



- (51) **International Patent Classification:** Not classified
- (21) **International Application Number:**
PCT/IN2012/000275
- (22) **International Filing Date:**
17 April 2012 (17.04.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
1336/CHE/2011 18 April 2011 (18.04.2011) IN
- (71) **Applicant (for all designated States except US):** **INEDA SYSTEMS PVT. LTD** [IN/IN]; 8-2-120/115/c, Sudha Enclave, Road No. 2, Banjara Hills, Hyderabad 500034 (IN).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** **KANIGICHERLA, Balaji** [US/IN]; 8-2-120/115/c, Sudha Enclave, Road No. 2, Banjara Hills, Hyderabad 500034 (IN). **VOLETI, Siva Raghuram** [IN/IN]; 8-2-120/115/c, Sudha Enclave, Road No. 2, Banjara Hills, Hyderabad 500034 (IN). **GUTTA, Srinivasa** [IN/IN]; 8-2-120/115/c, Sudha Enclave, Road No. 2, Banjara Hills, Hyderabad 500034 (IN). **DOM-METI, Surya Narayana** [IN/IN]; 8-2-120/115/c, Sudha Enclave, Road No. 2, Banjara Hills, Hyderabad 500034

(IN). **ANNAPUREDDY, Narsi Reddy** [IN/IN]; 8-2-120/115/c, Sudha Enclave, Road No. 2, Banjara Hills, Hyderabad 500034 (IN). **ANNAPURNA, Sarada** [IN/IN]; 8-2-120/115/c, Sudha Enclave, Road No. 2, Banjara Hills, Hyderabad 500034 (IN). **TANDABOINA, Krishna Mohan** [IN/IN]; 8-2-120/115/c, Sudha Enclave, Road No. 2, Banjara Hills, Hyderabad 500034 (IN).

(74) **Agents:** **LAKSHMIKUMARAN, Varadachari et al.**; B-6/10, Safdarjung Enclave, New Delhi 110029 (IN).

(81) **Designated States (unless otherwise indicated, for every kind of national protection available):** AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States (unless otherwise indicated, for every kind of regional protection available):** ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU,

[Continued on next page]

(54) **Title:** POWER MANAGEMENT IN MULTI HOST COMPUTING SYSTEMS

(57) **Abstract:** The present subject matter discloses methods and systems of power management in multiple host computing system (100) running multiple operating systems. In one embodiment, a multi host platform management module (126) facilitates power management in multi host computing system (100). In said implementation, the multi host platform management module (126) includes an arbitration and interpreter module (128) to intercept and arbitrate power management commands issued by any of the hosts of the multi host computing system (100).

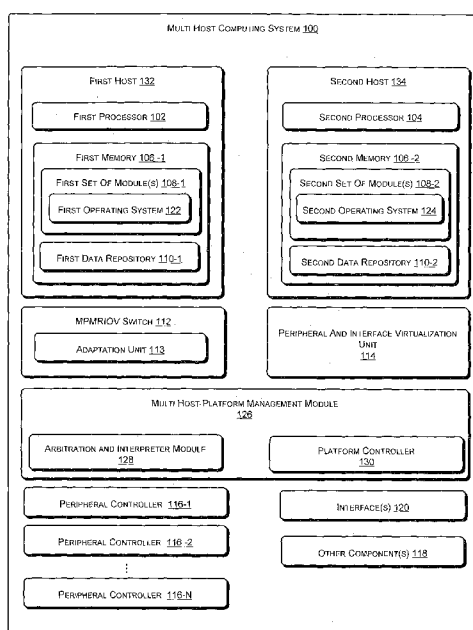


Fig. 1



TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

POWER MANAGEMENT IN MULTI HOST COMPUTING SYSTEMS

FIELD OF INVENTION

[0001] The present subject matter relates to power management and, particularly but not exclusively, to power management of hardware platforms in multi host computing systems running multiple operating systems.

BACKGROUND

[0002] Conventionally, computing systems implement various techniques to implement power management, which is controlled when a peripheral device is to be switched on, put in a low power state such as stand-by mode or is to be turned off. Conventional computing systems receive power through an alternating current (AC) power plug-in or through installed batteries that are usually chargeable. In computing systems, power management at hardware level is achieved by using various techniques such as clock gating, power gating, turning of power supply to parts of circuits, which are in a state of inactivity for a prolonged period of time, etc. Further, other techniques such as dynamic voltage scaling, or dynamic frequency scaling, or a combination of both are used to implement power management in computing systems.

[0003] In the dynamic voltage scaling technique, the voltage supplied to a component is increased or decreased, depending upon circumstances. For example the voltage may be decreased to conserve power, particularly in laptops, netbooks and other mobile devices, where energy comes from a battery and is limited. On the other hand, the voltage supply to a component may be increased in order to increase processing performance or to increase reliability. In the dynamic frequency scaling technique, the frequency of a processor is automatically adjusted either to conserve power or to reduce the amount of heat generated by the chip. Sometimes dynamic voltage scaling and dynamic frequency scaling are used concurrently to optimize power consumption.

