US 20090217070A1

(19) **United States**

(12) **Patent Application Publication**

Kurts et al.

(10) **Pub. No.: US 2009/0217070 A1**

(43) **Pub. Date: Aug. 27, 2009**

(54) **DYNAMIC BUS PARKING**

(75) Inventors: **Tsvika Kurts**, Haifa (IL); **Efraim Rotem**, Haifa (IL)

Correspondence Address:
INTEL/BSTZ
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040 (US)

(73) Assignee: **INTEL CORPORATION**

(21) Appl. No.: **12/435,784**

(22) Filed: **May 5, 2009**

Related U.S. Application Data

(63) Continuation of application No. 11/172,110, filed on Jun. 30, 2005, now Pat. No. 7,529,955.

Publication Classification

(51) **Int. Cl.**
G06F 1/32 (2006.01)

(52) **U.S. Cl.** **713/322**; 713/323; 713/320; 711/104

(57) **ABSTRACT**

Systems and methods of power management provide for issuing a power saving message from a processor toward a controller and using the controller to conduct a power saving activity in response to the power saving message. In one embodiment, the power saving message is issued by de-asserting a bus arbitration signal and the power saving activity can include disabling one or more input buffers of the controller.

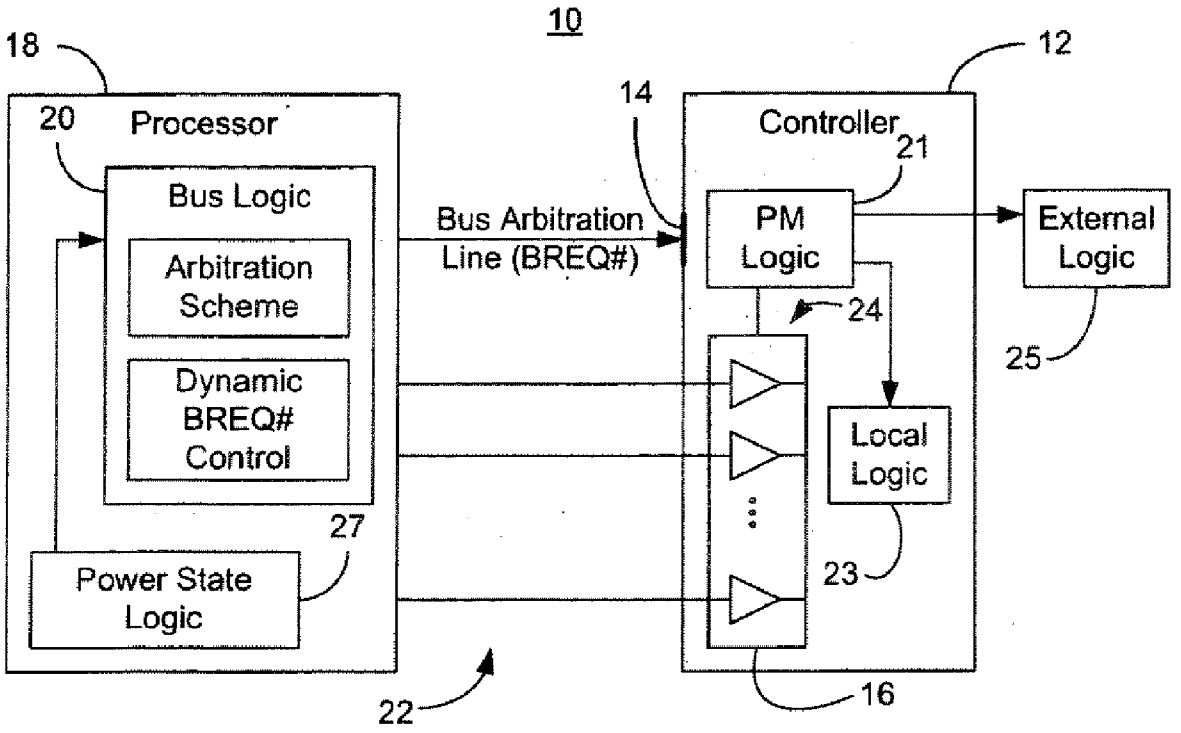
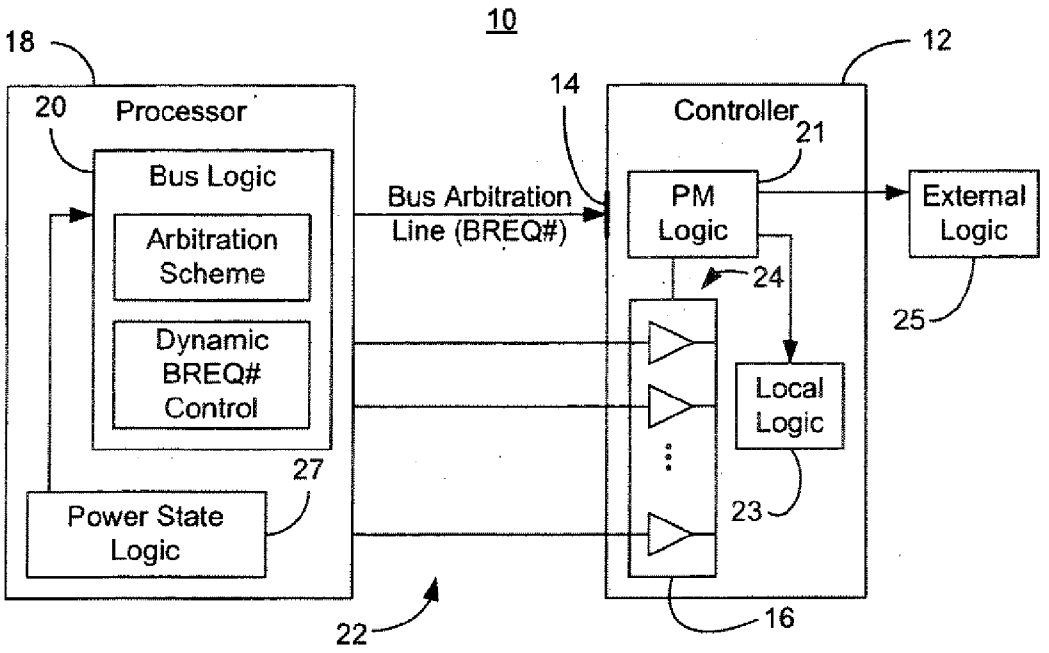


