US 20090172434A1

(19) **United States**

(12) **Patent Application Publication**

Kwa et al.

(10) **Pub. No.: US 2009/0172434 A1**

(43) **Pub. Date: Jul. 2, 2009**

(54) **LATENCY BASED PLATFORM COORDINATION**

(76) Inventors: **Seh W. Kwa**, San Jose, CA (US); **Robert Gough**, Cornelius, OR (US); **Neil Songer**, Santa Clara, CA (US); **Jaya L. Jeyaseelan**, Cupertino, CA (US); **Barnes Cooper**, Tigard, OR (US); **Nilesh V. Shah**, Folsom, CA (US)

Correspondence Address:
Caven & Aghevli LLC
c/o CPA Global
P.O. BOX 52050
MINNEAPOLIS, MN 55402 (US)

(21) Appl. No.: **12/006,251**

(22) Filed: **Dec. 31, 2007**

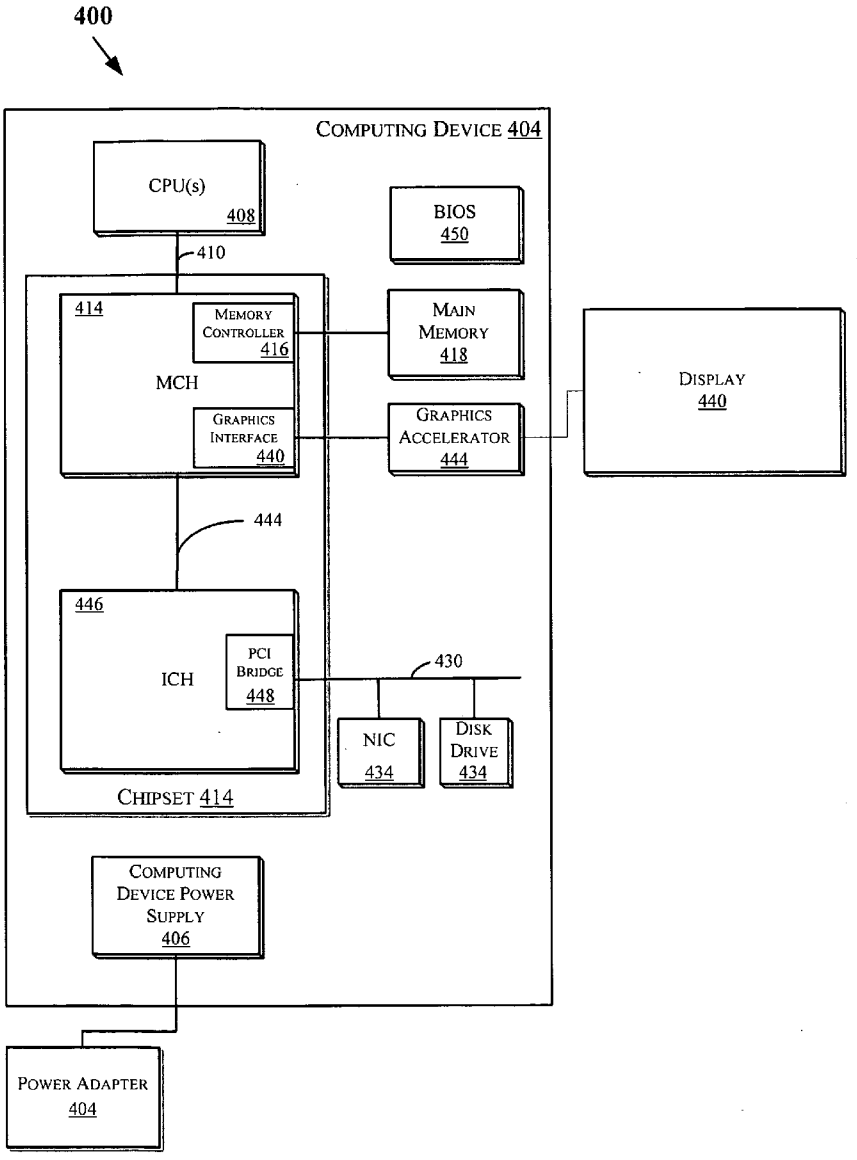
Publication Classification

(51) **Int. Cl.**
G06F 1/26 (2006.01)
G06F 1/32 (2006.01)

(52) **U.S. Cl.** **713/320**; 713/300

(57) **ABSTRACT**

In some embodiments, an electronic apparatus comprises at least one processor, a plurality of components, and a policy engine comprising logic to receive latency data from one or more components in the electronic device, compute a minimum latency tolerance value from the latency data, and determine a power management policy from the minimum latency tolerance value.