[0004] Further certain software tools are also used to implement power management in computing systems. For example Advanced Power Management (APM) is used to implement power management in computing devices at a Basic Input Output System (BIOS) level. APM uses a layered approach to manage peripheral devices and hardware components. APM-aware applications such as device drivers communicate with an operating system specific APM driver. This APM driver communicates to the APM-aware BIOS, which in turn controls the peripheral device at the hardware level. It is also possible for the device driver to communicate directly

with the peripheral device. In general, the APM driver acts as an intermediary between the BIOS and the operating system.

[0005] In recent years, Advanced Configuration and Power Interface (ACPI) specification provides an open standard for unified operating system-centric device configuration and power management. The ACPI specification defines platform-independent interfaces for hardware discovery, configuration, power management and monitoring. The specification is central to Operating System-directed configuration and Power Management (OSPM); which describes a system implementing ACPI. The basic difference between ACPI and APM is that ACPI assigns the responsibility of power management to the operating system whereas APM assigns the responsibility of power management to the BIOS. ACPI has been adopted as the standard by the computer industry to implement power management techniques in computing systems.

SUMMARY

[0006] This summary is provided to introduce concepts related to power management of hardware platform in multi host computing systems running multiple operating systems. This summary is not intended to identify essential features of the claimed subject matter nor is it intended for use in determining or limiting the scope of the claimed subject matter.

[0007] In an embodiment, a multi host platform management module facilitates power management in a multi host computing system. In said implementation, the multi host platform management module includes an arbitration and interpreter module to intercept and arbitrate power management commands issued by any of the hosts of the multi host computing system.

BRIEF DESCRIPTION OF THE FIGURES

[0008] The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Some embodiments of system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

[0009] Fig. 1 shows exemplary components of a multi host computing system, in accordance with an embodiment of the present subject matter.

[0010] Fig. 2 shows the exemplary components of a multi host detachable system, according to an embodiment of the present subject matter.

[0011] Fig. 3a shows the exemplary components of the multi host computing system, in accordance with an embodiment of the present subject matter.

5 [0012] Fig. 3b shows the exemplary components of the multi host computing system, in accordance with an embodiment of the present subject matter.

[0013] Fig. 4 shows the exemplary components of a single host power monitoring system, in accordance with an embodiment of the present subject matter.

10 [0014] Fig. 5 shows the exemplary components of a multi host platform management module, in accordance with an embodiment of the present subject matter.

[0015] It should be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative systems embodying the principles of the present subject matter. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudo code, and the like represent various processes which may be
15 substantially represented in computer readable medium and so executed by a computing system or processor, whether or not such computing system or processor is explicitly shown.

DETAILED DESCRIPTION

[0016] Systems and methods for power management of a hardware platform in multi host computing systems running multiple operating systems are described herein. The systems
20 and methods can be implemented in a variety of computing devices such as laptops, desktops, workstations, tablet-PCs, smart phones, etc. Although the description herein is described with reference to certain computing systems, the systems and methods may be implemented in other electronic devices, albeit with a few variations, as will be understood by a person skilled in the art.

25 [0017] Conventionally, ACPI specification is used to implement power management schemes in a computing system. The ACPI specification defines various states of activity of the processor, peripheral devices, hardware platform as a whole, etc., to actively implement power management schemes. For example, in a X86 processor running Microsoft® Windows®, various processor states such as C0 (running), C1 (Halt), etc., hardware platform states such as S0
30 (running), S3 (standby), S4 (hibernate), etc., and peripheral device states such as D0 (on), D3 (off), etc., are specified. The state of a processor or the hardware platform as a whole, or a

peripheral device is usually determined by monitoring the activity of the peripheral device, the processor, etc. The computing systems usually use a platform power manager to optimize power consumption by complying with the APCI specifications.

[0018] However, in a multi host computing system that is running multiple operating systems, the conventional platform power manager is unable to implement power schemes in a proper way. For example, say a multi host system includes Host 1 and Host 2 running operating systems OS 1 and OS 2 respectively. If the OS 1 issues a shutdown command, the conventional platform power manager will switch off power supply to the entire hardware platform irrespective of the activity state of the OS 2 which may be actively running on the Host 2. Similarly, if the OS 1 monitors a peripheral device to be idle and issues a command to move the peripheral device to a low power state, the conventional platform power manager will move the device to a low power state irrespective of the activity status of the peripheral device with respect to the OS 2. In case the OS 1 and the OS 2 are heterogeneous operating systems, i.e. have different file system, different protocols, etc., both will have their own implementation of the ACPI specification, since the ACPI is implemented by the operating system. Further, in implementation of the ACPI, the power management at OS interface level and power management command structure varies from operating system to operating system. Thus exchanging power state information from one operating system to another is difficult. This makes conventional platform power manager unsuitable for use in a multi host computing system.

[0019] The present subject matter discloses methods and systems of power management in multiple host computing systems running multiple operating systems. A multi host computing system is a multi processor computing system which has a plurality of processors which are similar or different and consists of same or varying processing power and is capable of running multiple operating systems simultaneously. Further the multi host computing systems are capable of sharing the hardware platform such as peripheral devices like display devices, audio devices, input devices such as keyboard, mouse, touchpad, etc. among the plurality of processors running multiple operating systems simultaneously.