FIG. 1


10
18
20
Processor
Bus Logic
Arbitration Scheme
Dynamic BREQ# Control
27
Power State Logic
22
14
Bus Arbitration Line (BREQ#)
12
Controller
21
PM Logic
External Logic
25
24
Local Logic
23
16

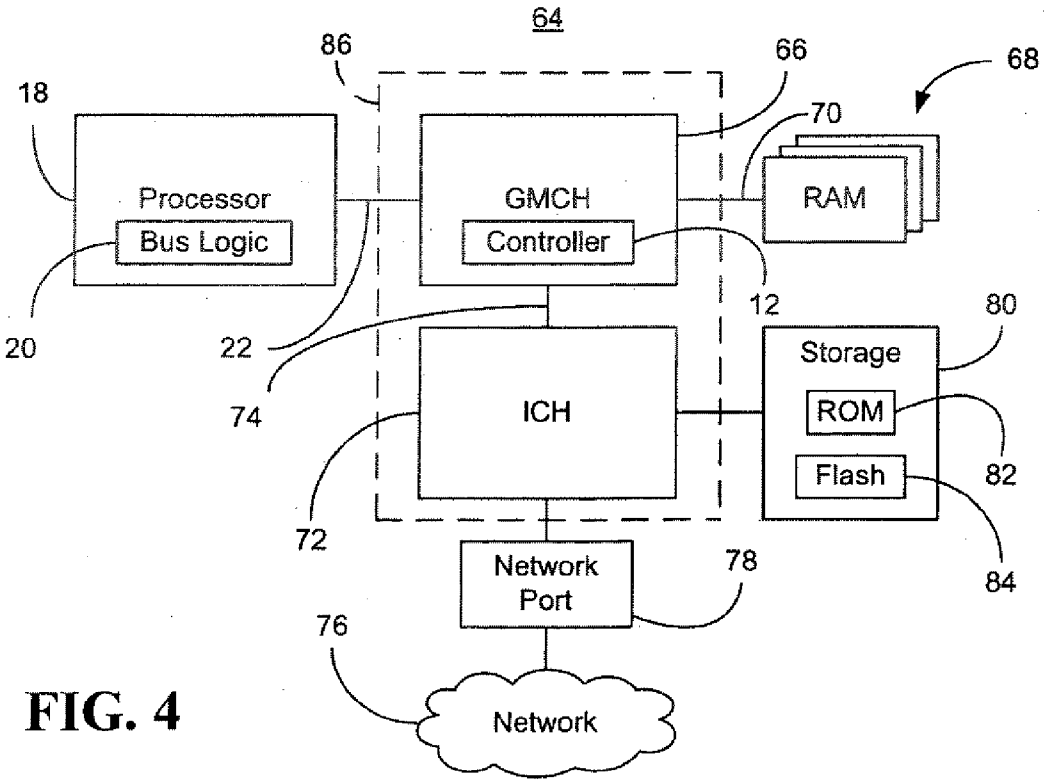

64
86
66
68
18
70
Processor
Bus Logic
GMCH
Controller
RAM
20
22
12
80
Storage
74
ICH
ROM
Flash
82
72
78
84
Network Port
76
Network

