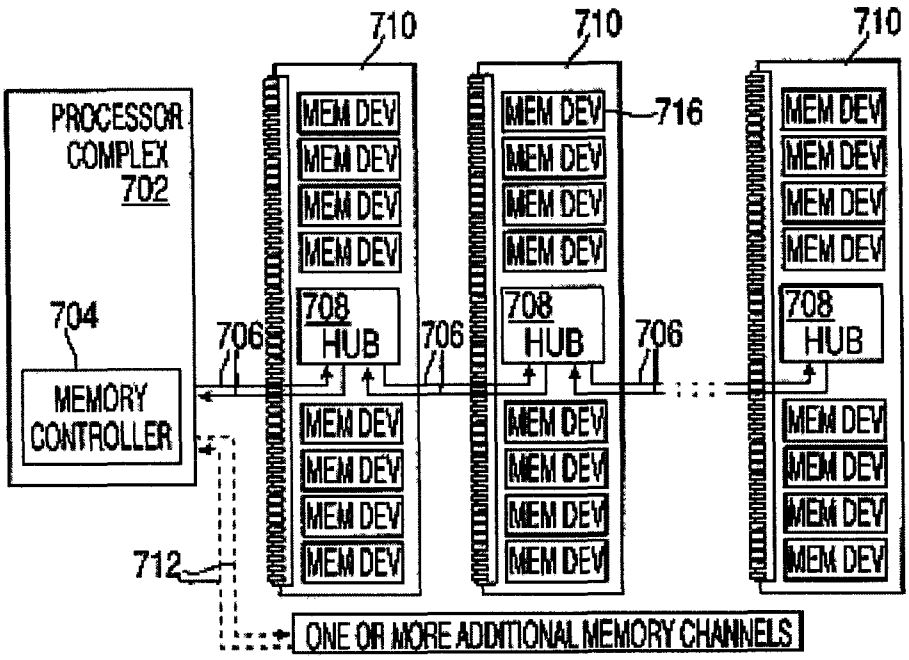

710
710
710
PROCESSOR COMPLEX 702
704
MEMORY CONTROLLER
706
706
706
MEM DEV
708 HUB
716
712
ONE OR MORE ADDITIONAL MEMORY CHANNELS

