METHOD AND APPARATUS FOR POWER MODE TRANSITION IN A MULTI-THREAD PROCESSOR

FIELD

The invention relates to processor power modes. More particularly, the invention relates to a method and apparatus for power mode transition in a multi-thread processor.

BACKGROUND

A processor consumes power as it performs various functions, such as mathematical operations. The amount of power used by the processor will impact, for example, how long a battery in a mobile computer will last. Designers, therefore, have attempted to limit the power used by a processor.

Even when not performing mathematical operations, the generation and distribution of internal clock signals that synchronize the processor's operation will consume a considerable amount of power. To save power, a processor may be designed to operate in a reduced power state when inactive. In the reduced power state, all but a few internal clocks are turned off, which saves power and may extend the life of a battery.

For example, a "sleep" power mode allows most of the internal clocks on a chip to be turned off when the system is idle. A processor may be in the sleep mode, for example, when the system is waiting for an external event, such as the opening of a lid on a laptop computer, or when the computer operating system (OS) is waiting

for an Input/Output (I/O) operation.

To aid in energy efficient computing, in some implementations the processor is placed into an even lower power state referred to as a "deep sleep" power mode. The deep sleep mode may be entered, for example, by stopping a clock input signal to the processor after the processor has entered the sleep power mode. This allows the processor to maintain the operational state of elements in the chip, but only draws power equivalent to the processor's leakage current.

With highly complex processors, such as out-of-order processors, some internal "clean-up" may be desired before the internal clocks are disabled. Such clean up is typically performed by micro-code which, for example, cleans up the operational state, drains queues, puts the processor to sleep and waits for an event, or "alarm," that marks the end of the hibernation.

A "stop grant" power mode, which itself is a low power mode, is typically used for this clean up process. Typically, such a processor transitions from an active mode to the stop grant mode, based on, for example, a pin on the chip being asserted by a "chipset" that controls the processor. When the desired operations have been performed by the micro-code in the processor, the processor sends the chipset an indication, such as a stop grant acknowledge Special Bus Cycle (SBC) over a bus. The chipset may then transition the processor into the sleep and deep sleep modes by, for example, asserting another pin or stopping the bus clock.

SUMMARY

In accordance with an embodiment of the present invention, a first indication is issued, including a first identifier associated with a first logical processor in a processor, that the first logical processor has entered a power mode. A second indication is also issued, including a second identifier associated with a second logical processor in the processor, that the second logical processor has entered the power mode.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of MT processor power mode states according to an embodiment of the present invention.

FIG. 2 is a block diagram of a computer system according to an embodiment of the present invention.

FIG. 3 is a timing diagram of MT processor power modes, and associated signals, according to an embodiment of the present invention.

FIG. 4 is a block flow diagram of a method to transition MT processor power modes according to an embodiment of the present invention.

FIG. 5 is a block diagram of a computer system having two MT processors according to an embodiment of the present invention.

DETAILED DESCRIPTION

An arrangement such as the ones described in the background section, however, may pose problems when used in a shared resources multi-processing environment, which allows a single processor to perform several mathematical operations substantially simultaneously. For example, a multi-thread (MT) processor contains several independent "logical processors," or "threads," and it is possible for a one logical processor to enter a power mode, such as the stop grant mode, while the other logical processor remains in the active mode. This may be thought of as a thread sleep, in which a single thread is put to sleep. All clock signals in the MT processor should not be turned off if even one thread is still in the active mode because the operations performed by that thread may still need synchronization. When every logical processor in a MT processor enter thread sleep state, the clocks on the MT processor may be turned off.

It is possible that such a MT processor may be designed to send a stop grant acknowledge SBC as soon as one thread enters the stop grant mode. This, however, may cause a problem because the chipset might try to force the MT processor into the sleep mode, even though one of the threads is not ready. The chipset, of course, could be designed to wait a predetermined amount of time before the MT processor enters sleep mode, but this would slow the process down, and there is still no

guarantee that the other thread will be ready

Another approach would be to have the processor issue a stop grant acknowledge SBC only when all of the threads in the processor have completed the clean up process. This, however, is difficult because the micro-code would have to be "thread aware." That is, the micro-code would have to execute different commands, such as commands to issue a stop grant acknowledge SBC, depending on the thread on which it is executing.