[0020] In one embodiment, the multi host computing system uses a Multi-Root Input Output Virtualization (MRIOV) switch electronically connected to at least one of the plurality of the processors, a Peripheral and Interface Virtualization Unit (PIVU) connected to the MRIOV

switch to enable peripheral sharing between the multiple operating systems running on the multiple processors. In another embodiment, other techniques may be used to enable peripheral sharing between the multiple operating systems running on the multiple processors. Various types of peripheral devices may be connected to the multi host computing system. For example, the multi host computing system may include or may be connected to various storage controllers like Serial Advanced Technology Attachments (SATA), NAND flash memory, Multimedia Cards (MMC), Consumer Electronics Advanced Technology Attachment (CEATA); connectivity modules like baseband interfaces, Serial Peripheral Interfaces (SPI), Inter-integrated Circuit (I2C), infrared data association (IrDA) compliant devices; media controllers like camera, integrated inter chip sound (I2S); media accelerators like audio encode-decode engines, video encode-decode engines, graphics accelerator; security modules like encryption engines, key generators; communication modules like Bluetooth, Wi-Fi, Ethernet; universal serial bus (USB) connected devices like pen drives, memory sticks, etc.

[0021] In one implementation, the multi host computing system includes a Multi Host Platform Management Module (MHPMM), to implement ACPI specifications in a multi host computing system. The MHPMM further includes an arbitration and interpreter module, henceforth referred to as AI module, to intercept, prioritize, arbitrate, and interpret power management command issued by various operating systems running on the multiple hosts. The MHPMM can be implemented as a hardware device, a software module or a firmware running on a microcontroller, etc.

[0022] In one embodiment of the multi host computing system, each of the operating systems running on the multiple hosts runs a kernel level driver which generates power management commands. The kernel level driver may be thought of as similar to the ACPI driver of the conventional computing systems. The power management commands generated by any of the operating systems are received by the AI module. The power management commands may be related to the hardware platform as a whole, state of a peripheral device, etc., and may originate from any of the operating systems running on the multi host computing system in any order. The AI module buffers the power management commands received from any of the operating systems, interprets them, and sends system responses back to the operating system. The AI module further translates the power management commands to low level platform specific commands and forwards them to a platform controller for execution. In certain cases the AI

module may send fake system responses back to the operating system. For example, say the OS 1 issues a command to power off the Hard Disk Drive (HDD) unit. The AI module receives the command and checks the activity state of the HDD unit and finds that HDD unit is being actively used by the OS 2. Then the AI module will send a system response indicating that the HDD unit has been tuned off without actually turning it off or changing its state. Further the AI module saves the context and the power state of various devices, hardware platform with respect to each of the operating systems running on the multi host computing system. For example, in the above mentioned scenario, the OS 1 sends a power management command to power on the HDD unit. The AI module will receive the command, check the state of the HDD unit and then send a system response to OS 1. While sending the system response, the AI module will retrieve that OS 1 had previously send a power management command to turn off the HDD unit and will account for the same when sending the system response. These and other features and advantages will be described in greater detail in conjunction with the following figures

[0023] It will also be appreciated by those skilled in the art that the words during, while, and when as used herein are not exact terms that mean an action takes place instantly upon an initiating action but that there may be some small but reasonable delay, such as a propagation delay, between the initial action and the reaction that is initiated by the initial action. Additionally, the word "connected" is used throughout for clarity of the description and can include either a direct connection or an indirect connection.

[0024] Fig. 1 shows the exemplary components of the multi host computing system, henceforth referred to as the system 100, according to an embodiment of the present subject matter. The system 100 can either be a portable electronic device, like laptop, notebook, netbook, tablet computer, etc., or a non-portable electronic device like desktop, workstation, server, etc. The system 100 comprises a first processor 102 and a second processor 104. The first processor 102 and the second processor 104 are coupled to a first memory 106-1 and a second memory 106-2 respectively.

[0025] The first processor 102 and the second processor 104 can be implemented as one or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, or any devices that manipulate signals based on operational instructions. Among other capabilities, the first processor 102 and the second

processor 104, may be configured to fetch and execute computer-readable instructions and data stored in the first memory 106-1 and the second memory 106-2.

[0026] The first memory 106-1 and the second memory 106-2 can include any computer-readable medium known in the art including, for example, volatile memory (e.g., RAM) and/or non-volatile memory (e.g., flash, etc.). The first memory 106-1 and the second memory 106-2 include first set of module(s) 108-1 and second set of module(s) 108-2 respectively (collectively referred to as module(s) 108). Further, first memory 106-1 and the second memory 106-2 also include a first data repository 110-1 and a second data repository 110-2 respectively. The module(s) 108 usually includes routines, programs, objects, components, data structure, etc., that perform a particular task or implement particular abstract data types.

[0027] In one embodiment, the system 100 includes a Multi-Protocol Multi-Root Input Output Virtualization (MPMRIOV) switch 112 which facilitates the communication of the system 100 with connected peripherals (not shown in figure). It may be mentioned that Peripheral Component Interconnect Special Interest Group (PCI-SIG), an electronics industry consortium responsible for specifying the Peripheral Component Interconnect (PCI) and Peripheral Component Interconnect Express (PCIe) computer buses, states Multi-Root Input Output Virtualization (MRIOV) as the industry standard for enabling virtualization of peripherals among multiple processors.