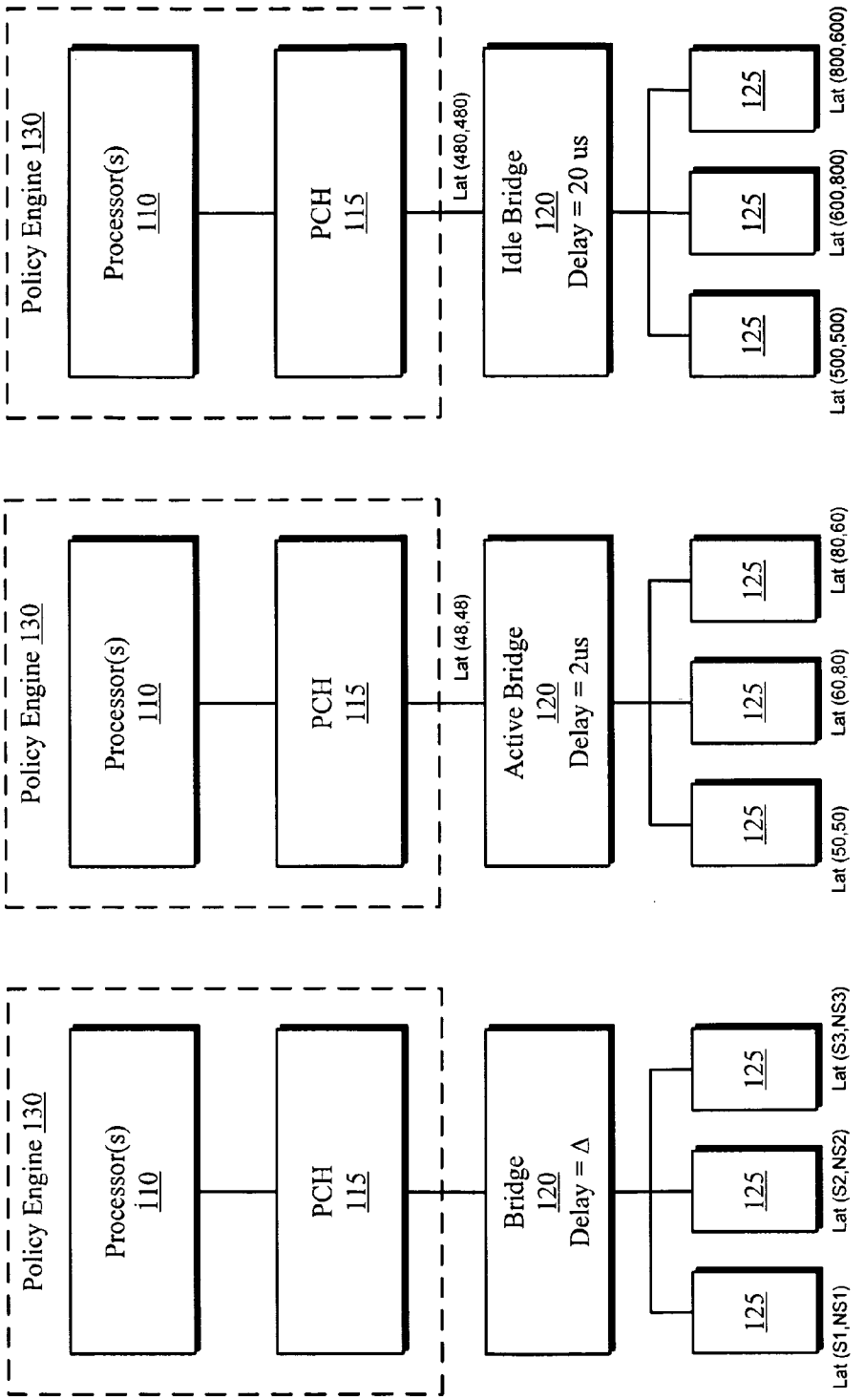


FIG. 1A

FIG. 1B

FIG. 1C

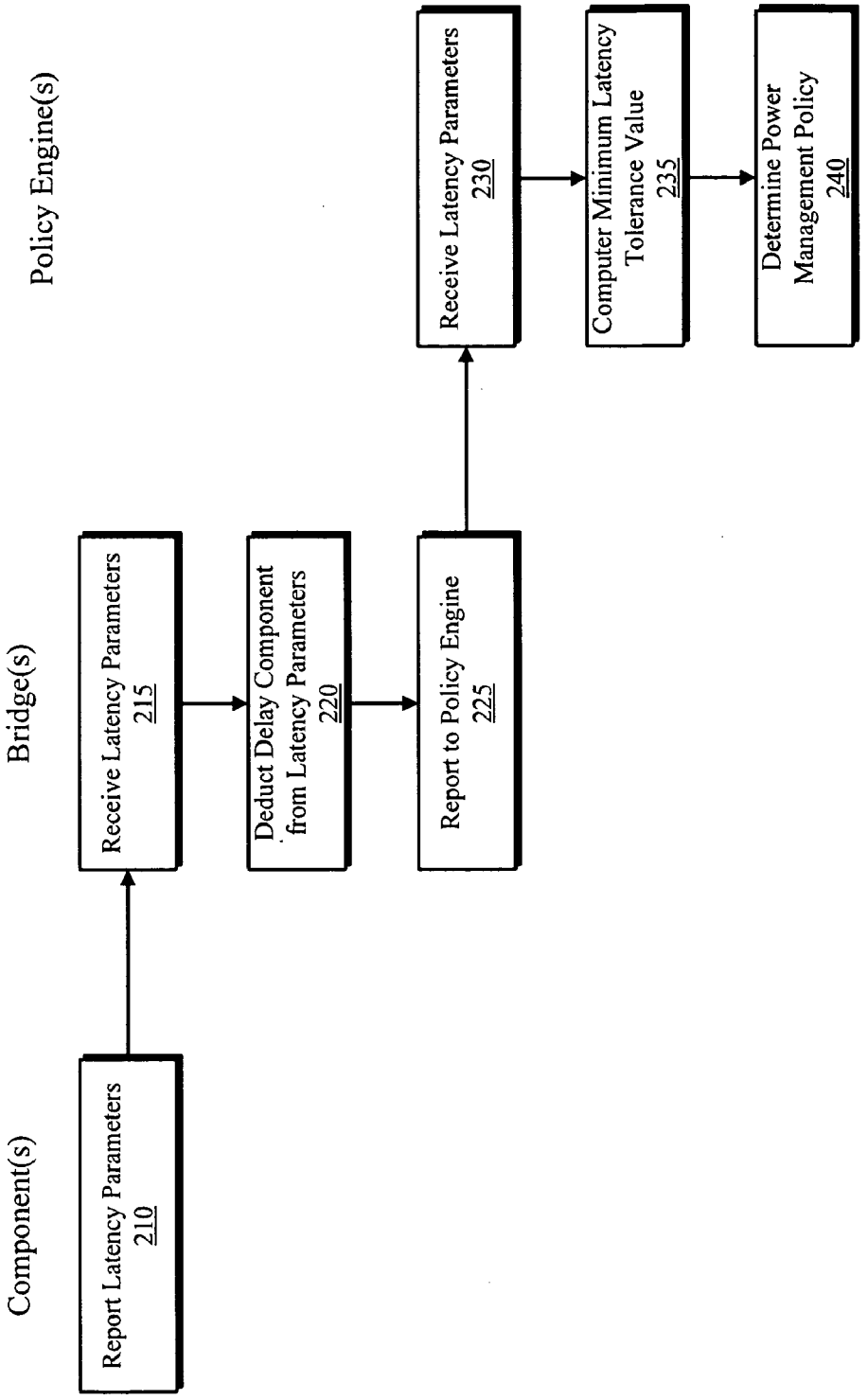

Component(s)
Report Latency Parameters 210
Bridge(s)
Receive Latency Parameters 215
Deduct Delay Component from Latency Parameters 220
Report to Policy Engine 225
Policy Engine(s)
Receive Latency Parameters 230
Computer Minimum Latency Tolerance Value 235
Determine Power Management Policy 240