Moreover, some computer systems will have a number of MT processors. The power mode transitions in such a case will have to account for, and keep track of, multiple threads running on multiple processors.

In view of the foregoing, it can be appreciated that a need exists for a method and apparatus to address the problems discussed above.

An embodiment of the present invention is directed to a method and apparatus for power mode transition in a MT processor. Referring now in detail to the drawings wherein like parts are designated by like reference numerals throughout, FIG. 1 shows a block diagram of MT processor power mode states according to an embodiment of the present invention. When the MT processor is in the active power mode 110 the processor's internal clocks are running causing a relatively large amount of power to be consumed. When the MT processor is to be transitioned to an energy efficient mode, a chipset controlling the processor may assert a signal on a stop clock pin (STPCLK#).

When the MT processor samples the signal on the stop clock pin as "asserted," stop clock micro-code running in the MT processor will clean up the appropriate operational states and set up the correct "break events," or events that will cause the MT processor to wake up. Eventually, as a thread goes to sleep the micro-code associated with that thread stops running. When the threads in the MT processor are asleep, the hardware turns some of the internal clocks off to reduce the amount of power being used. It should be noted that the core clocks may actually be left running, as in a debug mode, or the clock may be turned on to process a "snoop," in which case the processor will respond normally to the inquiry. When the processor senses a break event, it turns the internal clocks back on and returns to the

active power 110 mode.

According to this particular embodiment of the present invention, when the stop clock micro-code executes for one logical processor in the MT processor, a stop grant acknowledge SBC is issued, including an identifier associated with that particular logical processor. When the micro-code for all of the logical processors have executed, the MT processor may enter the stop grant mode. The chipset may then assert the signal on the sleep pin (SLP#), which places the processor in a sleep mode 130. After waiting an appropriate amount of time, the chipset may turn off the clocks by stopping a clock input signal to the processor (BCLK). This places the processor in a deep sleep power mode 140. As is also shown in FIG. 1, the processor may be returned to the active power mode 110 by, for example, starting the BCLK, de-asserting the SLP# and de-asserting the STPCLK#.

FIG. 2 is a block diagram of a computer system according to an embodiment of the present invention. The computer system includes a MT processor 200 and a chipset 300 which communicate over a bus 250. The MT processor 200 contains a first logical processor, or "thread," 210 with an associated processor identifier signal 1. The MT processor 200 also contains a second logical processor 220 with an associated processor identifier signal 2. Although the MT processor 200 shown in FIG. 2 has two logical processors 210, 220, an alternative embodiment of the present invention may be used with processors having any number of threads. Similarly, any numbering or labeling system could be used for the processor identifier signals or numbers. In addition to communicating over the bus 250, the chipset 300 may assert a signal on the stop clock pin, a sleep pin, and may start and stop the bus clock to control the MT processor 200.

When one of the logical processors 210, 220 enters a thread sleep state, the associated architectural state is saved and some of the resources may be used by the other thread. As a logical processor 210, 220 completes the STPCLK# micro-code flow, it issues a stop grant acknowledge SBC over the bus 250, including the identifier signal of the associated thread. Thus, the MT processor 200 produces two SBCs before the clocks are turned off.

Refer now to FIG. 3, which is a timing diagram of MT processor power

modes, and associated signals, according to an embodiment of the present invention. As described above, after the signal on the stop clock pin is asserted by the chipset 300, each thread 210, 220 in the MT processor 200 will issue a separate stop grant acknowledge SBC, including the logical processor ID signal associated with that thread. These are shown in FIG. 3 as the first and second thread stop grant acknowledge SBCs. After all of the threads have issued a stop grant acknowledge SBC, the MT processor 200 may transition from the active state to the stop grant state. Between the time the stop clock pin is asserted and the last logical processor issues a stop grant acknowledge SBC, the state of the MT processor 200 should be considered indeterminate. The chipset 300 should therefore wait for a stop grant acknowledge SBC from the threads before asserting the sleep pin. As described above with respect to FIG. 1, after waiting an appropriate amount of time the chipset 300 may transition the MT processor 200 to the deep sleep mode by turning off the bus clock input signal to the MT processor 200.