FIG. 7

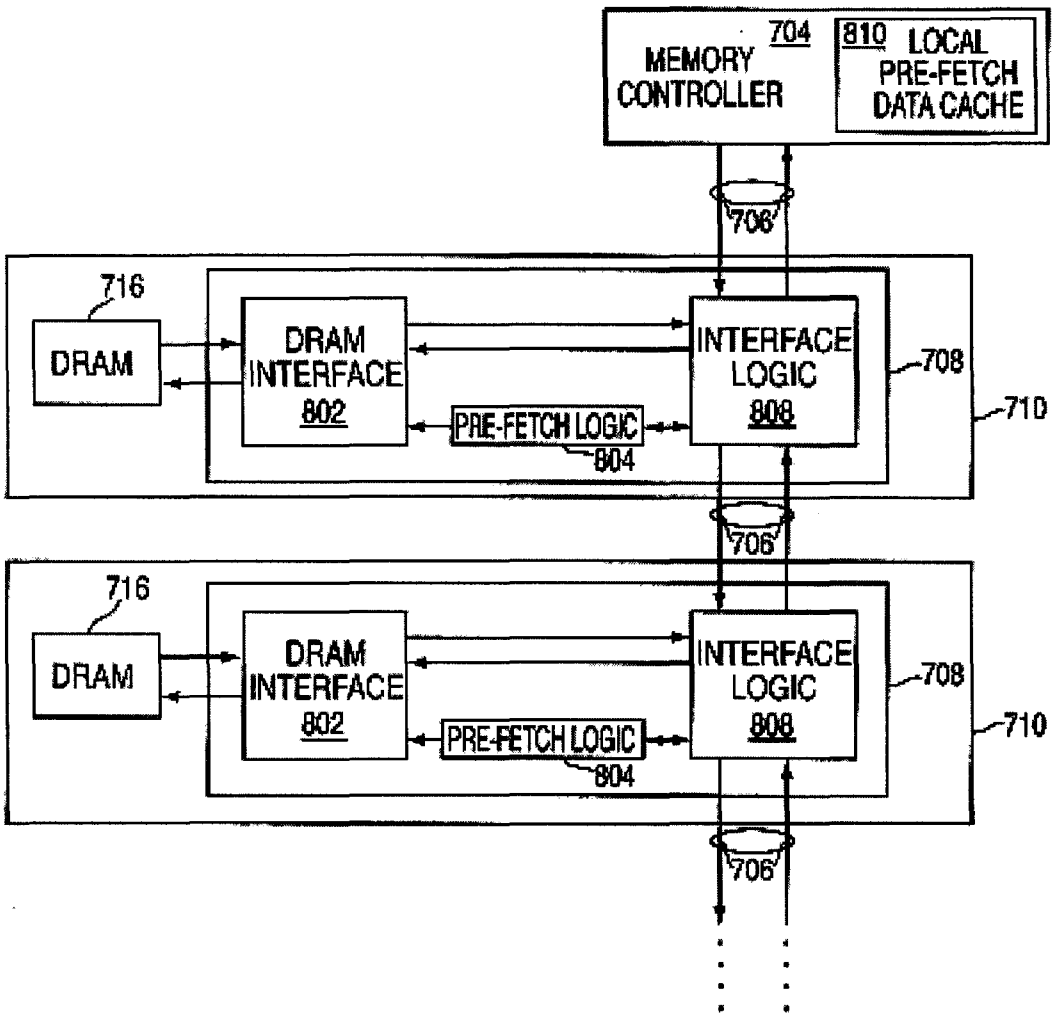

704
MEMORY CONTROLLER
810 LOCAL PRE-FETCH DATA CACHE
706
716
DRAM
DRAM INTERFACE 802
INTERFACE LOGIC 808
PRE-FETCH LOGIC 804
708
710
706
716
DRAM
DRAM INTERFACE 802
INTERFACE LOGIC 808
PRE-FETCH LOGIC 804
708
710
706

FIG. 8

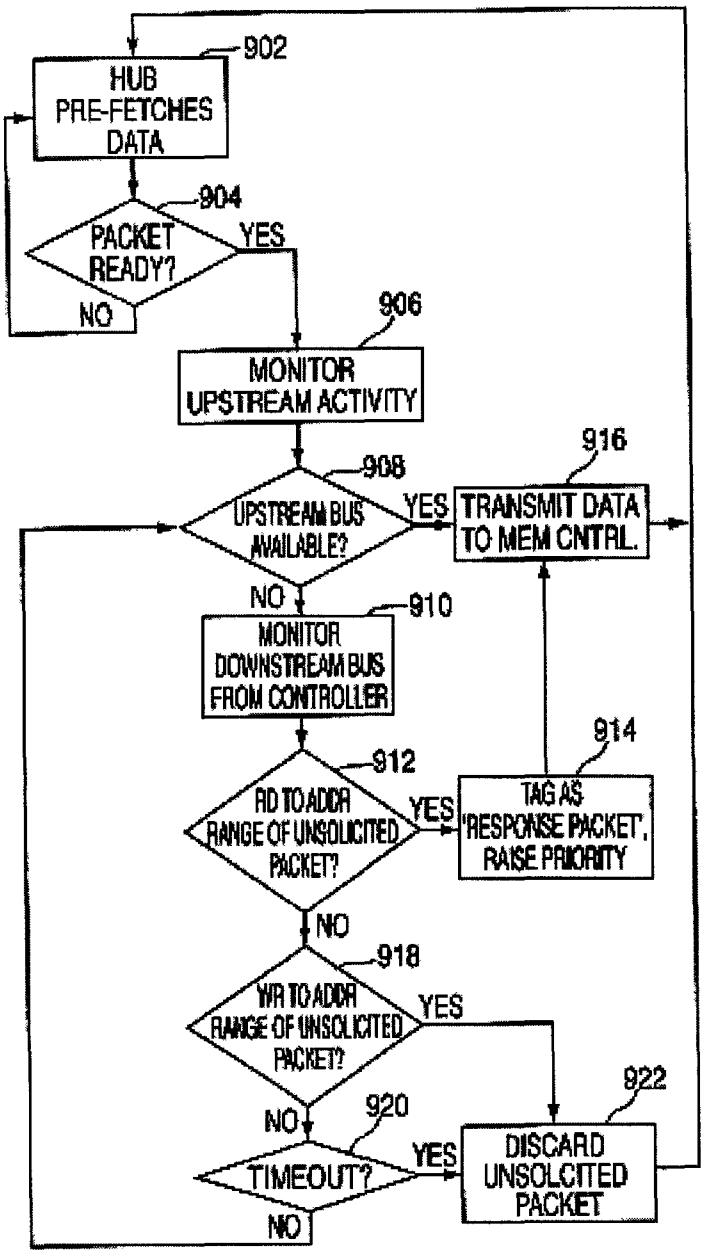

902
HUB PRE-FETCHES DATA
904
PACKET READY?
YES
NO
906
MONITOR UPSTREAM ACTIVITY
908
UPSTREAM BUS AVAILABLE?
YES
916
TRANSMIT DATA TO MEM CNTRL.
NO
910
MONITOR DOWNSTREAM BUS FROM CONTROLLER
912
RD TO ADDR RANGE OF UNSOLICITED PACKET?
YES
914
TAG AS 'RESPONSE PACKET', RAISE PRIORITY
NO
918
WR TO ADDR RANGE OF UNSOLICITED PACKET?
YES
NO
920
TIMEOUT?
YES
NO
922
DISCARD UNSOLCITED PACKET