FIG. 2

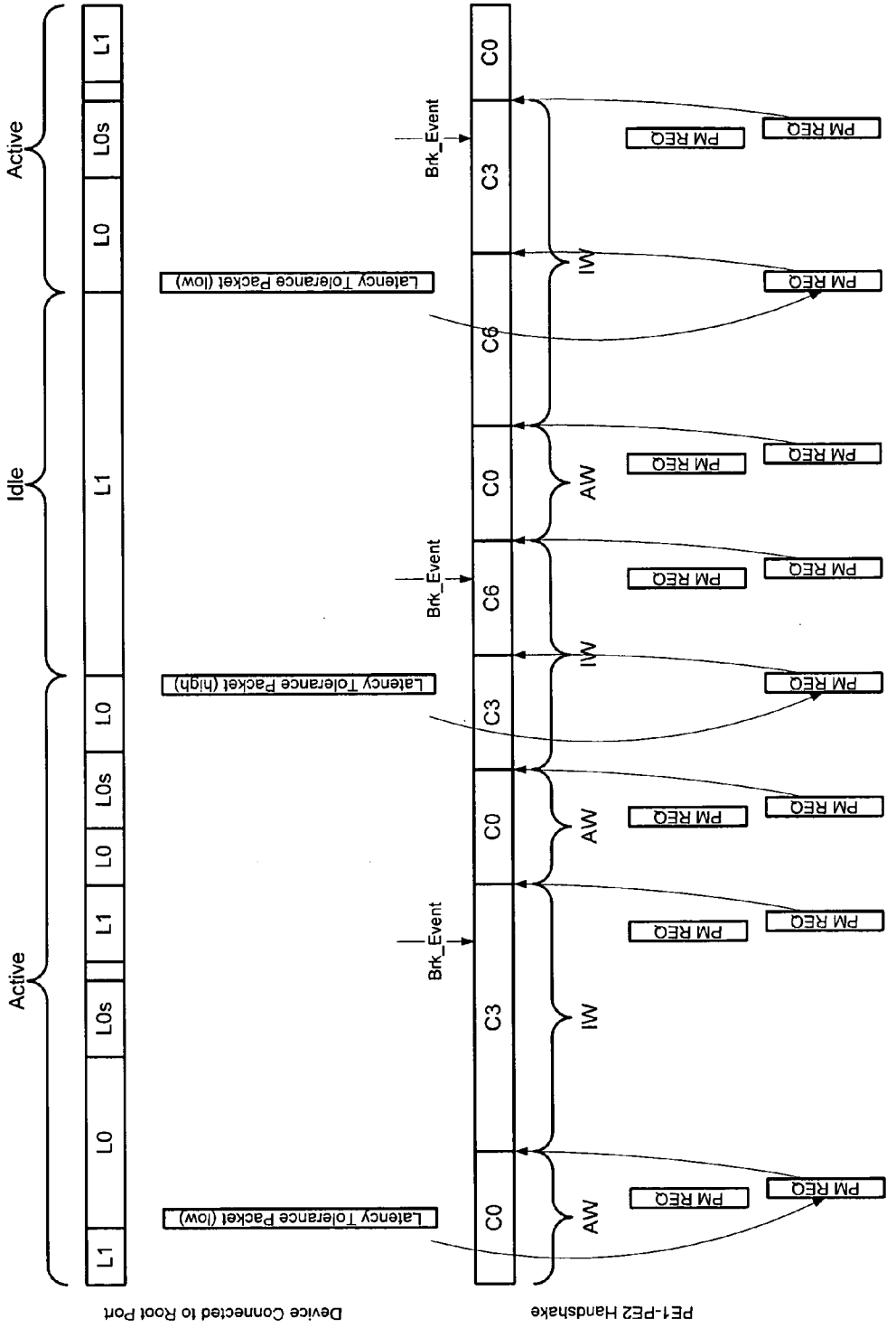

Active
Idle
Active
L1
L0
L0s
L1
L0
L0s
L1
L0
L0s
L1
L0
L0s
L1
Device Connected to Root Port
PE1-PE2 Handshake
C0
C3
C6
C0
C3
C6
C0
C3
C0
Brk_Event
Brk_Event
Brk_Event
Latency Tolerance Packet (low)
Latency Tolerance Packet (high)
Latency Tolerance Packet (low)
AW
IW
AW
IW
AW
IW
PM REQ
PM REQ
PM REQ
PM REQ
PM REQ
PM REQ
PM REQ
PM REQ
PM REQ
PM REQ