FIG. 4

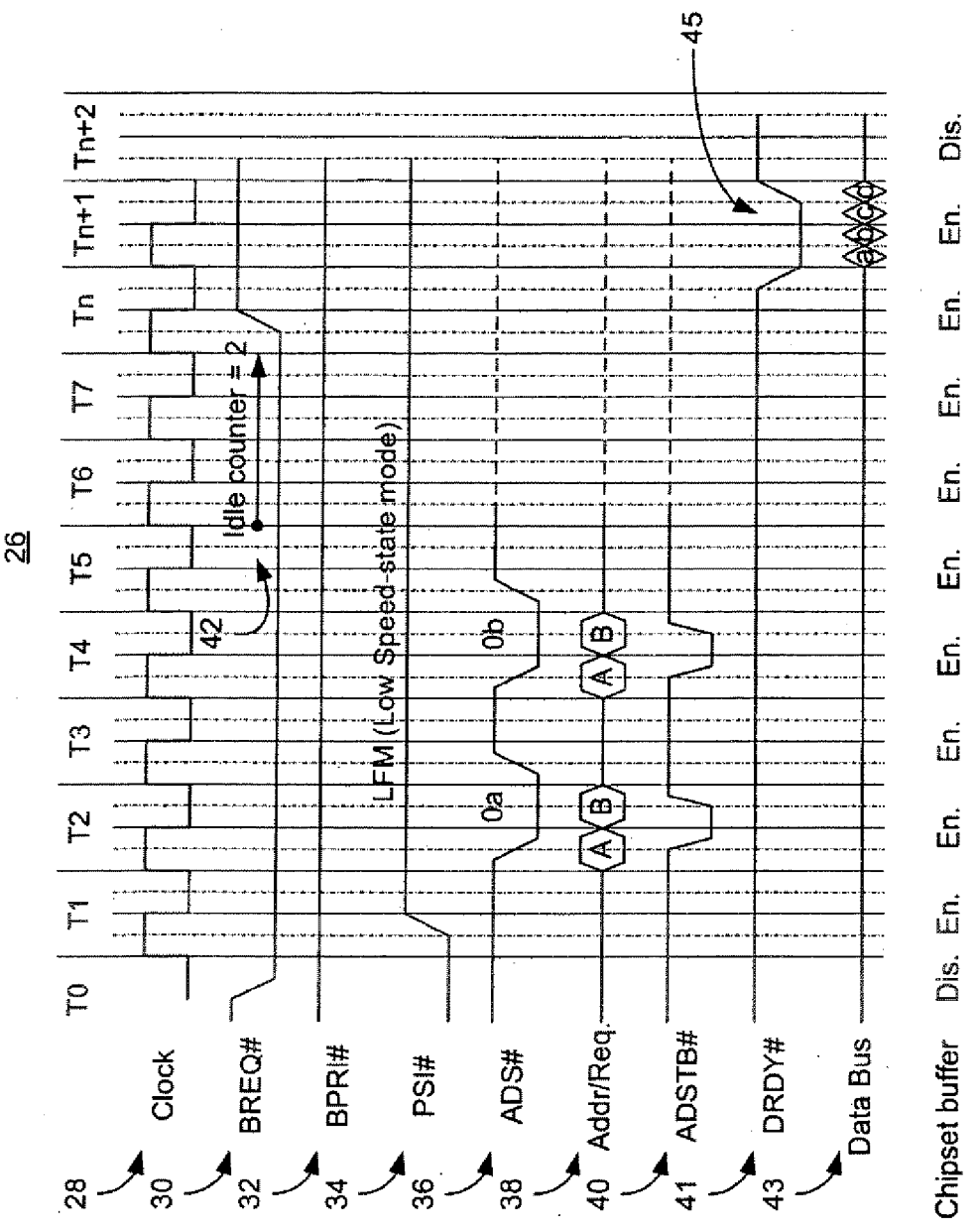

26
28
30 Clock
32 BREQ#
34 BPRi#
36 PSI#
38 ADS#
40 Addr/Req.
41 ADSTB#
43 DRDY#
Data Bus
Chipset buffer
T0
T1
T2
T3
T4
T5
T6
T7
Tn
Tn+1
Tn+2
Dis.
En.
42
Idle counter = 2
LFM (Low Speed-state mode)
0a
0b
A
B
45
a
b
c
d

