

- 2 -

otherwise inactive) another microprocessor in the system may continue to perform data transactions on the system bus. The problem is that some the bus transactions may attempt to read/write data stored in a modified state in a powered-down or otherwise inactive microprocessor. Unless there exists some mechanism for monitoring bus activity and updating shared memory locations, data coherency will be lost. Therefore, MP computer systems have a need for a mechanism which makes inactive processors both aware and responsive to bus activity that may attempt to access stale data.

As will be seen, the present invention provides a multiprocessing computer system in which individual processors monitor bus traffic to maintain cache coherency while operating in a reduced power mode. In accordance with the present invention, an inactive or powered-down processor responds to certain bus transactions by writing back modified data to the system bus in a reduced power mode of operation. In addition, the invention functions without latency or intervention from the operating system. Thus, the invention provides a totally transparent way of interacting with an external bus while minimizing power consumption.

- 4 -

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood more fully from the detailed description which follows and from the accompanying drawings, which, however, should not be taken to limit the invention to the specific embodiments shown, but rather are for explanation and understanding only.

Figure 1 is a generalized block diagram of a microprocessor operating in accordance with the present invention.

Figure 2 is a diagram showing state transitions for the clock control architecture utilized in one embodiment of the present invention.

Figure 3 is a timing diagram illustrating the operation of one aspect of the present invention.

Figures 4A - 4D are a set of diagrams illustrating various exemplary operations performed in one embodiment of the present invention.

Figure 5 illustrates a test register which includes bits allowing software to disable certain features in accordance with the present invention.

- 6 -

signal on line 45 is also coupled to a portion of interrupt logic unit 29 and bus unit 40. (The distinction between the ICLK and CLK clock signals is discussed in more detail below.) During normal operation microprocessor 20 operates with a clock (i.e., ICLK and CLK) frequency of approximately 100 MHz. Variations of processor 20 may also support a variety of core/bus ratios. For instance, in alternative implementations bus frequencies of 50 MHz and 60 MHz are supported.

In one embodiment, bus unit 40 and interrupt logic unit 29 include logic circuitry enabling microprocessor 20 to be utilized in a MP system. By way of example, bus cycles issued and received by bus unit 40, as well as interrupt signals received and provided by interrupt logic unit 29, are compatible with the well-known Pentium™ processor. Microprocessor 20 also includes logic that maintains cache consistency and which arbitrates for an external bus.

In one embodiment, microprocessor 20 also includes power management features which allow the processor to enter a reduced power dissipation state (i.e., a HALT or STANDBY mode of operation). When functioning in a normal mode of operation at peak power, microprocessor 20 may dissipate as much as 10 watts; in a HALT or STANDBY mode of operation, as little as 700 milliwatts of power is consumed. One of the ways in which microprocessor 20 implements power management functions, as well as operating system independent functions, is by a system management mode of operation. System management mode (SMM) consists of an interrupt (SMI), an alternative address space and an instruction (SRET). The system management interrupt causes a system management interrupt request to be latched within logic unit 29. When the latch SMI# is recognized on an instruction boundary, microprocessor 20 enters SMM.

With continuing reference to Figure 1, in one implementation microprocessor 20 includes an external pin (labeled STPCLK#) and associated circuitry that can be used to regulate the power of the microprocessor. The STPCLK# pin is disclosed more fully in co-pending application entitled, "Method And Apparatus For Asynchronously Stopping The Clock In A Processor", Serial No. 07/970,576, filed

- 8 -

state in which both the ICLK and CLK signals are coupled to all portions of the IC's internal logic.

The STOP_GRANT state, represented by block 53, can be entered by asserting the external STPCLK# pin of microprocessor 20. In the STOP_GRANT state the integrated circuit operates in a reduced power consumption mode wherein most of the internal functional units of the processor are deactivated (i.e., ICLK off). On the other hand, bus unit 40, the tag arrays of caches 25 and 26, PLL 30 and portions of the interrupt logic unit 29 remain active. Once the STOP_GRANT bus cycle has been placed on the bus and the BRDY# signal is returned, the processor is in the STOP_GRANT state. Practitioners in the art understand the BRDY# signal to indicate that the external system has presented valid data on the data pins in response to a read, or that the external system (e.g., main memory, other processor, etc.) has accepted the processor data in response to a write request. In one embodiment, the processor returns to the normal execution state approximately ten clock periods after STPCLK# has been deasserted. A RESET also brings the processor out of the STOP_GRANT state, returning it to the normal state.

Processor 20 recognizes inputs to bus unit 40 for maintaining cache coherency (e.g., invalidation's and external snoops) by monitoring bus traffic when in a reduced power mode of operation. For example, while in the STOP_GRANT state, the processor latches transitions on external interrupt signals (e.g., SMI#, NMI, INTR, FLUSH#, R/S# and INIT). All of these interrupts are taken after the de-assertion of STPCLK#; that is, upon re-entering the normal state.