FIG. 3

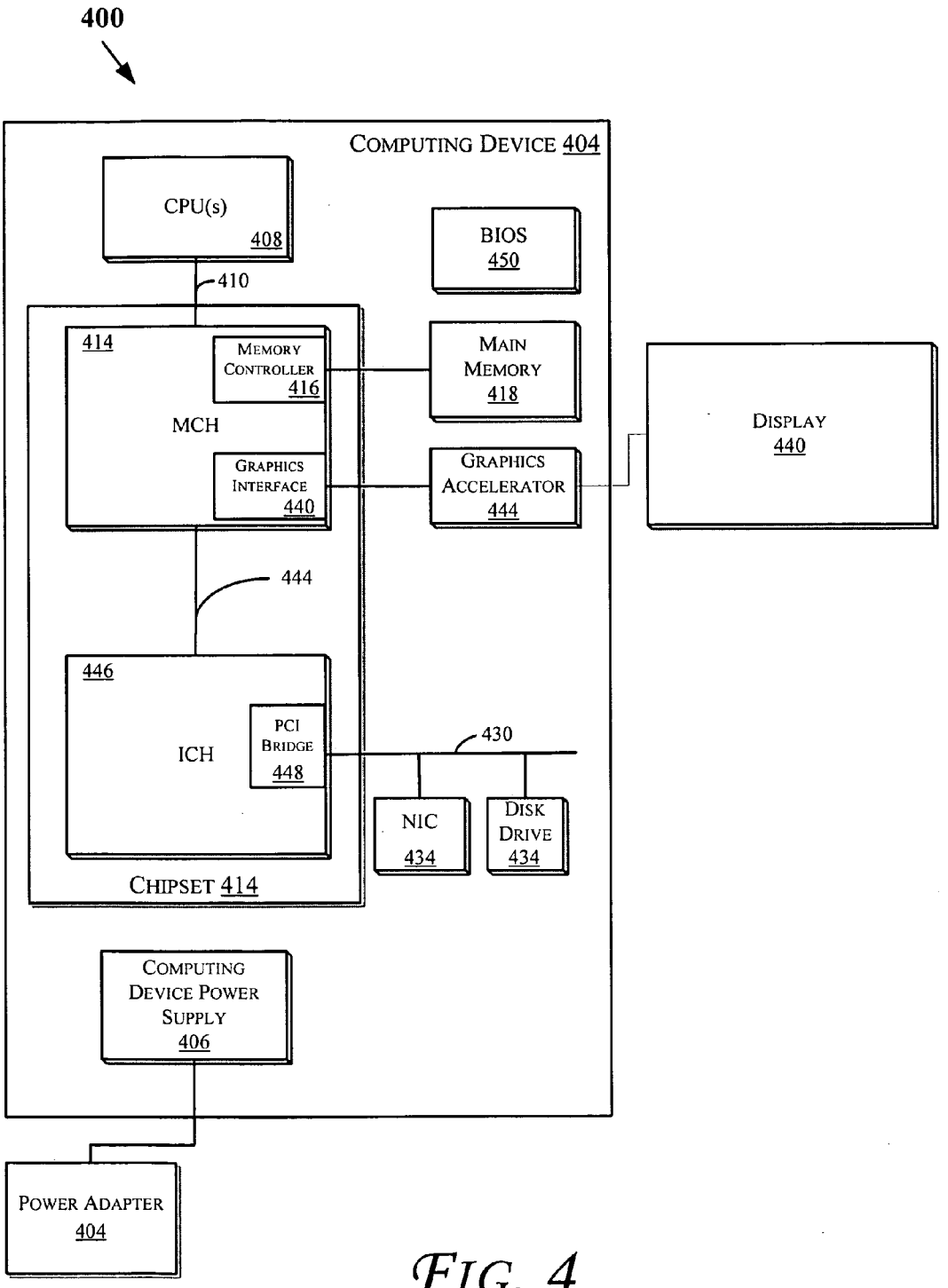

400
COMPUTING DEVICE 404
CPU(s)
408
410
BIOS
450
414
MEMORY CONTROLLER
416
MCH
MAIN MEMORY
418
GRAPHICS INTERFACE
440
GRAPHICS ACCELERATOR
444
DISPLAY
440
444
446
ICH
PCI BRIDGE
448
430
NIC
434
DISK DRIVE
434
CHIPSET 414
COMPUTING DEVICE POWER SUPPLY
406
POWER ADAPTER
404

FIG. 4

LATENCY BASED PLATFORM COORDINATION

RELATED APPLICATIONS

[0001] None.

BACKGROUND

[0002] Power management of the interconnected devices is becoming more of a concern as computers implement mobile system platforms where the computers and devices are battery powered. One of the biggest challenges of implementing an aggressive platform power management for mobile PC client and handheld devices is the lack of awareness of device latency tolerance to main memory accesses (DMA) and application latency dependency to facilitate power policy decisions. Deeper sleep states gain greater power savings, but at the cost of longer resume time. For example, deeper sleep states helps microprocessors achieve very low power, but require up to 200 microseconds to resume versus keeping the processor in a "lighter" (shallower) sleep state. Platform phase-locked loop (PLL) shutdown requires 20-50 microseconds to resume, versus 10's of nanoseconds with clock gating.

[0003] Due to the lack of awareness in device latency tolerance, some computing platforms maintain system resources in an available state (especially data paths and system memory) even during idle states. Maintaining these resources in an available state consumes power.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The detailed description is described with reference to the accompanying figures, in which:

[0005] FIGS. **1**A-**1**C are schematic block diagrams of portions of an apparatus that supports latency based platform coordination, according to some embodiments.

[0006] FIG. **2** is a flowchart illustrating operations in a method to implement latency based platform coordination, according to some embodiments.

[0007] FIG. **3** is a schematic timing diagram of an example of latency reporting and policy engine coordination, according to some embodiments.

[0008] FIG. **4** is a schematic illustration of a computer system, in accordance with some embodiments.

DETAILED DESCRIPTION

[0009] Described herein are exemplary systems and methods for implementing latency based platform coordination which, in some embodiments, may be implemented in an electronic device such as, e.g., a computer system. In the following description, numerous specific details are set forth to provide a thorough understanding of various embodiments. However, it will be understood by those skilled in the art that the various embodiments may be practiced without the specific details. In other instances, well-known methods, procedures, components, and circuits have not been illustrated or described in detail so as not to obscure the particular embodiments.

[0010] Embodiments of systems which implement latency based platform coordination will be explained with reference to FIGS. **1**A-**1**C and FIG. **2**. FIGS. **1**A-**1**C are schematic block diagrams of portions of an apparatus that supports latency based platform coordination, according to some embodiments. FIG. **2** is a flowchart illustrating operations in a method to implement latency based platform coordination, according to some embodiments.

[0011] Referring first to FIG. **1**A, a system to implement latency based platform coordination comprises one or more processors **110** and a platform control hub (PCH) **115**, which in combination are sometimes referred to as the root complex. A policy engine **130** is implemented in the system as an abstract device which comprises logic to implement latency based platform coordination. In some embodiments, the policy engine **130** may be implemented as logic instructions stored on a computer readable medium which, when executed by a processor configure the processor to implement latency based platform coordination operations. In some embodiments, the policy engine **130** may be reduced to logic, for example in a programmable logic device such as a field programmable gate array (FPGA) or may be reduced to hardwired circuit logic. The policy engine **130** may be implemented as a single, discrete entity, or may be distributed between multiple processing components in the root complex.