FIG. 2A

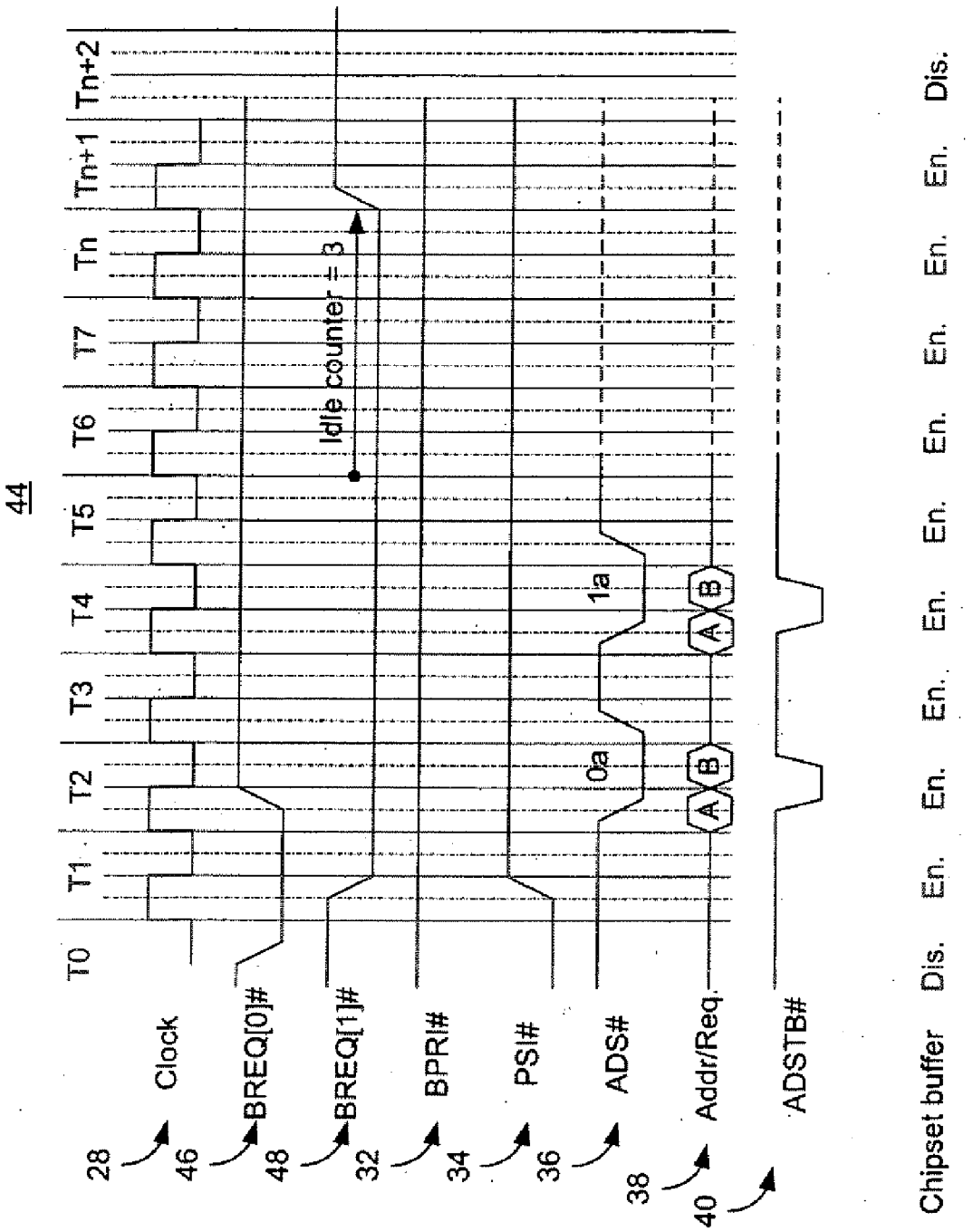

44
T0
T1
T2
T3
T4
T5
T6
T7
Tn
Tn+1
Tn+2
Idle counter = 3
Clock
BREQ[0]#
BREQ[1]#
BPRI#
PSI#
ADS#
Addr/Req.
ADSTB#
28
46
48
32
34
36
38
40
0a
1a
A
B
Chipset buffer
Dis.
En.
En.
En.
En.
En.
En.
En.
Dis.

FIG. 2B

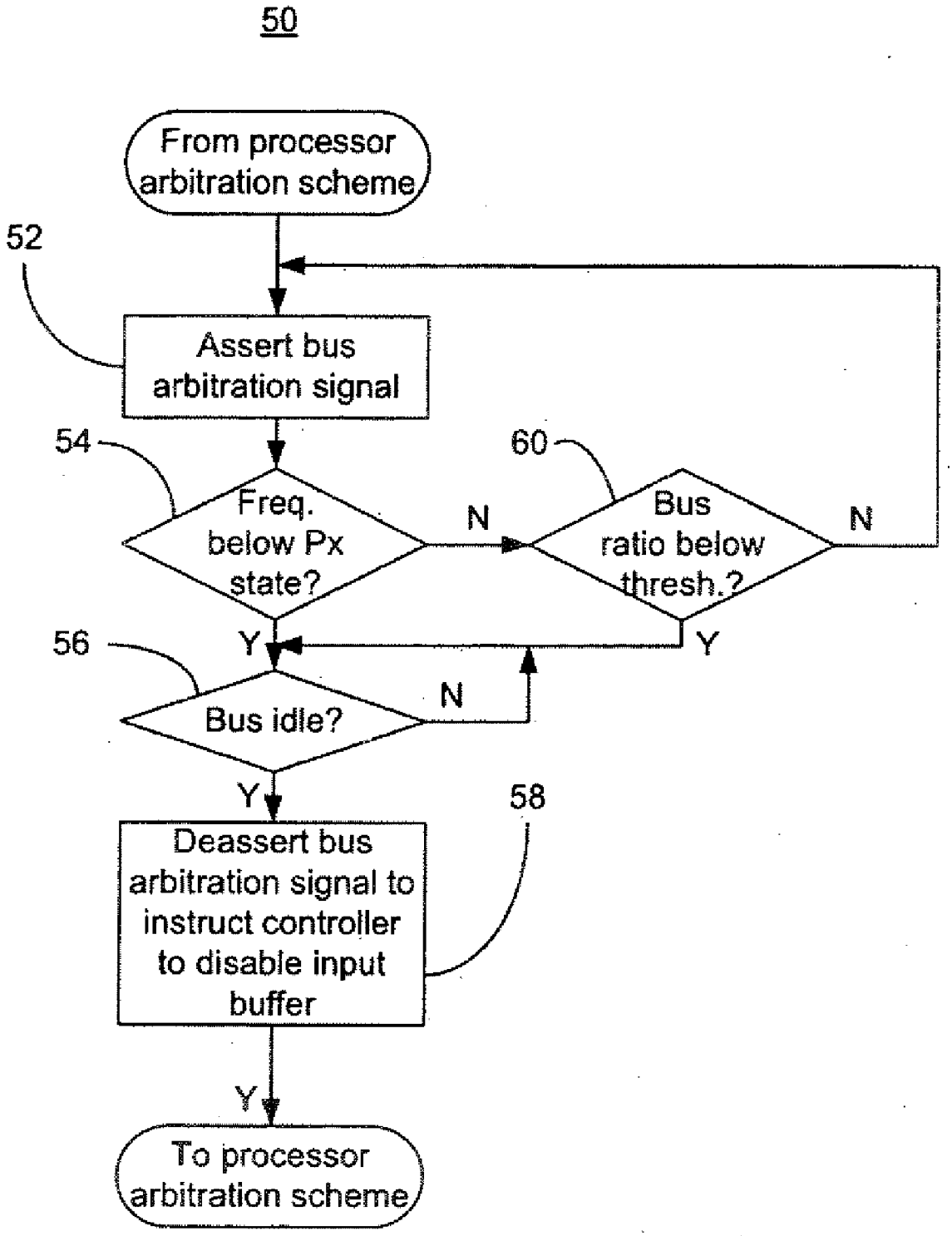

50
From processor arbitration scheme
52
Assert bus arbitration signal
54
Freq. below Px state?
N
60
Bus ratio below thresh.?
N
Y
Y
56
Bus idle?
N
Y
58
Deassert bus arbitration signal to instruct controller to disable input buffer
Y
To processor arbitration scheme

FIG. 3

DYNAMIC BUS PARKING

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] The present application is related to U.S. patent application Ser. No. 10/317,798, filed on Dec. 11, 2002; U.S. patent application Ser. No. 10/317,776, filed on Dec. 11, 2002; U.S. patent application Ser. No. 10/931,565, filed on Aug. 31, 2004; U.S. patent application Ser. No. 10/442,595, filed on May 21, 2003; and U.S. Pat. No. 6,842,035, issued on Jan. 11, 2005.