The AUTO_HALT state, denoted by block 50 in Figure 2, is entered whenever the HALT instruction is executed. In AUTO_HALT state the internal clock (ICLK) is decoupled from a majority of the internal logic while the continuous clock signal CLK keeps selected functional units of the chip operational. Upon the occurrence of the INTR, NMI, SMI#, RESET or INIT, the processor transitions back to the normal operating state, i.e., state 51. As explained earlier, the AUTO_HALT state results in a significant reduction in power

- 10 -

processor snoops -- by keeping the cache units tag comparison logic enabled by CLK. This is illustrated in Figure 1, wherein the CLK signal on line 45 is coupled to the tag comparison logic of caches 25 and 26. On the other hand, the ICLK signal is disabled to the data array portions of the caches to minimize power consumption. When a snoop occurs, the MESI (i.e., Modified, Exclusive, Shared, Invalid) cache protocol bits are updated when the need arises. Portions of the processor (e.g., at least the data cache, associated cache control logic, and buses between the data cache and bus interface) other than the tag arrays are awakened only if a write-back cycle is required. In an alternative embodiment, more aggressive power saving may be achieved by disabling the tag comparison logic in certain situations, e.g., all the cache entries are invalid, or to power down the tag arrays until a snoopable transaction has been detected then provide a slightly increased power state to allow the tags to perform the snoop operation.

Note that Figures 1-4 illustrate only one exemplary embodiment of the present invention; however, more complex implementations are possible to provide a variety of power saving techniques involving a variety of different clocking schemes. The same is true with respect to the particular protocol employed once snooping has been performed. For instance, instead of having the first processor write back the data on the system bus, the first processor might absorb the write and keep the dirty line. Thus, practitioners in the art will appreciate the many different ways that the present invention may be implemented.

Figure 3 illustrates a timing diagram showing the latency between a STPCLK# request and the STOP_GRANT bus cycle. Note that for the illustrative embodiment, there is a delay of approximately ten clocks between the STPCLK# request and the STOP_GRANT bus cycle. This latency is dependent upon the current instruction, the amount of data in the CPU write buffers, and system memory performance.

Data cache 26 of processor 20 utilizes the MESI protocol to enforce cache consistency. A line in the data cache can be in the Modified, Exclusive, Shared or Invalid state, whereas a line in instruction cache 25 can be either in the Valid or Invalid state. The

- 12 -

also asserts a private arbitration pin, BPBREQ#, to indicate that processor PB has generated a bus request (in this example, it is assumed that processor PA currently owns external bus 60). Note that a HITM# signal is also coupled to bus 60 to inhibit another bus master from accessing the data until after the line has been completely written back. Processor PA then completes the write cycle on external bus 60 as if processor PB did not exist.

The external snoop happens just as the write cycle completes on bus 60, but before processor PB has had a chance to write the modified data back to system memory. Figure 4B illustrates processor PB asserting the HITM# signal to inform the system that the snoop address is cached in the dual processor pair and is in a Modified state. The external snoop in this example hits the same line that caused the PHITM# signal to be asserted.

In Figure 4C, an arbitration exchange has occurred on external bus 60 and processor PB now owns the bus. At this point, processor PB writes back the M state line. From the viewpoint of the computer system, it appears as if a single processor was completing the snoop transaction. Note that in Figure 4C, two private arbitration pins coupled between the two processors are utilized to indicate that bus ownership has been granted to processor PB (or that processor PA requests ownership back after completion of the writeback operation).

Finally, in Figure 4D processor PA reruns the original write cycle after processor PB has granted the bus back to processor PA. It is important to recognize that processor PB has remained in a reduced power consumption mode of operation throughout the entire snoop and writeback processes represented by Figures 4A - 4D. This is a key feature of the present invention since it provides a considerable advantage in computer systems limited by power consumption requirements, but yet having a need for cache coherency.

With reference now to Figure 5, there is shown a specialized test register 12 which includes bits to allow software to disable certain features of microprocessor 20. For example, the AUTO_HALT feature may be disabled to setting bit 6 in register 12 to a "1". In this setting, the

**CLAIMS**

What is claimed is:

1. A multiprocessor computer system comprising:
a bus;
a main memory coupled to the bus;
a first processor coupled to the bus for performing data transactions with the main memory;
a second processor having a bus unit coupled to the bus, a local cache with an associated tag array, a phase-locked loop (PLL) for generating a clock signal, and a means for reducing a power consumption by decoupling the clock signal from a portion of the internal logic of the second microprocessor while the bus unit, PLL and associated tag array remain coupled to the clock signal;
the bus unit of the second microprocessor further including means for snooping the bus when in a reduced power mode of operation and means for signaling to the first microprocessor that a write cycle on the bus by the first microprocessor is to a modified cache line in the local cache of the second microprocessor.
2. The multiprocessor computer system of claim 1 wherein the second microprocessor further comprises means for writing back the modified cache line to the main memory on the bus following completion of the write cycle of the second microprocessor.
3. The multiprocessor computer system of claim 2 wherein the first microprocessor further comprises means responsive to the signaling means of the first microprocessor for regenerating the write cycle after the second microprocessor completes write-back of the modified cache line.



-16-

8. The multiprocessor computer system of claim 7 wherein the register further includes a third bit which, when set, disables the interrupt control unit.