FIG. 9

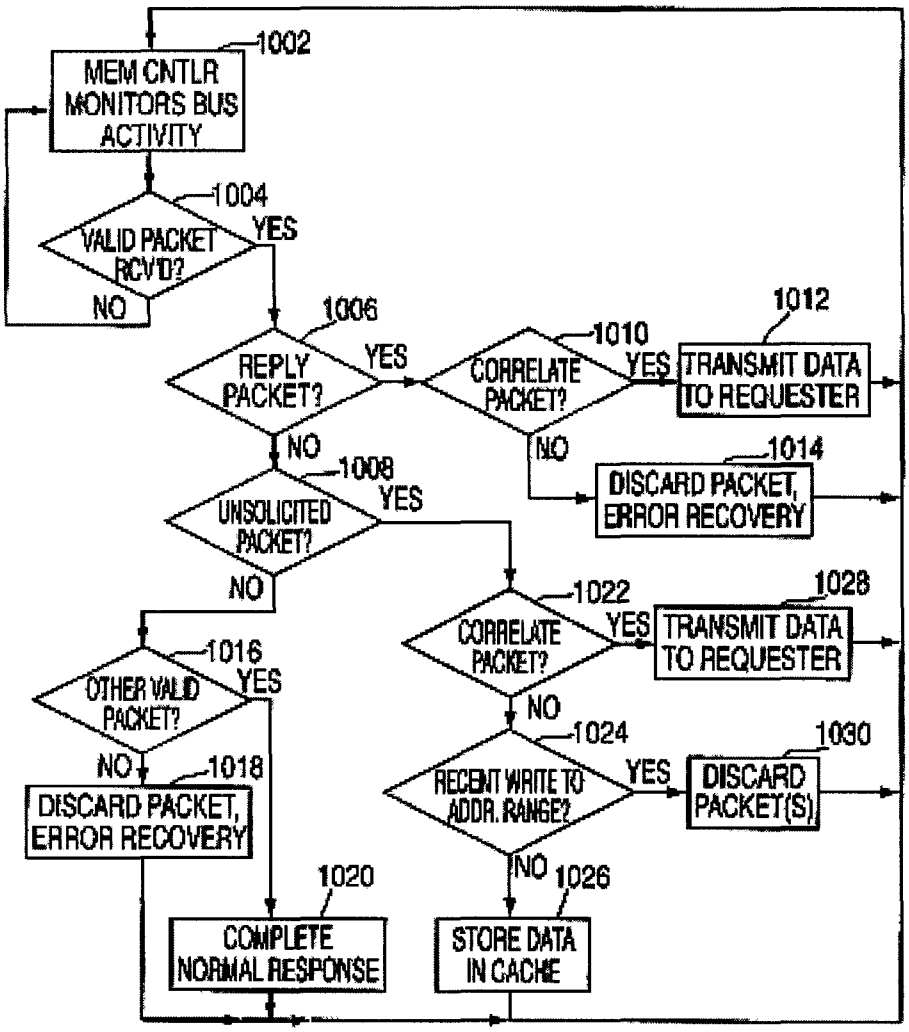


FIG. 10

		CHANNEL LANES																	
		18	17	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			ECC							STS		CMD		TAG					
TRANSFER	0																		
	1																		
	2																		
	3		DATA																
	4																		
	5																		
	6																		
	7																		

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2007/054929

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F13/16

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

B. FIELDS SEARCHED

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

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 470 734 A1 (NCR CO [US] INTEL CORP [US]) 12 February 1992 (1992-02-12) figure 1 column 2, line 47 - line 50 column 3, line 17 - line 46 column 6, line 26 - line 54	1-34
A	US 2005/071542 A1 (WEBER FREDERICK D [US] ET AL) 31 March 2005 (2005-03-31) the whole document	4,6,15, 17,21, 23,26
A	US 2004/260909 A1 (LEE TERRY R [US] ET AL) 23 December 2004 (2004-12-23) cited in the application the whole document	1-34

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
29 August 2007	06/09/2007

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	VAN DER MEULEN, E

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2007/054929

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2003/009632 A1 (ARIMILLI RAVI KUMAR [US] ET AL ARIMILLI RAVI KUMAR [US] ET AL) 9 January 2003 (2003-01-09) paragraphs [0021], [0022] abstract	1-34

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2007/054929

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0470734 A1	12-02-1992	DE 69132186 D1	15-06-2000
		DE 69132186 T2	18-01-2001
		JP 3323212 B2	09-09-2002
		JP 4233641 A	21-08-1992
		US 5530941 A	25-06-1996
US 2005071542 A1	31-03-2005	NONE	
US 2004260909 A1	23-12-2004	US 2006288172 A1	21-12-2006
US 2003009632 A1	09-01-2003	NONE	