FIG. 4 is a block flow diagram of a method to transition MT processor power modes according to an embodiment of the present invention. After beginning in the active mode at 410, the chipset 300 asserts the signal on the stop clock pin at 420. The chipset 300 then waits until stop grant acknowledge SBCs have been received from both threads at 430 and 440. the chipset 300 may use the identifier associated with the stop grant acknowledge SBCs to decide which threads have completed the stop grant micro-code. Only when both stop grant acknowledge SBCs have been received will the chipset 300 assert the sleep pin at 450, causing the MT processor 200 to enter the sleep mode, before completing at 490. Although not shown in FIG. 4, the chipset 300 may then transition the MT processor 200 to the deep sleep mode by waiting an appropriate amount of time and turning off the bus clock input to the MT processor 200.

Using such a method, confusion about the power mode state of the MT processor 200 may be avoided. Suppose, for example, that the chipset 300 asserts the signal on the stop clock pin. One thread may immediately act on the stop clock signal by executing the stop clock micro-code. The other thread, however, may be involved with a higher priority interrupt, such as page fault handling. In this case,

the chipset 300 will be aware that the MT processor 200 has not entered the stop grant state because only one stop grant acknowledge SBC will have issued.

This embodiment of the present invention will also avoid confusion when there are several MT processors working together. For example, consider FIG. 5 which is a block diagram of a computer system having two MT processors 200, 202 according to an embodiment of the present invention. The first MT processor 200 has two threads 210, 220 and the second MT processor 202 has two threads 212, 222. The chipset 300 may determine which thread in which MT processor have performed the stop clock micro-code by evaluating the identifier included in a stop grant SBC on the bus 250.

Because the MT processor 200 sends out a stop grant acknowledge SBC for the first thread 210, 220 that reaches that code, even if other threads have not performed the associated logic, the micro-code may be created without being "thread aware." That is, the micro-code is simplified because it may execute the similar commands regardless of the thread 210, 220 on which it is executing.

Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of this embodiment of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention. For example, although a MT processor with two logical processors was used to illustrate an embodiment of the present invention, it will be appreciated that MT processors having another number of logical processors could also fall within the scope of the invention. Moreover, although software or hardware are described to control certain functions, such functions may be performed using either software, hardware or a combination of software and hardware, as is well known in the art.

* C L M S *

What is claimed is:

1. A method of entering a power mode in a processor, comprising:

issuing a first indication, including a first identifier associated with a first logical processor in the processor, that the first logical processor has entered the power mode; and

issuing a second indication, including a second identifier associated with a second logical processor in the processor, that the second logical processor has entered the power mode.

2. The method of claim 1, wherein the power mode is a stop grant mode.

3. The method of claim 2 wherein issuing the first indication comprises issuing the first indication as a first special bus cycle, and wherein issuing the second indication comprises issuing the second indication as a second special bus cycle.

4. The method of claim 3 wherein the first and the second special bus cycles are stop grant special bus cycles.

5. The method of claim 1 wherein the first identifier is a first processor identification signal associated with the first logical processor and the second identifier comprises a second processor identification signal associated with the second logical processor.

6. A method of transitioning from an active mode to a low power mode in a processor including a first logical processor associated with a first identifier and a second logical processor associated with a second identifier, comprising:

receiving an instruction to enter the low power mode;

issuing a first indication, including the first identifier, that the first logical processor has entered the low power mode; and

issuing a second indication, including the second identifier, that the second

This Sheet Has Been Replaced

logical processor has entered the low power mode.

7. The method of claim 6, wherein the low power mode comprises a stop grant mode, and further comprising:

receiving an instruction to enter a sleep mode after both the first and the second indications have been issued.