[0028] The MPMRIOV switch 112 comprises an adaptation unit 113, which facilitates communication with peripherals, that may be non-PCI and non-PCIe compliant peripherals, with the system 100. A peripheral and interface virtualization unit 114 is coupled to a plurality of peripheral controllers 116-1, 116-2, ..., 116-N, collectively referred to as peripheral controllers 116 hereinafter. The peripheral and interface virtualization unit 114 helps in virtualization of the physical devices and facilitates their simultaneous sharing among multiple operating systems or multiple processors. The physical devices may include, but are not limited to, printer, keyboard, mouse, and display unit. The system 100 may also include other components 118 required to provide additional functionalities to the system 100. In other embodiments, the system 100 virtualizes the peripheral devices and the hardware platform without using the MPMRIOV switch 112 and the peripheral and interface virtualization unit 114, and may use other techniques to virtualize the peripheral devices and the hardware platform.

[0029] The peripherals can be configured to be used exclusively by either of the first processor 102 or the second processor 104 or by both the first processor 102 and the second processor 104 simultaneously. Additionally, the system 100 has one or more interface(s) 120 to connect to external networks, systems, peripherals, devices, etc.

5 [0030] When the system 100 is booted, a primary operating system is loaded. In one example, a first operating system 122, referred to as OS-A hereinafter, running on the first processor 102 may be designated as the primary operating system while a second operating system 124 referred to as OS-B, running on the second processor 104 is treated as the secondary operating system. The system 100 can concurrently run multiple operating systems on the first
10 processor 102 and the second processor 104. If multiple operating systems are present, the system 100 allows the user to designate any of the operating systems as the primary operating system. The user can change the primary operating system according to user's choice and/or requirement. The system 100 also allows the user to switch from one operating system to another operating system seamlessly.

15 [0031] The system 100 also includes a Multi Host Platform Management Module (MHPMM) 126, to implement power management of the hardware platform and of the peripheral devices. In one embodiment, the MHPMM 126 includes an arbitration and interpreter (AI) module 128, which is configured to receive and interpret power management commands. The MHPMM 126 further includes a platform controller 130 which monitors, controls, and
20 maintains the power and activity states of various peripheral devices (not shown) and interface(s) 120.

[0032] The system 100, for the ease of explanation, has been depicted as having two hosts, a first host 132 and a second host 134. However, it will be known to those skilled in the art that the same concepts may be extended to any number of hosts. The first host 132 includes the
25 first processor 102 and the first memory 106-1. Similarly, the second host 134 includes the second processor 104 and the second memory 106-2. The other components of the system 100 are concurrently shared by the two hosts 132 and 134.

[0033] The hosts 132 and 134 may run homogeneous or heterogeneous operating systems. As mentioned earlier, each operating system has its own implementation of ACPI
30 specifications. The hosts 132 and 134 monitor, and control the power and the activity states of the hardware platform as a whole and the peripheral devices independently through the

MHPMM 126. However, the hosts 132 and 134 are unaware of the activity state of any peripheral devices or of the hardware platform with respect to the other host. For example, the audio controller may not be used and hence idle in the first host 132 and may be actively used by the second host 134 to render audio from any media player being used by a user to play a song.

5 In this scenario, the first host 132 is unaware that the audio controller is being used actively by the second host 134.

[0034] The hosts 132 and 134 issue power management commands independently. The MHPMM 126 receives and buffers the power management commands. The power management commands are processed by the AI module 128. The AI module 128 may implement any of the
10 conventional algorithms such as round robin so that the power management commands issued by any of the hosts 132 and 134 have a fair chance of being processed. The AI module 128 interprets the received power management commands and sends a system response back to the operating system which generated the power management command.

[0035] For example, say the AI module 128 receives a power management command
15 from the first host 132. The AI module 128 then obtains the status of a peripheral device or of the hardware platform from the peripheral and interface virtualization unit 114. A peripheral device may be solely used by either of the hosts 132 and 134 or may be shared between the two hosts 132 and 134. If a peripheral device is solely used by the first host 132, the AI module 128 translates the power management commands related to the peripheral device to a low level
20 platform specific instruction and passes it to the platform controller 130 for execution. If the peripheral device is shared by the two hosts 132 and 134, then the AI module 128 determines the state of the peripheral device. If the peripheral device is being used by the other host, i.e. the second host 134, then the AI module 128 does not change the state of the peripheral device and fakes a system response indicating the power state of the peripheral device has been successfully
25 changed in accordance with the request from first host 132.