BACKGROUND

[0002] 1. Technical Field

[0003] Certain embodiments of the present invention generally relate to power management. In particular, some embodiments relate to platform level power management.

[0004] 2. Discussion

[0005] As the components of modern day computing systems continue to grow in functionality and computing form factors continue to decrease in size, computer designers and manufacturers are often faced with significant challenges related to increased power consumption. As a result, a number of techniques have been developed to reduce power consumption. For example, some approaches involve powering down system components when their functionality is not needed. While these solutions can be suitable under certain circumstances, there remains considerable room for improvement.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The various advantages of the embodiments of the present invention will become apparent to one skilled in the art by reading the following specification and appended claims, and by referencing the following drawings, in which:

[0007] FIG. **1** is a block diagram of an example of an apparatus according to one embodiment;

[0008] FIGS. **2**A and **2**B are timing diagrams of examples of input buffer disabling schemes according to first and second embodiments, respectively;

[0009] FIG. **3** is a flowchart of an example of a method of power management according to one embodiment;

[0010] FIG. **4** is a block diagram of an example of a system according to one embodiment.

DETAILED DESCRIPTION

[0011] In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the embodiments of the present invention. It will be evident, however, to one skilled in the art that the embodiments of the present invention may be practiced without these specific details. In other instances, specific apparatus structures and methods have not been described so as not to obscure the embodiments of the present invention. The following description and drawings are illustrative of the embodiments of the invention and are not to be construed as limiting the embodiments of the invention.

[0012] Some portions of the detailed description, which follow, may be presented in terms of algorithms and symbolic representations of operations on data bits or binary digital signals within a computer memory. These algorithmic descriptions and representations may be the techniques used by those skilled in the data processing arts to convey the substance of their work to others skilled in the art. For example, certain logic described herein may be implemented using hardware techniques such as complementary metal oxide semiconductor (CMOS) technology or transistor-transistor logic (TTL), controller firmware, microcode, software techniques, and any combination thereof. The components described herein may also be incorporated into one or more integrated circuit (IC) packages (i.e., chips) which are fabricated on a die cut from a wafer.

[0013] Any use of the terms "first", "second", etc. does not necessarily infer a chronological relationship, and is used to facilitate discussion only. Unless specifically stated otherwise, as apparent from the following discussions, it is appreciated that throughout the specification, discussions utilizing terms such as "processing," "computing," "calculating," "determining," or the like, refer to the action and/or processes of a computer or computing system, or similar electronic computing device, that manipulate and/or transform data represented as physical, such as electronic, quantities within the computing system's registers and/or memories into other data similarly represented as physical quantities within the computing system's memories, registers or other such information storage, transmission or display devices. Furthermore, terms such as "coupled" or "coupling" can be used to refer to any connection, direct or indirect, between the components in question and can involve a mechanical, electrical, optical, electromagnetic, or other type of relationship.

[0014] As used herein, the term "transaction" designates a bus activity that is related to a single bus access request. A transaction may include several phases, where each phase is associated with a specific set of bus signals to communicate a particular type of information. For one embodiment, exemplary phases may include, for example, an arbitration phase, a request phase, a snoop phase, a response phase and/or a data phase.

[0015] In the request phase, the requesting agent can drive request control and address information on the bus. During a subsequent snoop phase, it may be determined, whether sought after data is stored locally and/or whether the transaction is likely to be completed in order with respect to previously issued transactions. In a response phase, a response agent can report to the requesting agent information indicating whether the requested transaction has succeeded or failed. If the requested transaction includes data transfer, a data phase, also referred to as a data transfer phase, may be initiated in response to the assertion of a data ready signal.

[0016] FIG. **1** shows an apparatus **10** having a controller **12** and a processor **18**. The processor **18** and the controller **12** may communicate with one another over a bus **22**, where in one example the bus **22** may be a pipelined front side bus (FSB) that includes address, data and control portions. The address, data and control portions may also be referred to as address, data and control buses. In another example, the bus **22** may be part of a point-to-point interconnection fabric that enables direct communication among multiple components of a computing system. The illustrated controller **12**, which may be part of a graphic model and memory controller hub (GMCH), commonly used on computing system chipsets, has a power saving message line input **14** and power management logic **21**. The controller **12** may also include one or more input buffers **16** coupled to the bus **22** and the power management logic **21**, local logic **23** coupled to the power management logic **21**, and external logic **25** coupled to the power management logic **21**.

[0017] The illustrated input buffer **16** has a plurality of sense amplifiers **24** that may consume a significant amount of power when enabled for receiving data. The illustrated local logic **23** may include internal components such as functional blocks for performing the various operations provided by the controller **12**. For example, the local logic **23** could conduct memory or input/output (**10**) management operations. The illustrated external logic **25** could include external components such as one or more memory device (e.g., system memory) functional blocks, where the external logic **25** may be managed by the controller **12**. The illustrated local logic **23** and external logic **25** may also consume a significant amount of power when enabled for operation.

[0018] The processor **18** can have power state logic **27** and bus logic **20**, which may also be referred to as a bus agent, capable of issuing a power saving message to the controller **12**. The illustrated power management logic **21** of the controller **12** is able to conduct a power saving activity in response to the power saving message. The power saving message may be conveyed in a number of different ways. For example, in the illustrated example, the processor uses a bus arbitration line to communicate the power saving message to the controller **12**.