[0012] The system further comprises a plurality of components **125** coupled to the policy engine **130** by a bridge/switching device **120**. In some embodiments, each of the plurality of components reports (operation **210**) its snoop latency, alone or in combination with its non-snoop latency to the policy engine to the policy engine **130**. In the embodiment depicted in FIG. **1**, the latency parameters may be reported as a tuple, in which the snoop latency parameter is represented by the symbol Sn, where an identifies the component, and in which the non-snoop latency is represented by the symbol NSn, where an identifies the component. Thus, Lat(S1, NS1) represents the snoop and non-snoop latency parameters for the first component. Similarly, Lat(S2, NS2) represents the snoop and non-snoop latency parameters for the second component and Lat(S3, NS3) represents the snoop and non-snoop latency parameters for the third component. In practice, the system may comprise dozens or even hundreds of components.

[0013] In the embodiment depicted in FIG. **1**A, each of the components **125** reports its latency parameters through the bridge/switching device **120**, which receives the parameters at operation **215**. In other embodiments, one or more of the components **125** may be coupled directly to one of the processors **110** in the policy engine **130**, such that the device could report its latency parameter directly to the policy engine **130**. In some embodiments, the bridge/switching device **120** has a characteristic delay indicated in the drawings by the symbol Δ. The delay, Δ, associated with the bridge/switching device **120** may be variable as a function of the switching/transmission capacity associated with the bridge/switching device **120**, the traffic flowing through the bridge/switching device **120**, and the power state of the bridge/switching device **120**. For example, a bridge/switching device that is an inactive/idle state or sleep state would have a higher characteristic delay than a bridge/switching device **120** that is an active state. Similarly, a bridge/switching device **120** with a high traffic load would have a higher characteristic delay than a bridge/switching device **120** with a low traffic load.

[0014] In some embodiments, the bridge/switching device **120** comprises logic to selectively report latency parameters from the components **125** coupled to the bridge/switching device **120**. In addition, in some embodiments the bridge/switching device **120** comprises logic to modify the reported latency parameters in order to compensate for the delay, Δ,

associated with the bridge/switching device **120**. In one embodiment, the bridge/switching device implements logic to deduct the characteristic delay, Δ, associated with the bridge/switching device **120** from each of the latency parameters for each of the components coupled to the bridge/switching device **120**, at operation **220**. The bridge/switching device **120** may further implement logic to report the latency parameters to the policy engine **130**, at operation **225**. For example, the bridge/switching device **120** may report to the policy engine the MIN(Lat(S1−Δ, NS1−Δ), Lat(S2−Δ, NS2−Δ), Lat(S3−Δ, NS3−Δ)).

[0015] The policy engine **130** receives the reported latency parameters from the bridge/switching device **120** at operation **130**. In some embodiments, the policy engine **130** implements logic to compute a minimum latency tolerance value (operation **235**) from the latency parameters reported into the policy engine **130**. The policy engine **130** then uses a minimum latency tolerance value to determine a power management policy for the system.

[0016] FIG. **1**B is a schematic illustration of an example in which the system is in an active mode. Each of the components **125** reports their respective latency values into the bridge/switching device **120**. In an active state, the bridge/switching device **120** has a characteristic delay of 2 μs. As described above, the bridge/switching device receives the latency parameters from the respective components **125** and deducts the characteristic delay of 2 μs from the reported parameters. The bridge/switching device **120** then reports the minimum latency parameter tuple to the policy engine **130**.

[0017] FIG. **1**C is a schematic illustration of an example in which the system is in an active mode. Each of the components **125** reports their respective latency values into the bridge/switching device **120**. In an active state, the bridge/switching device **120** has a characteristic delay of 20 μs. As described above, the bridge/switching device receives the latency parameters from the respective components **125** and deducts the characteristic delay of 2o μs from the reported parameters. The bridge/switching device **120** then reports the minimum latency parameter tuple to the policy engine **130**.

[0018] FIG. **3** is a schematic timing diagram of an example of latency reporting and policy engine coordination, according to some embodiments. FIG. **3** illustrates the utilization of latency reporting while two policy engines (PE**1** and PE**2**) share latency information and coordinate to steer the appropriate C-states for microprocessors, memory controller power management and any other platform PLL power management. A device exhibiting a bursty traffic pattern with intermediate low power states when active and thus reporting a low latency tolerance. The microprocessor may resist entering deeper sleep states that would impact performance such as flushing its caches. However, when the device is idle and reports an extended latency tolerance, that information becomes helpful to enhance utilization of deeper sleep states for the platform and microprocessor while armed with the knowledge that any visible degradation impact is unlikely.