[0036] The platform controller 130 also implements actions defined by hardware switches such as power on, shutdown, sleep, etc., which are usually vendor specific. The platform controller 130 enables or disables the hardware switches in accordance with the instructions received from the AI module 128. For example, say the first operating system 122
30 running on the first host 132 is turned off. The AI module 128 interprets this request and forwards the request to the platform controller 130 to turn off the power supply to the first host

132 and the peripheral devices which are being exclusively used by the first host 132. All the shared peripherals which are used by both the hosts 132 and 134 undergo no transition in power state, however, a fake signal is sent to the first host 132 indicating that all shared peripherals are powered off

5. [0037] In case of a trigger, such as user pressing a power button, to switch on the first host 132, the AI module 128 receives the request and checks the host for which the trigger is meant, such as the first host 132. The AI module 128 then instructs the platform controller 130 to enable power supply to the first host 132 and the peripheral devices being exclusively used by the first host 132. Additionally, the AI module 128 checks the status of the shared peripherals. If
10. any of the shared peripherals are powered off or are in a low power state, then the AI module 128 passes an instruction to the platform controller 130 to power on the shared peripheral devices. Then the first operating system 122 is booted, the kernel drivers are loaded and a notification is sent to the user. The AI module 128 then becomes ready to arbitrate power management commands from both the hosts 132 and 134.

15. [0038] Thus, the MHPMM 126 makes the system 100 more suitable and implements power management techniques such as ACPI between multiple hosts such as hosts 132 and 134. The MHPMM 126 resolves conflicting power management commands received from either of the hosts 132 and 134 and resolves them, so that both the hosts 132 and 134 can run smoothly. Moreover the MHPMM 126 can interpret power management requests sent by various types of
20. operating systems such as the first operating system 122 and the second operating system 124, with the help of the kernel level driver. Thus, MHPMM 126 helps to enable power management techniques in multi host system 100.

[0039] Fig. 2 shows an exemplary multi host detachable system 200. In one embodiment the multi host detachable system 200 is a tablet laptop system. The multi host detachable system
25. 200 includes a display unit 202 and a base unit 204. The display unit 202 includes a first host 206. The first host 206, among other components, has a first processor 208 and first memory 210. Similarly, the base unit 204 comprises a second host 212. The second host among other components includes a second processor 214 and a second memory 216.

[0040] The first processor 208 and the second processor 214 can be implemented as one
30. or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, or any devices that manipulate signals based

on operational instructions. Among other capabilities, the first processor 208 and the second processor 214 can be configured to fetch and execute computer-readable instructions and data stored in the first memory 210 and the second memory 216 respectively.

[0041] The first memory 210 and the second memory 216 can include any computer-readable medium known in the art including, for example, volatile memory (e.g., RAM) and/or non-volatile memory (e.g., flash, etc.). The first memory 210 and the second memory 216 include module(s) and data. The modules usually includes routines, programs, objects, components, data structure, etc., that perform particular task or implement particular abstract data types.

[0042] The display unit 202 and the base unit 204 include a first platform management module 218-1 and a second host platform management module 218-2, collectively referred to as platform management modules 218 hereinafter. The platform management modules 218 are structurally and functionally similar to MHPMM 126. The first host platform management module 218-1 includes a first arbitration and interpreter module 220-1 and a first platform controller 222-1. Similarly, the second host platform management module 218-2 includes a second arbitration and interpreter module 220-2 and a second platform controller 222-2. The arbitration and interpreter modules 220-1 and 220-2, collectively referred to as arbitration and interpreter modules 220, are structurally and functionally similar to AI module 128. Similarly platform controllers 222-1 and 222-2, collectively referred to as platform controllers 222, are structurally and functionally similar to the platform controller 130.

[0043] Moreover, the display unit 202 and the base unit 204 include a first peripheral and interface virtualization unit 224-1 and a second first peripheral and interface virtualization unit 224-2 respectively (collectively referred to as peripheral and interface virtualization units 224), to enable sharing of peripheral devices among the base unit 202 and the display unit 204. The peripheral and interface virtualization units 224 are structurally and functionally similar to the peripheral and interface virtualization unit 114. Further, the display unit 202 and the base unit 204 include a first battery 226-1 and a second battery 226-2 to power the display unit 202 and the base unit 204 respectively. The first battery 226-1 and the second battery 226-2 can be electronically coupled to a first battery charger 228-1 and a second battery charger 228-2 respectively (collectively referred to as battery chargers 228).

[0044] The battery chargers 228 are connected to a first AC adaptor 230-1 and a second AC adaptor 230-2 which facilitate the connection of the battery chargers 228 to an AC supply. The display unit 202 and the base unit 204 also includes a set of first interfaces 232-1 and a set of second interfaces 232-2, which enable the multi host detachable system 200 to communicate
5 with other computing and communication devices either directly or through a network.

[0045] The base unit 204 further includes a switch 234 which can detect the attachment or detachment of the display unit 202 and trigger events based on the detection. The switch 234 may be implemented as a hardware device such as a latch or a software module or as a firmware. Each of the display unit 202 and the base unit 204 may include other components (not shown in
10 figure) for providing different functionalities. It would be appreciated by those skilled in the art that the switch 234, as described, may be included in the display unit 202.

[0046] In one implementation, when the multi host detachable system 200 is booted or initialized in an attached state. i.e., the display unit 202 and the base unit 204 are connected to each other, the second platform controller 222-2 acts as the master platform controller
15 irrespective of whether the user input is to boot the first host 206 or the second host 212. The first peripheral and interface virtualization unit 224-1 and the second peripheral and interface virtualization unit 224-2 communicate with each other using a communication link as indicated by 236. The communication between the peripheral and interface virtualization units 224 enable each of the display unit 202 and the base unit 204 to use any peripheral connected to either of the
20 units 202 and 204.