[0019] In particular, the illustrated bus logic **20** asserts a bus arbitration signal toward the line input **14** whenever the processor needs to communicate with the controller **12** over the bus **22**. In the illustrated example, the bus arbitration signal is identified as the "BREQ#" signal. For the purposes of this discussion, a "#" at the end of a signal name indicates that the associated signal is an active low signal (i.e., considered to be asserted when it drives a logic low level on the external bus). It will be appreciated that active high signals may be used instead with corresponding changes in associated circuitry to provide similar functionality. Further, for one embodiment, one or more of the signals associated with the bus **22** are low voltage swing signals that have a voltage swing smaller than full swing. It can also be understood that the bus arbitration signal may be an encoded message rather than the physical signal BREQ#.

[0020] The bus logic **20** is also able to de-assert the bus arbitration signal in order to instruct the controller **12** to conduct the power saving activity. The power saving activity may involve disabling one or more internal components of the controller **12** and/or disabling one or more external components that are managed by the controller **12**. For example, the disabled internal components could include the sense amplifiers **24** of the input buffer **16**, the local logic **23**, and so on. Similarly, the disabled external components could include the external logic **25**, where conducting the power saving activity may involve placing a system dynamic random access memory (DRAM) into a power down state (e.g., self refresh) if the controller **12** is a memory controller. The power management logic **21** of the controller **12** can therefore detect the de-assertion of the bus arbitration signal and conduct various power saving activities in response to detecting the de-assertion.

[0021] In the case of the disabling of the input buffer **16**, the bus **22** can be said to be dynamically "parked" from the perspective of the controller **12**. Accordingly, by enabling the processor **18** to selectively conduct power saving activities such as disabling the input buffer **16** of the controller **12**, the bus logic **20** provides substantial power savings over conventional approaches. These power savings can lead to greater battery life in mobile platforms as well as reduced temperatures. In fact, extending the power features of the bus **22** to the controller **12** side of the bus **22**, represents a significant advantage over conventional power management architectures.

[0022] Turning now to FIG. **2**A, an example of the signaling for one power management scheme **26** is shown in greater detail. In the illustrated example, a bus clock signal **28** is shown in relation to a bus arbitration signal (BREQ#) **30**, a bus priority signal (BPRI#) **32**, a power status indicator signal (PSI#) **34**, an address strobe common clock signal (ADS#) **36**, an address/request signal (Addr/Req.) **38** and an address synchronous strobe signal (ADSTB#) **40**. A data ready signal (DRDY#) **41** and data bus signal **43** are also shown. In the illustrated example, the bus arbitration signal **30** is asserted by the processor bus logic during the T**0** clock cycle to gain bus ownership in order to communicate with the controller. The illustrated common clock and address strobe signals **36**, **40**, facilitate the transfer of information via the address/request signal **38** (i.e., "A" and "B") during the T**2**-T**4** clock cycles. In particular, the common clock signal **36** is used to indicate that a new transaction is to be issued to the bus and the asynchronous strobe signal **40** is used to sample the double pump address lines.

[0023] The illustrated bus arbitration signal **30** is de-asserted during the Tn clock cycle. The controller input buffers, which were disabled in the T**0** clock cycle, can be disabled upon completion of the last burst data transfer associated with the DRDY# assertion **45**. In particular, although de-assertion of the bus arbitration signal **30** indicates that the processor has completed an issue request, all data transfers related to this request should complete before the input buffers are disabled. Thus, the internal buffers of the controller can become disabled in the illustrated Tn+2 clock cycle. The result can be a substantial reduction in power consumption within the controller.

[0024] It should be noted each time the bus arbitration signal **30** is re-asserted, there may be a minimal impact on performance because a re-arbitration may be necessary before the next transaction can take place on the bus. For example, re-arbitration could impose two clocks of additional latency (e.g., T**0** and T**1** clock cycles in the example shown) on the system. Frequent transitions into and out of the processor-initiated power saving state could therefore cause an undesirable minimum reduction in performance. Accordingly, an idle counter **42** can be used to ensure that the bus has been idle for a sufficient amount of time (i.e., T**6** and T**7** clock cycles in the example shown) before de-asserting the bus arbitration signal **30**. It should also be noted that the idle counter value can programmed to any value (i.e., zero and up) depending upon the circumstances. In this regard, the processor bus logic may need to retrieve the counter value from a programmable memory location such as the basic input/output system (BIOS) erasable programmable read only memory (EPROM) and compare the number of idle clock cycles to the counter value in determining whether to de-assert the bus arbitration signal **30**.

[0025] The decision whether to de-assert the bus arbitration signal **30** can also be made based on the operating frequency of the processor. In this regard, the processor could be capable of selectively reducing its frequency/voltage during periods of relative inactivity or to otherwise save power. Thus, de-assertion of the bus arbitration signal **30** could be limited to time periods when the operating frequency of the processor is below some frequency threshold.

[0026] In one example, a processor can enter a low frequency mode (LFM) in order to achieve greater battery life. If the processor is able to enter such a low frequency mode, it may be desirable to limit the use of the bus arbitration de-assertion to the LFM periods in order to "hide" the potential performance penalty associated with the de-assertion. Simply put, the de-assertion penalty may be negligible in comparison to the performance penalty associated with the LFM. Thus, in the illustrated example, the processor bus logic ensures that the LFM signal **34** is asserted before proceeding with the de-assertion of the bus arbitration signal **30**. The frequency at which the power saving activity is activated may be programmable. For example, it is possible to store the value 1.3 GHz, where each time the processor enters a frequency equal to or lower than 1.3 GHz, the power saving message will be asserted. Alternatively, the power saving message could be asserted each-time the performance state (e.g., P state) of the processor falls below a predetermined level. The same applies to a plurality of cores/processors where the combined state is evaluated.