8. The method of claim 7 wherein issuing the first indication comprises issuing the first indication as a first special bus cycle, and wherein issuing the second indication comprises issuing the second indication as a second special bus cycle.

9. The method of claim 8 wherein the first and second special bus cycles comprise stop grant special bus cycles.

10. The method of claim 7 wherein the first identifier comprises a first processor identification signal associated with the first logical processor and the second identifier comprises a second processor identification signal associated with the second logical processor.

11. A method of monitoring the power mode of a processor having a first logical processor associated with a first identifier and a second logical processor associated with a second identifier, comprising:

receiving a first indication, including the first identifier, that the first logical processor has entered a power mode; and

receiving a second indication, including the second identifier, that the second logical processor has entered the power mode.

12. The method of claim 11, further comprising:

instructing the processor to enter a second power mode when the both the first and the second indications have been received.

This Sheet Has Been Replaced

13. A method of monitoring the power modes of a first processor having a first logical processor associated with a first identifier and a second logical processor associated with a second identifier and a second processor having a third logical processor associated with a third identifier and a fourth logical processor associated with a fourth identifier, comprising:

receiving an indication that the first processor has entered a power mode; comprising:

receiving a first indication, including the first identifier, that the first logical processor has entered the power mode, and

receiving a second indication, including the second identifier, that the second logical processor has entered the power mode; and

receiving an indication that the second processor has entered the power mode comprising:

receiving a third indication, including the third identifier, that the third logical processor has entered a power mode, and

receiving a fourth indication, including the fourth identifier, that the fourth logical processor has entered the power mode.

14. A method of transitioning a processor having an active mode, a first low power mode and a second low power mode, the processor also having a first logical processor associated with a first identifier and a second logical processor associated with a second identifier, comprising:

receiving a first indication, including the first identifier, that the first logical processor has entered the first low power mode;

receiving a second indication, including the second identifier, that the second logical processor has entered the first low power mode; and

transitioning the processor to the second low power mode when both the first and second indications have been received.

15. The method of claim 14 wherein the first low power mode is a stop grant mode, the second low power mode is a sleep mode, and receiving the first indication

This Sheet Has Been Replaced

comprises receiving the first indication as a first special bus cycle, and wherein receiving the second indication comprises receiving the second indication as a second special bus cycle.

16. A processor, comprising:

a first logical processor configured to issue a first indication, including a first identifier associated with said first logical processor, that said first logical processor has entered a power mode; and

a second logical processor configured to issue a second indication, including a second identifier associated with said second logical processor, that said second logical processor has entered the power mode.

17. The processor of claim 16, wherein the processor further comprising a number of internal clocks and the power mode is a mode in which some of said internal clocks are stopped.

18. The processor of claim 17, wherein the first and the second indications comprise stop grant acknowledge special bus cycles.

19. A computer system, comprising:

a processor configured to issue a first indication, including a first identifier associated with a first logical processor, that the first logical processor has entered a first power mode and further configured to issue a second indication, including a second identifier associated with a second logical processor, that the second logical processor has entered the first power mode; and

a processor controller coupled to said processor and configured to instruct said processor to enter a second power mode when the first and the second indications have been received from said processor.

20. The computer system of claim 19, wherein the first power mode is a low power mode and the second power mode is a mode using less power than the low

This Sheet Has Been Replaced

* C L M S *

power mode.

21. The computer system of claim 20, wherein the first and the second indications are stop grant acknowledge special bus cycles.

22. An article of manufacture comprising a computer readable medium having stored thereon instructions which, when executed by a processor, cause the processor to perform steps to enter a power mode, comprising:

issuing a first indication, including a first identifier associated with a first logical processor in the processor, that the first logical processor has entered a power mode; and

issuing a second indication, including a second identifier associated with a second logical processor in the processor, that the second logical processor has entered the power mode.

23. The article of manufacture of claim 22, wherein the power mode is a stop grant mode.

24. The article of manufacture of claim 22, wherein the first and the second indications comprise stop grant acknowledge special bus cycles.

This Sheet Has Been Replaced