[0047] On receiving a user input to detach the display unit 202 and the base unit 204, the platform controllers 222 implement power management techniques in the base unit 204 and the display unit 202 independent of each other. The peripheral and interface virtualization units 224 communicate the detachment instructions with each other and enable/disable the peripherals
25 based on the detachment.

[0048] In one embodiment, when the display unit 202 and the base unit 204 are attached, the second ac adaptor 230-2 may be used to supply power to both the first and second battery chargers 228-1 and 228-2, to charge the batteries 226. In a detached state, the first AC adaptor 230-1 can used to supply power to the first battery charger 228-1 to charge the first battery 226-
30 1, and similarly the second AC adaptor 230-2 can used to supply power to the second battery charger 228-2 to charge the second battery 226-2. The charging of the batteries 226 is controlled

by the switch 234. Further other configurations are also possible. For example, in an attached state, the first battery 226-1 and the second battery 226-2 may be connected in series or in parallel or may not be connected at all. It should be appreciated by those skilled in the art that the above mentioned implementations are not exhaustive and other configurations would also be possible with no or little modifications as known by those skilled in the art.

[0049] Fig. 3a shows the multi host computing system 100, in accordance with a second embodiment of the present subject matter. In said embodiment, both the first host 132 and the second host 134 implement ACPI compliant power management schemes. The first operating system 122 includes a first kernel 302-1, a first ACPI driver(s) 304-1, a first device driver(s) 306-1 and a first operating system power manager 308-1, henceforth referred to as the first OSPM 308-1. Similarly, the second operating system 124 includes a second kernel 302-2, a second ACPI driver(s) 304-2, a second device driver(s) 306-2 and a second operating system power manager 308-2, henceforth referred to as the second OSPM 308-2.

[0050] The first kernel 302-1 and the second kernel 302-2, collectively referred to as the kernels 302, are the core component of the operating systems 122 and 124. The Kernels 302 acts as a bridge between applications and the actual data processing done at the hardware level. The kernels' 302 responsibilities include managing the system's 100 hardware resources such as processing power, memory utilization, etc. The Kernels 302 provide the lowest-level abstraction layer for the resources especially for processors, 102 and 104, and peripheral devices that an application software accesses to provide various functionalities to the user. The kernels 302 facilitate the access of peripheral devices by application software through various techniques such as inter-process communication mechanisms and system calls.

[0051] The first set of ACPI drivers 304-1 and the second set of ACPI drivers 304-2, collectively referred to as the ACPI drivers 304, interact with the kernels 302 and the device drivers, 306-1 and 306-2, to implement the ACPI compliant power management schemes. The first ACPI driver 304-1 controls a first set of ACPI registers 310-1 and the second ACPI driver 304-2 controls a second set of ACPI registers 310-2. The first set of ACPI registers 310-1 and the second set of ACPI registers 310-2 is collectively referred to as ACPI registers 310. The ACPI register 310 facilitate the implementation of various fixed and generic features to be supported by the peripheral devices and the hardware platform. Since ACPI defines power management schemes to be implemented by the operating system, each of the first operating system 122 and

the second operating system 124 include the first OSPM 308-1 and the second OSPM 308-2 respectively. The OSPM 308 controls the ACPI registers 310 to implement various power management schemes.

[0052] As mentioned earlier, the ACPI registers implement both fixed and generic features to be supported by the peripheral devices and the hardware platform. The fixed features have exact definition for implementation. If a fixed feature is to be used, the fixed feature has to be implemented as per ACPI specifications as the OPSM 308 manipulates the registers implementing the fixed feature, of the ACPI registers in a pre-defined way and expects a pre-defined response. However, it should be appreciated by those skilled in the art, that though it is not necessary to use all the fixed features mentioned in ACPI specifications, if a fixed feature is to be implemented, the implementation has to be as per ACPI specifications. In one implementation, the OSPM 308 controls the various registers responsible for implementing the fixed features such as the PM1 status registers, PM1 enable registers, PM1 control registers, PM timer register, PM2 control register, Processor control register, Processor LVL2 register, Processor LVL3 register, etc.

[0053] Further a set of generic registers are defined in the ACPI registers 310, to implement various value added power management features to a peripheral device or to the hardware platform. The ACPI provides a mechanism to allow generic features to be defined in the ACPI namespace of the OPSM 308. In one embodiment, the programming bits are stored in the generic hardware address spaces and the top-level bits are stored in the general purpose event registers such as General Purpose Event 0 Status Register, General Purpose Event 0 Enable Register, etc. In said embodiment, the General Purpose registers are used to generate an event or signal required by the platform, which can further trigger a platform specific control process like turning off the display unit. The status registers are used to capture any system level events and depending on which any platform specific decisions can be taken.

[0054] Fig. 3b shows the multi host computing system 100, in accordance with a third embodiment of the present subject matter. In said embodiment, the first host 132 implements ACPI compliant power management schemes and the second host 134 implements APM compliant power management schemes. As stated earlier, the first host 132 runs the first operating system 122 which includes the first kernel 302-1, the first ACPI driver(s) 304-1, the first device driver(s) 306-1 and the first OSPM 308-1. In said embodiment, the second host 134

includes a set of APM aware applications 312. The APM aware applications 312 are a set of application that facilitates the power management in the system 100 through an operating system dependent APM driver 314. The APM driver 314 communicates control instructions to an APM Basic Input Output System 316, also refereed to as the APM BIOS 316. The APM BIOS 316 provides power management functionality for the hardware platform and peripheral devices adhering to the APM specification.