[0027] Alternatively, the state of the LFM signal **34** could be ignored altogether when deciding whether to de-assert the bus arbitration signal **30**. One circumstance in which such an approach may be acceptable could be in the case of a very small computing platform (e.g., handheld personal digital assistant) where battery life may always take priority over performance.

[0028] Although not shown, the processor bus logic could further base the decision whether to de-assert the bus arbitration signal **30** on the bus ratio (i.e., the ratio of the processor operating frequency to the bus operating frequency). For example, the processor bus logic could confirm that the bus ratio is below a particular threshold to ensure that the de-assertion related performance loss is not visible. Such a technique could be employed in addition to or instead of the raw frequency check provided by LFM signal **34**. If the LFM signal **34** is not used, the bus ratio approach could facilitate acceptable operation even when the processor is in the high frequency mode.

[0029] It should also be noted that the bus power signal **32**, which may travel from the controller to the processor, can also be de-asserted by the controller in order to effectively disable the input buffers of the processor and achieve even greater power savings.

[0030] FIG. **2**B shows an example of a power management scheme **44** in which the controller supports more than one processor. In this example, a first processor is associated with a first bus arbitration signal (BREQ[0]#) **46** and a second processor is associated with a second bus arbitration signal (BREQ[1]#) **48**. The two bus arbitration signals **46**, **48** can be ORed together so that the input buffer of the controller is enabled whenever either of the processors is asserting the signal. Thus, in the illustrated example, the controller input buffer is enabled until the Tn+2 clock cycle, where both bus arbitration signals **46**, **48** are de-asserted. A multiprocessor bus arbitration protocol can be used to manage the assertion/de-assertion of the signals **46**, **48**. It can also be seen that in the illustrated example, a counter value of three clock cycles is used so that the last processor to attempt to de-assert its respective bus arbitration signal (i.e., the second processor in the example shown) is forced to wait for the T**6** through Tn clock cycles to complete before disabling the controller input buffer via bus arbitration signal **48**.

[0031] Turning now to FIG. **3**, a method **50** of power management is shown. The method **50** may be implemented as processor bus logic using hardware techniques such as complementary metal oxide semiconductor (CMOS) technology or transistor-transistor logic (TTL), controller firmware, microcode, software techniques, and any combination thereof. In the illustrated example, processing block **52** provides for asserting a bus arbitration signal from a processor toward a controller having a bus arbitration line input and an input buffer. If it is determined at block **54** that the operating frequency of the processor is at or below a frequency threshold (e.g., at or below a particular Advanced Configuration and Power Interface/ACPI Specification, Rev. 3.0, Sep. **2**, **2004**, Px state; in low frequency mode, etc.), block **56** provides for determining whether a bus disposed between the processor and the controller has been idle for a certain amount of time. As already discussed, the determination at block **56** may involve comparing the number of idle clock cycles to a counter value, which may be retrieved from a programmable memory location. If the bus is sufficiently idle, the bus arbitration signal can be de-asserted at block **58** in order to instruct the controller to disable its input buffer.

[0032] If it is determined at block **54** that the operating frequency of the processor is not at or below the frequency threshold, block **60** provides for determining whether the bus ratio is below a certain threshold. For example, it could be determined that if the bus ratio is less than 2:1, the relative processor performance is such that the additional performance loss associated with the de-assertion may be negligible. Thus, if block **60** determines that the bus ratio is below the relevant threshold, a check of the bus idleness can still be conducted at block **56**. Otherwise, the process returns to block **52** for re-assertion of the bus arbitration signal.

[0033] FIG. **4** shows a computing system **64**, which could be part of a server, desktop personal computer (PC), notebook PC, personal digital assistant (PDA), wireless "smart" phone, and so on. The illustrated system **64** has a microprocessor **18**, which may have one or more processor cores (not shown), where each core may be fully functional with instruction fetch units, instruction decoders, level one (L1) cache, execution units, and so on. The microprocessor **18** may also include bus logic **20** as already discussed. The microprocessor **18** can communicate with a graphic model and memory controller hub (GMCH) **66**, also known as a Northbridge, via a front side bus **22**. As already noted, the front side bus **22** could alternatively be replaced by a point-to-point fabric that interconnects each of the components in the system **64**. The GMCH **66**, which may include a controller **12**, can communicate with system random access memory (RAM) **68** via a memory bus **70**. In one embodiment, the RAM **68** includes dynamic random access memory (DRAM) modules. The modules of the RAM **68** may be incorporated in to a single inline memory module (SIMM), dual inline memory module (DIMM), small outline DIMM (SODIMM), and so on. The GMCH **66** may also include one or more graphics devices (not shown).

[0034] The illustrated GMCH **66** communicates with an I/O controller hub (ICH) **72**, also known as a Southbridge, via a hub bus **74**. The GMCH **66** and the ICH **72** are sometimes referred to as a chipset **86**. The microprocessor **18** may also be operatively connected to a network **76** via a network port **78** through the ICH **72**. The ICH **72** may also be coupled to storage **80**, which may include a read only memory (ROM) **122**, flash memory **84**, etc. In one embodiment, the bus logic

20 is configured to selectively assert a bus arbitration signal toward the controller **12** in order to reduce the power consumption of the chipset **86**.

[0035] As already noted, various embodiments of the disclosed subject matter may be implemented in hardware, firmware, software, or combination thereof, and may be described by reference to or in conjunction with program code, such as instructions, functions, procedures, data structures, logic, application programs, design representations or formats for simulation, emulation, and fabrication of a design, which when accessed by a machine results in the machine performing tasks, defining abstract data types or low-level hardware contexts, or producing a result.

