THIS SHEET HAS BEEN REPLACED

BUS BRIDGE CIRCUIT AND METHOD USING SNOOP AHEAD OPERATIONS

FIELD OF THE INVENTION

The present invention pertains to the field of computer systems. More particularly, this invention relates to an integrated bus bridge and memory controller that enables data streaming to a shared memory resource while maintaining cache coherency.

BACKGROUND

Prior computer systems commonly include a main processor or host processor that communicates with other elements of the computer system via a processor or host bus. Other elements coupled to such a processor or host bus typically include one or more cache memories as well as a main memory for the computer system. Such a main memory is typically comprised of dynamic random access memories (DRAM).

Prior computer systems may also include one or more peripheral buses that enable communication among a variety of peripheral

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

components. Such a computer system may also include bridge circuits that enable communication between the processor or host bus and the peripheral buses. Such a bridge circuit typically enables communication between the main or host processor and the various bus agents coupled to the peripheral bus.

In addition, the bus agents coupled to such a peripheral bus may require access to the main memory of the computer system. In such a system, the data stored in the main memory may be modified either by the main processor or host processor or other elements coupled to the processor bus as well as various bus agents coupled to the peripheral buses.

In some prior computer systems, such peripheral buses usually enable extremely high speed or high bandwidth communication. For example, the peripheral component interconnect (PCI) published bus standard allows data communication bandwidth of up to 133 megabytes per second. Unfortunately, prior computer systems typically limit the available bandwidth on such a peripheral bus well below the maximum allowable provided by the published standard.

Typically, the bandwidth on such a peripheral bus is limited by the cache coherency transactions that occur on the host or processor bus during memory access transactions that initiate from the peripheral bus. For example, one prior computer system employs an integrated bridge and memory controller circuit that functions as a bus bridge

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

between the host or processor bus and a peripheral bus while at the same time functioning as a memory controller for the main memory of the computer system. Such a system typically limits each atomic data transfer transaction on the peripheral bus targeted for the main memory to a single data line or cache line.

For example, during a memory write transaction in such a system that originates on the peripheral bus, a prior integrated bridge and memory controller usually receives a data line over the peripheral bus and generates a corresponding snoop request over the processor bus. After the appropriate invalidate line or write back transaction on the host or processor bus, the integrated bridge and memory controller then transfers the data line to the main memory. Unfortunately, such a bridge and memory controller usually cannot immediately accept a subsequent data line over the peripheral bus during a data streaming transaction. Instead, such an integrated bridge and memory controller usually signals an end to the write transaction on the peripheral bus. Thereafter, the initiating bus agent on the peripheral bus typically initiates another write transaction to transfer the next data line to the main memory.

Such a system that limits atomic transfers to a single data line ensures that the integrated bridge and memory controller obtains a data line address from the originating bus agent on the peripheral bus for each data line transfer to the main memory. The integrated bridge and memory controller typically uses the address for each data line to initiate

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

a snoop request over the processor or host bus. A memory controller could generate addresses for each data line transferred over the peripheral bus in order to provide the proper snoop request transactions on the processor bus. However, such address generation logic typically increases the complexity and cost of such a memory controller and drives up the overall system cost.

SUMMARY OF THE INVENTION

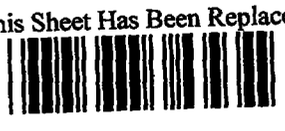
An integrated bus bridge and memory controller circuit is disclosed that enables access to a shared memory with high bandwidth data streaming. The circuit enables access to the shared memory over a processor bus and over a high bandwidth peripheral bus. The circuit performs a series of snoop ahead transactions over the processor bus during data streaming transactions to the shared memory that originate over the peripheral bus.