[0055] In said implementation, the APM BIOS 316 is configured to communicate with the MHPMM 126. In this scenario, the APM BIOS communicates control instructions for power management to the MHPMM126 and does not interact with the hardware platform or the peripheral devices directly. In another implementation, since APM specification does not impose any restriction on the implementation of programming model for power management in the hardware platform and the peripheral devices, the APM BIOS 316 can be configured so as to follow the ACPI defined specifications. In said implementation, the MHPMM 126 intercepts, analyzes and responds to the control instructions generated by the APM BIOS 316.

[0056] It should be understood by those skilled in the art that though Fig. 3a and Fig. 3b describe the system 100 in context of the first host 132 and second host 134, the same concept can be extended for an number of hosts albeit little modifications. Further, for clarity of explanation, in Fig. 3a the system 100 has been described as having both the hosts 132 and 134 implementing power management schemes compliant with ACPI standards, whereas in Fig. 3b the system 100 has been described as having the first host 132 complying with ACPI specifications and the second host 134 complying with APM specification. However, the same should not be construed as a limitation. The disclosed embodiments are exemplary implementations of the system 100. The system 100 can be modified to include any number of hosts, with each host implementing any power management technique, albeit slight variations as would be known by those skilled in the art.

[0057] Fig. 4 shows the exemplary components of a single host power monitoring system 400, in accordance with one embodiment of the present subject matter. The single host power monitoring system 400, henceforth referred to as the SHPMS 400, is an implementation of the multi host computing system 100 wherein one of the hosts is assigned the responsibility of power management of the whole SHPMS 400. For the ease of explanation, the SHPMS is described as having two hosts and complying with ACPI specifications. However, it should be appreciated by

those skilled in the art that the same concept may be extended to any number of hosts albeit little variations.

5 [0058] In one embodiment, the SHPMS 400 includes a power monitoring host 402 and a guest host 404. The power monitoring host 402 is the host which has been assigned the responsibility of power management of the whole SHPMS 400. As stated before, though the SHPMS 400 has been depicted as having a single guest host 404, in another implementation, the SHPMS 400 may include any number of guest hosts. The power monitoring host 402 includes a set of ACPI drivers 406. The ACPI drivers 406 are analogous to the ACPI drivers 304. The ACPI drivers 406 communicate with a set of ACPI registers 408 to implement various ACPI
10 compliant power management schemes. The ACPI registers 408 are analogous to the ACPI registers 310. Further, the power monitoring host 402 includes an ACPI server module 410, which manages the power scheme for the SHPMS 400. The ACPI server module 410 virtualizes all the ACPI features and presents each of the hosts 402 and 404 with a separate cop of ACPI feature set.

15 [0059] The guest host 404 runs an ACPI filter 412 which filter all ACPI related commands and instructions. In one implementation, the ACPI server module 410 and the ACPI filter 412 exchange information over a low latency high bandwidth data bus 414. In one embodiment, the low latency high bandwidth data bus 414 may be implemented as a inter host inter process communication (IPC) link. Further the SHPMS 400 also includes a set of ACPI
20 Hardware Controls 416, such as a switch, Sleep Button, Thermal Controls, etc., which also trigger specific power related actions. For example, pressing a key combination may trigger a shutdown and power off instruction in the SHPMS 400.

[0060] In operation, the ACPI server module 410 also ensures that all power state change options implemented by the guest host 404, is also implemented by the power monitoring host
25 402. For example, if the guest host 404 sets an input pin "A" as a wake up source, the ACPI server module 410 also sets the input pin "A" as a wake up source for the power monitoring host 402. Hence if the input pin "A" toggles in a low power state, such as sleep mode, the ACPI server module 410 first wakes up the power monitoring host 402 and brings it to a fully operational state. In turn the power monitoring host 402 wakes up the guest host 404 and brings
30 it to a full operational state. Thus it follows that during boot up, the power monitoring host 402 is the first to boot up and initiate the boot process for the guest host 404. Similarly during shutdown

or powering off, the power monitoring host 402 shutdowns the guest host 404 and is the last to be shutdown or powered off. It should be appreciated by those skilled in the art that though the SHPMS 400 has been described in context of complying with ACPI specifications, the same concept can be extended to comply with other power management standards such as APM.

5 **[0061]** Fig. 5 shows the exemplary components of the Multi Host Platform Management Module (MHPMM) 126, in accordance with an embodiment of the present subject matter. As mentioned earlier, the MHPMM 126 includes the AI module 128 and the platform controller 130. In one implementation, the AI module 128 includes a first host low pin controller input (LPC In) 502-1 and a second host low pin controller input (LPC In) 502-2 which receives power
10 management and control instructions from the first host 132 (not shown) and the second host 134 (not shown) respectively. The AI module 128 also includes an operating system aware routing module 504, referred to as OS aware routing module 504 which arbitrates power management commands from both the operating systems running on the first host 132 and the second host 134. Further the OS aware routing module 504 also keeps a track of power related events such as
15 shutdown, standby, sleep, etc., and also sends system responses to respective operating systems running on either of the hosts 132 and 134.