[0036] For simulations, program code may represent hardware using a hardware description language or another functional description language which essentially provides a model of how designed hardware is expected to perform. Program code may be assembly or machine language, or data that may be compiled and/or interpreted. Furthermore, it is common in the art to speak of software, in one form or another as taking an action or causing a result. Such expressions are merely a shorthand way of stating execution of program code by a processing system which causes a processor to perform an action or produce a result.

[0037] Program code may be stored in, for example, volatile and/or non-volatile memory, such as storage devices and/or an associated machine readable or machine accessible medium including solid-state memory, hard-drives, floppy-disks, optical storage, tapes, flash memory, memory sticks, digital video disks, digital versatile discs (DVDs), etc., as well as more exotic mediums such as machine-accessible biological state preserving storage. A machine readable medium may include any mechanism for storing, transmitting, or receiving information in a form readable by a machine, and the medium may include a tangible medium through which electrical, optical, acoustical or other form of propagated signals or carrier wave encoding the program code may pass, such as antennas, optical fibers, communications interfaces, etc. Program code may be transmitted in the form of packets, serial data, parallel data, propagated signals, etc., and may be used in a compressed or encrypted format.

[0038] Program code may be implemented in programs executing on programmable machines such as mobile or stationary computers, personal digital assistants, set top boxes, cellular telephones and pagers, and other electronic devices, each including a processor, volatile and/or non-volatile memory readable by the processor, at least one input device and/or one or more output devices. Program code may be applied to the data entered using the input device to perform the described embodiments and to generate output information. The output information may be applied to one or more output devices. One of ordinary skill in the art may appreciate that embodiments of the disclosed subject matter can be practiced with various computer system configurations, including multiprocessor or multiple-core processor systems, minicomputers, mainframe computers, as well as pervasive or miniature computers or processors that may be embedded into virtually any device. Embodiments of the disclosed subject matter can also be practiced in distributed computing environments where tasks may be performed by remote processing devices that are linked through a communications network.

[0039] Although operations may be described as a sequential process, some of the operations may in fact be performed in parallel, concurrently, and/or in a distributed environment, and with program code stored locally and/or remotely for access by single or multi-processor machines. In addition, in some embodiments the order of operations may be rearranged without departing from the spirit of the disclosed subject matter. Program code may be used by or in conjunction with embedded controllers.

[0040] Those skilled in the at can appreciate from the foregoing description that the broad techniques of the embodiments of the present invention can be implemented in a variety of forms. Therefore, while the embodiments of this invention have been described in connection with particular examples thereof, the true scope of the embodiments of the invention should not be so limited since other modifications will become apparent to the skilled practitioner upon a study of the drawings, specification, and following claims.

1. An apparatus comprising:
a controller; and
a processor having bus logic to issue a power saving message, the controller having power management logic to conduct a power saving activity in response to the power saving message.

2. The apparatus of claim **1**, wherein the bus logic is to determine whether to issue the power saving message based on an operating frequency of the processor.

3. The apparatus of claim **2**, wherein the bus logic is to determine whether to issue the power saving message based on a ratio of the operating frequency of the processor to an operating frequency of a bus disposed between the processor and the controller.

4. The apparatus of claim **2**, further including a plurality of processors, the operating frequency to be shared by each of the plurality of processors.

5. The apparatus of claim **1**, wherein the bus logic is to determine whether to issue the power saving message based on an idleness of a bus disposed between the processor and the controller.

6. The apparatus of claim **5**, wherein the bus logic is to count a number of idle clock cycles on the bus and compare the number of idle clock cycles to a counter value.

7. The apparatus of claim **6**, wherein the bus logic is to retrieve the counter value from a programmable memory location.

8. The apparatus of claim **1**, wherein the power saving message is to include a de-assertion of a bus arbitration signal.

9. The apparatus of claim **1**, wherein the power saving activity is to be selected from a group comprising disabling an internal component of the controller and disabling an external component that is managed by the controller.

10. The apparatus of claim **9**, wherein the internal component of the controller is to be selected from a group comprising an input buffer and a functional block.

11. The apparatus of claim **9**, wherein the external component is to include a memory device.

12. The apparatus of claim **1**, further including a chipset memory controller hub, the memory controller hub including the controller.

13. A method comprising:
issuing a power saving message from a processor toward a controller; and
using the controller to conduct a power saving activity in response to the power saving message.

14. The method of claim **13**, further including determining whether to issue the power saving message based on an operating frequency of the processor.

15. The method of claim **14**, further including determining whether to issue the power saving message based on a ratio of the operating frequency of the processor to an operating frequency of a bus disposed between the processor and the controller.

16. The method of claim **14**, wherein the operating frequency is shared by each of a plurality of processors.

17. The method of claim **13**, further including determining whether to issue the power saving message based on an idleness of a bus disposed between the processor and the controller.

18. The method of claim **17**, further including: counting a number of idle clock cycles on the bus; and comparing the number of idle clock cycles to a counter value.

19. The method of claim **18**, further including retrieving the counter value from a programmable memory location.

20. The method of claim **13**, wherein the issuing includes de-asserting a bus arbitration signal.

21. The method of claim **13**, wherein the using is selected from a group comprising disabling an internal component of the controller and disabling an external component that is managed by the controller.

22. A system comprising:

a small outline dual inline memory module (SODIMM);

a chipset coupled to the SODIMM; and

a processor coupled to the chipset, the processor having bus logic to issue a power saving message, the chipset having power management logic to conduct a power saving activity in response to the power saving message.

23. The system of claim **22**, wherein the bus logic is to determine whether to issue the power saving message based on an operating frequency of the processor.

24. The system of claim **23**, wherein the bus logic is to determine whether to issue the power saving message based on a ratio of the operating frequency of the processor to an operating frequency of a bus disposed between the processor and the chipset.

25-30. (canceled)

* * * * *