Other features and advantages of the present invention will be apparent from the accompanying drawings, and from the detailed description that follows below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements, and in which:

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

Figure 1 illustrates a computer system for one embodiment which comprises a processor, a cache memory, a main memory, and a bridge and memory controller circuit;

Figure 2 illustrates the bridge and memory controller for one embodiment that includes a write buffer, a peripheral read prefetch buffer, and a peripheral write buffer along with a state machine that controls snoop ahead transactions;

Figure 3 illustrates a write transaction targeted for the main memory that originates over the peripheral bus wherein the write transaction comprises an address phase followed by a series of data phases of line 0 data - line n data;

Figure 4 illustrates the format of data stored in the peripheral write buffer for one embodiment including a set of double word data values each with a corresponding snoop status flag;

Figure 5 illustrates a read transaction from the main memory over the peripheral bus wherein the read transaction comprises an address phase, and a set of wait states, followed by a set of data phases comprising line 0 data - line n data.

DETAILED DESCRIPTION

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

Figure 1 illustrates a computer system 10 for one embodiment. The computer system 10 comprises a processor 12, a cache memory 14, a main memory 18, and a bridge and memory controller 16. In addition, the processor 12 contains an internal level one (L1) cache memory (not shown).

The processor 12 coupled for communication over a processor bus 24. The cache memory 14 is also coupled to the processor bus 24 and functions as a level two (L2) cache memory for the processor 12. The main memory 18 is comprised of dynamic random access memory (DRAM) and functions as the main memory storage for the computer system 10.

The bridge and memory controller 16 functions as a memory controller for the main memory 18. In addition, the bridge and memory controller 16 functions as a cache controller for the cache memory 14. The bridge and memory controller 16 enables read and write access to the main memory 18 from the processor bus 24 as well as a peripheral bus 26.

The bridge and memory controller 16 also functions as a bus bridge between the processor bus 24 and the peripheral bus 26. The bridge and memory controller 16 enables transactions originating on the processor bus 24 to propagate to the peripheral bus 26. In addition, the bridge and memory controller 16 enables transactions originating on the

THIS SHEET HAS BEEN REPLACED

peripheral bus 26 from one of a set of bus agents 20 and 22 to propagate to the processor bus 24.

The bridge and memory controller 16 also enables the bus agents 20 and 22 to access data stored in the main memory 18. The bridge and memory controller 16 coordinates accesses to the main memory 18 that originate on the peripheral bus 26 with accesses to the main memory 18 that originate on the processor bus 24. In addition, the bridge and memory controller 16 initiates snoop request transactions on the processor bus 24 during memory accesses to the main memory 18 from the peripheral bus 26 in order to maintain cache coherency among the caches coupled to the processor bus 24. The caches coupled to the processor bus 24 include the internal L1 cache in the processor 12 along with the L2 cache memory 14.

Figure 2 illustrates the bridge and memory controller 16 for one embodiment. The bridge and memory controller 16 includes a write buffer 42, a peripheral read prefetch buffer 44, and a peripheral write buffer 46. In addition, the bridge and memory controller 16 includes a state machine 40 that coordinates data transfers between the main memory 18 and the bus agents 20 and 22 via the peripheral bus 26, and that initiates snoop request transactions to the cache memories on the processor bus 24.

The write buffer 42 buffers write data targeted for the main memory 18 that originates from agents coupled to the processor bus 24

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

and agents coupled to the peripheral bus 26. A multiplexer 48 controlled by the state machine 40 selects either a data path from the processor bus 24 or a data path from the peripheral bus 26. The write buffer 42 compensates for the latency of DRAM accesses via a DRAM write path 52 in comparison to the speed of data transfer via the processor bus 24 or the peripheral bus 26.

The peripheral write buffer 46 buffers write data targeted for the main memory 18 that originates from bus agents on the peripheral bus 26. The peripheral write buffer 46 stores write data along with flags that are maintained by the state machine 40. The flags maintained by the state machine 40 indicate whether the data stored in the peripheral write buffer 46 corresponds to data lines having a completed snoop request transaction over the processor bus 24. The state machine 40 merges the data from the peripheral write buffer 46 into the write buffer 42 through the multiplexer 48 only after the corresponding snoop request transaction for the data line is complete over the processor bus 24.