[0019] FIG. **4** is a schematic illustration of an architecture of a computer system which may implement latency based platform coordination n accordance with some embodiments. Computer system **400** includes a computing device **402** and a power adapter **404** (e.g., to supply electrical power to the computing device **402**). The computing device **402** may be any suitable computing device such as a laptop (or notebook) computer, a personal digital assistant, a desktop computing device (e.g., a workstation or a desktop computer), a rack-mounted computing device, and the like.

[0020] Electrical power may be provided to various components of the computing device **402** (e.g., through a computing device power supply **406**) from one or more of the following sources: one or more battery packs, an alternating current (AC) outlet (e.g., through a transformer and/or adaptor such as a power adapter **404**), automotive power supplies, airplane power supplies, and the like. In one embodiment, the power adapter **404** may transform the power supply source output (e.g., the AC outlet voltage of about 110 VAC to 240 VAC) to a direct current (DC) voltage ranging between about 7 VDC to 12.6 VDC. Accordingly, the power adapter **404** may be an AC/DC adapter.

[0021] The computing device **402** may also include one or more central processing unit(s) (CPUs) **408** coupled to a bus **410**. In one embodiment, the CPU **408** may be one or more processors in the Pentium® family of processors including the Pentium® II processor family, Pentium® III processors, Pentium® IV processors, Core and Core2 processors available from Intel® Corporation of Santa Clara, Calif. Alternatively, other CPUs may be used, such as Intel's Itanium®, XEON™, and Celeron® processors. Also, one or more processors from other manufactures may be utilized. Moreover, the processors may have a single or multi core design.

[0022] A chipset **412** may be coupled to the bus **410**. The chipset **412** may include a memory control hub (MCH) **414**. The MCH **414** may include a memory controller **416** that is coupled to a main system memory **418**. The main system memory **418** stores data and sequences of instructions that are executed by the CPU **408**, or any other device included in the system **400**. In some embodiments, the main system memory **418** includes random access memory (RAM); however, the main system memory **418** may be implemented using other memory types such as dynamic RAM (DRAM), synchronous DRAM (SDRAM), and the like. Additional devices may also be coupled to the bus **410**, such as multiple CPUs and/or multiple system memories.

[0023] In some embodiments, main memory **418** may include a one or more flash memory devices. For example, main memory **418** may include either NAND or NOR flash memory devices, which may provide hundreds of megabytes, or even many gigabytes of storage capacity.

[0024] The MCH **414** may also include a graphics interface **420** coupled to a graphics accelerator **422**. In one embodiment, the graphics interface **420** is coupled to the graphics accelerator **422** via an accelerated graphics port (AGP). In an embodiment, a display (such as a flat panel display) **440** may be coupled to the graphics interface **420** through, for example, a signal converter that translates a digital representation of an image stored in a storage device such as video memory or system memory into display signals that are interpreted and displayed by the display. The display **440** signals produced by the display device may pass through various control devices before being interpreted by and subsequently displayed on the display.

[0025] A hub interface **424** couples the MCH **414** to an input/output control hub (ICH) **426**. The ICH **426** provides an interface to input/output (I/O) devices coupled to the computer system **400**. The ICH **426** may be coupled to a peripheral component interconnect (PCI) bus. Hence, the ICH **426** includes a PCI bridge **428** that provides an interface to a PCI bus **430**. The PCI bridge **428** provides a data path between the CPU **408** and peripheral devices. Additionally, other types of

I/O interconnect topologies may be utilized such as the PCI Express™ architecture, available through Intel® Corporation of Santa Clara, Calif.

[0026] The PCI bus **430** may be coupled to a network interface card (NIC) **432** and one or more disk drive(s) **434**. Other devices may be coupled to the PCI bus **430**. In addition, the CPU **408** and the MCH **414** may be combined to form a single chip. Furthermore, the graphics accelerator **422** may be included within the MCH **414** in other embodiments.

[0027] Additionally, other peripherals coupled to the ICH **426** may include, in various embodiments, integrated drive electronics (IDE) or small computer system interface (SCSI) hard drive(s), universal serial bus (USB) port(s), a keyboard, a mouse, parallel port(s), serial port(s), floppy disk drive(s), digital output support (e.g., digital video interface (DVI)), and the like.

[0028] System **400** may further include a basic input/output system (BIOS) **450** to manage, among other things, the boot-up operations of computing system **400**. BIOS **450** may be embodied as logic instructions encoded on a memory module such as, e.g., a flash memory module.

[0029] The terms "logic instructions" as referred to herein relates to expressions which may be understood by one or more machines for performing one or more logical operations. For example, logic instructions may comprise instructions which are interpretable by a processor compiler for executing one or more operations on one or more data objects. However, this is merely an example of machine-readable instructions and embodiments are not limited in this respect.