The peripheral read prefetch buffer 44 prefetches sequential data lines from the main memory 18 in preparation for read transactions on the peripheral bus 26. The state machine 40 controls prefetches into the peripheral read prefetch buffer 44 to ensure that a read to the main memory 18 via a read path 54 does not occur until after the snoop transaction for the corresponding data line as completed via the processor bus 24.

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

Figure 3 illustrates a write transaction targeted for the main memory 18 that originates over the peripheral bus 26. As shown, the write transaction comprises an address phase followed by a series of data phases comprising line 0 data - line n data. During the address phase, the peripheral bus 26 carries the address of data line 0 for the write transaction. Subsequently, the phases for the line 0 data - line n data each transfer an amount of data via the peripheral bus 26 equal to a cache line in the cache memory 14 or the internal cache of the processor 12.

To enable a high bandwidth write transactions to the main memory 18 from the peripheral bus 26, the state machine 40 performs a snoop ahead transaction for each data line in the write transaction. For example, during the address phase of the write transaction on the peripheral bus 26, the state machine 40 issues a line 0 snoop request over the processor bus 24. Thereafter, the cache memory 14 and the internal cache of the processor 12 each perform either a writeback operation, also known as a copyback transaction, or an internal invalidate cache line operation depending upon the clean or dirty status of the cache line specified by the line 0 snoop request.

Thereafter, during the line 0 data phase of the write transaction on the peripheral bus 26, the line 0 snoop transaction on the processor bus 24 completes. As a consequence, the line 0 data buffered in the peripheral write buffer 46 is marked by the state machine 40 with a snoop done (SD) flag. Initially, the state machine 40 stores a snoop not

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

done (SND) flag for the line 0 data contained in the peripheral write buffer 46 prior to the completion of the line 0 snoop request on the processor bus 24.

Later during the line 0 data phase, the state machine 40 generates a next sequential address corresponding to the line 1 data. Thereafter, the state machine 40 issues a line 1 snoop request over the processor bus 24. As shown, the state machine 40 continuously issues snoop request transactions over the processor bus 24 prior to the reception of the corresponding data line over the peripheral bus 26 during the write transaction. The snoop ahead transactions issues by the state machine 40 prevent the occurrence of wait cycles or delay cycles on the peripheral bus 26 during write transactions targeted for the main memory 18 through the peripheral write buffer 46.

Figure 4 illustrates the format of data stored in the peripheral write buffer 46 for one embodiment. The peripheral write buffer 46 stores a set of double word data values each with a corresponding snoop status flag. The snoop status flag indicates either snoop done (SD) or snoop not done (SND) depending upon the status of the corresponding snoop ahead request issued by the state machine 40 over the processor bus 24. The boundary in the peripheral write buffer 46 between data having an SD indication and data having an SND indication provides a snoop boundary between line n-1 and line n. The state machine 40 merges the data from the peripheral write buffer 46

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

into the write buffer 42 only if the corresponding snoop status flag indicates snoop done.

Figure 5 illustrates a read transaction from the main memory 18 over the peripheral bus 26. The read transaction on the peripheral bus 26 comprises an address phase, and a set of wait states, followed by a set of data phases comprising line 0 data - line n data. During the address phases of a read transaction, the originating bus agent on the peripheral bus 26 transfers the address of the initial data line being read from the main memory 18. As shown during the address phase, the state machine 40 issues a line 0 snoop request over the processor bus 24 that specifies the data line address provided during the address phase.

During the wait states on the peripheral bus 26, the line 0 snoop transaction on the processor bus 24 completes. Thereafter, a DRAM latency occurs while the peripheral read prefetch buffer 44 receives the corresponding data line from the main memory 18 via the read path 54. The line 0 prefetch to the peripheral read prefetch buffer 44 completes before the start of the line 0 data phase on the peripheral bus 26.

During the line 0 data phase on the peripheral bus 26, the peripheral read prefetch buffer 44 transfers the buffered read data over the peripheral bus 26 while the state machine 40 issues a line 1 snoop request over the processor bus 24 using an address generated as a next sequential cache line address from the address specified during

This Sheet Has Been Replaced



THIS SHEET HAS BEEN
REPLACED

the preceding address phase. Later in the line 0 data phase of the read transaction on the peripheral bus 26, the line 1 snoop transaction on the processor bus 24 completes. The completion of the line 1 snoop request enables a prefetch of the corresponding data line from the main memory 18 into the peripheral read prefetch buffer 44. Thereafter, the line 1 data from the peripheral read prefetch buffer 44 is transferred over the peripheral bus 26.

In addition, the state machine 40 monitors the occurrence of wait states on the peripheral bus 26 during read and write transactions. The state machine 40 counts the number of wait states that occur on the peripheral bus 26. The state machine 40 implements a disconnect time-out mechanism such that a disconnect is signaled over the peripheral bus 26 if the number of consecutive wait states on the peripheral bus 26 exceeds eight. The eight wait states conforms to the peripheral component interface (PCI) specification for maximum latency during data transfers. The disconnect indication on the peripheral bus 26 causes the bus agent that initiated the read transaction or the write transaction to terminate the transaction and try again later.

In the foregoing specification the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and

This Sheet Has Been Replaced



THIS SHEET HAS BEEN
REPLACED

drawings are accordingly to be regarded as illustrative rather than a restrictive sense.

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

* C L M S *

CLAIMS

What is claimed is:

1. A circuit that enables access to a memory over a processor bus and over a peripheral bus and that performs a series of snoop ahead transactions over the processor bus during access transactions to the memory that originate over the peripheral bus.

2. The circuit of claim 1, further comprising a write buffer for buffering write data targeted for the memory and circuitry that merges write data received over the processor bus with write data received over the peripheral bus into the write buffer.

3. The circuit of claim 2, further comprising a peripheral write buffer that buffers write data received over the peripheral bus and that stores a snoop done flag for the write data that indicates whether a corresponding snoop ahead transaction for the write data is complete.

4. The circuit of claim 2, further comprising a peripheral read prefetch buffer that prefetches read data from the memory during read transactions over the peripheral bus only after a corresponding snoop ahead transaction for the read data is complete.

5. The circuit of claim 2, further comprising circuitry for controlling a cache memory coupled to the processor bus.

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED



6. A computer system, comprising:

processor and a cache memory each coupled to a processor bus;

a set of bus agents coupled to a peripheral bus;

circuit that enables access to a memory via the processor bus and via the peripheral bus and that performs a series of snoop ahead transactions over the processor bus during access transactions to the memory that originate over the peripheral bus.

7. The computer system of claim 6, wherein the circuit includes a write buffer for buffering write data targeted for the memory and circuitry that merges write data received over the processor bus with write data received over the peripheral bus into the write buffer.

8. The computer system of claim 7, wherein the circuit includes a peripheral write buffer that buffers write data received over the peripheral bus and that stores a snoop done flag for the write data that indicates whether a corresponding snoop ahead transaction for the write data is complete.

9. The computer system of claim 7, wherein the circuit includes a peripheral read prefetch buffer that prefetches read data from the memory during read transactions over the peripheral bus only after a corresponding snoop ahead transaction for the read data is complete.

10. The computer system of claim 7, wherein the circuit includes for controlling the cache memory coupled to the processor bus.

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED



11. A method for accessing a shared memory comprising the step of performing a series of snoop ahead transactions over a processor bus during access transactions to the shared memory that originate over a peripheral bus.

12. The method of claim 11, further comprising the steps of buffering write data targeted for the shared memory and merging write data received over the processor bus with write data received over the peripheral bus.

13. The method of claim 12, further comprising the steps of buffering write data received over the peripheral bus and storing a snoop done flag for the write data that indicates whether a corresponding snoop ahead transaction for the write data is complete.

14. The method of claim 12, further comprising the step of prefetching read data from the shared memory during read transactions over the peripheral bus only after a corresponding snoop ahead transaction for the read data is complete.

This Sheet Has Been Replaced