[0030] The terms "computer readable medium" as referred to herein relates to media capable of maintaining expressions which are perceivable by one or more machines. For example, a computer readable medium may comprise one or more storage devices for storing computer readable instructions or data. Such storage devices may comprise storage media such as, for example, optical, magnetic or semiconductor storage media. However, this is merely an example of a computer readable medium and embodiments are not limited in this respect.

[0031] The term "logic" as referred to herein relates to structure for performing one or more logical operations. For example, logic may comprise circuitry which provides one or more output signals based upon one or more input signals. Such circuitry may comprise a finite state machine which receives a digital input and provides a digital output, or circuitry which provides one or more analog output signals in response to one or more analog input signals. Such circuitry may be provided in an application specific integrated circuit (ASIC) or field programmable gate array (FPGA). Also, logic may comprise machine-readable instructions stored in a memory in combination with processing circuitry to execute such machine-readable instructions. However, these are merely examples of structures which may provide logic and embodiments are not limited in this respect.

[0032] Some of the methods described herein may be embodied as logic instructions on a computer-readable medium. When executed on a processor, the logic instructions cause a processor to be programmed as a special-purpose machine that implements the described methods. The processor, when configured by the logic instructions to execute the methods described herein, constitutes structure for performing the described methods. Alternatively, the methods described herein may be reduced to logic on, e.g., a field programmable gate array (FPGA), an application specific integrated circuit (ASIC) or the like.

[0033] In the description and claims, the terms coupled and connected, along with their derivatives, may be used. In particular embodiments, connected may be used to indicate that two or more elements are in direct physical or electrical contact with each other. Coupled may mean that two or more elements are in direct physical or electrical contact. However, coupled may also mean that two or more elements may not be in direct contact with each other, but yet may still cooperate or interact with each other.

[0034] Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least an implementation. The appearances of the phrase "in one embodiment" in various places in the specification may or may not be all referring to the same embodiment.

[0035] Although embodiments have been described in language specific to structural features and/or methodological acts, it is to be understood that claimed subject matter may not be limited to the specific features or acts described. Rather, the specific features and acts are disclosed as sample forms of implementing the claimed subject matter.

What is claimed is:

1. A method to implement latency based platform coordination in an electronic device, comprising:
- receiving, in a policy engine, latency data from one or more components in the electronic device;
- computing a minimum latency tolerance value from the latency data; and
- determining a power management policy from the minimum latency tolerance value.

2. The method of claim **1**, wherein receiving, in a policy engine, latency data from one or more components in the electronic device comprises receiving a snoop latency tolerance and a non-snoop latency tolerance from the one or more components

3. The method of claim **2**, wherein:
- the latency data from the one or more components is transmitted via an intermediate bridge/switch device
- the bridge has at least one delay value for data transmitted via the bridge/switch device; and
- the bridge deducts the delay value from the latency data.

4. The method of claim **3**, wherein:
- the bridge comprises a first delay value when the bridge is in a low power state and a second delay value when the bridge is in an active power state; and
- the bridge deducts one of the first delay value or the second delay value from the latency data.

5. The method of claim **1**, wherein computing a minimum latency tolerance value from the latency data comprises:
- comparing a plurality of latency values received from a plurality of components; and
- selecting the lowest latency value from the plurality of latency values.

6. The method of claim **1**, wherein the policy engine monitors latency values over time during operation of the electronic device and updates power management policies as a function of changes in the latency tolerance values.

7. The method of claim **1**, wherein determining a power management policy from the minimum latency tolerance value comprises selecting a sleep state that permits the system to meet the minimum latency tolerance value.

8. An electronic apparatus, comprising:

at least one processor;

a plurality of components; and

a policy engine comprising logic to:

receive latency data from one or more components in the electronic device;

compute a minimum latency tolerance value from the latency data; and

determine a power management policy from the minimum latency tolerance value.

9. The electronic apparatus of claim **8**, wherein the policy engine further comprises logic to receive a snoop latency tolerance and a non-snoop latency tolerance from the one or more components

10. The electronic apparatus of claim **9**, wherein:

the latency data from the one or more components is transmitted via an intermediate bridge/switch device

the bridge has at least one delay value for data transmitted via the bridge/switch device; and

the bridge deducts the delay value from the latency data.

11. The electronic apparatus of claim **10**, wherein:

the bridge comprises a first delay value when the bridge is in a low power state and a second delay value when the bridge is in an active power state; and

the bridge deducts one of the first delay value or the second delay value from the latency data.

12. The electronic apparatus of claim **8**, wherein the policy engine further comprises logic to:

compare a plurality of latency values received from a plurality of components; and

select the lowest latency value from the plurality of latency values.

13. The electronic apparatus of claim **8**, wherein the policy engine further comprises logic to monitor latency values over time during operation of the electronic device and updates power management policies as a function of changes in the latency tolerance values.

14. The electronic apparatus of claim **8**, wherein the policy engine further comprises logic to select a sleep state that permits the system to meet the minimum latency tolerance value.

* * * * *