[0062] A virtualization controller interface 506 interacts with the peripheral and interface virtualization unit 114 to obtain the status of various peripheral devices. For example, the peripherals may be shared by both the hosts 132 and 134 or may be used exclusively by either of
20 the hosts 132 and 134. The AI module 128 also includes one or more hand off block(s) 508, which buffers the power management commands received from either of the hosts 132 and 134, converts the commands to a low level hardware instruction and forwards it to a low pin count (LPC) host driver 510 for execution.

[0063] The low pin count (LPC) host driver 510 is electronically coupled with a low pin
25 count (LPC) driver 512 of the platform controller 130. The LPC driver 512 is connected to various types of peripheral controller interface(s) 514. Example of such peripheral devices are input devices such as keyboard, mouse, cooling devices such as fans, power devices such as battery units, etc. The LPC driver 512 is used to control the peripheral devices. Further the platform controller 130 includes an interrupt handler 516 which analyzes and implements
30 interrupts generated by the system. For example, a system interrupt may be generated due to the pressing of the power button of the computing system. The platform controller 130 also includes

one or more hot keys 518 which may be used for implementing various power states or change the power scheme of either or both the hosts 132 and 134. For example, one of the hot key(s) 518 may be used to move the first host 132 to a low power state, another of the hot key(s) 518 may be used to switch off the second host 134.

5 **[0064]** In the said implementation the platform controller 130 includes a power event handler 520. The power event handler 520 controls the voltage regulators (not shown in figure) based on the power management commands received from the AI module 128. The voltage regulators control power supply to various peripheral devices and to the hardware platform as a whole. Further the MHPMM 126 may also include other components for facilitating additional
10 functionalities.

[0065] Moreover, the MHPMM 126 also implements various controls to comply with the power management specifications such as ACPI. The exemplary table (Table-1) states how various features are described in the ACPI programming model specifications and how the features are implemented by the MHPMM 126.

Table 1		
Feature Name	ACPI defined programming Model	Implementation by MHPMM 126
Power Management Timer	Fixed Hardware Feature Control Logic	Separate timers are implemented for each of the hosts.
Power Button	Fixed Hardware Event and Control Logic or Generic Hardware Event and Control Logic	Configurability to associate power button to a chosen host or based on pre-defined logic, associate power button with any of the hosts
Sleep Button	Fixed Hardware Event and Control Logic or Generic Hardware Event and Control Logic	Configurability to associate sleep button to a chosen host or based on pre-defined logic, associate sleep button with any of the hosts
Lid Switch	Generic Hardware Event Feature	Configurability to associate Lid switch to a chosen host or based on pre-defined logic, associate

		Lid switch with any of the hosts
Power Button Override	Not Applicable	Implemented as a hardware or software configurable feature to turn off any or all the hosts or trigger pre-defined actions based on platform.
Real Time Clock Alarm	Optional Fixed Hardware Event	Software or Hardware implemented feature to change the power state of any or all the hosts based on platform requirement.
Sleep/ Wake Control Logic	Fixed Hardware Event and Control Logic	Software or Hardware implemented feature to interpret power state of each host and change the power state of the system based on the interpretation.

[0066] For example, the system 100 includes a feature referred to as a power management timer. Conventionally, the power management timer is usually a 24 bit or a 32-bit free running timer. A free running timer is configured to run from end to end without reloading or stopping at intermediate states. The free running timer counts the input pulses from zero to the maximum count and on reaching the maximum count, sets a flag (which can be used to generate an interrupt) and resets itself to zero and continues the counting process. A free running timer can be used to generate interrupts at regular intervals and to generate accurate delays. The MHPMM 126 virtualizes the free running timer so that each host, such as the first host 132 and the second host 134, of the system 100 can access and use the free running timer independently.

[0067] In another example, the system 100 includes a power button. In conventional systems, the power button is usually used by the user to change the power state of the conventional system such as powering the conventional system on or off, etc. However in the system 100, the MHPMM 126 configures the power button to be associated with the hosts 132

and 134 of the system 100. In one implementation, the power button may be configured to be associated with a particular host, say the first host 132. In this scenario, the power button may be used to change the power state of the first host 132 without affecting the other hosts such as the second host 134. In another implementation, the power button may be configured to be associated with more than one host such as both the hosts 132 and 134. In such a scenario, the power button may change the power state of either or both the hosts 132 and 134 based on pre-defined programmed logic. The sleep button and the lid switch conventionally used in portable computing systems such as netbooks, laptops, may be configured and virtualized by the MHPMM 126 in a mechanism similar to that of the power button. Similarly, the functionalities of the other features as mentioned in Table 1 are also virtualized by the MHPMM 126.

[0068] Thus the multi host platform management module 126 facilitates power management in the multi host computing system 100. Though the multi host computing system 100 has been depicted as having two hosts 132 and 134, it should be appreciated by those skilled in the art that the same may be extended to multi host computing systems having any number of hosts with little or no modification.

[0069] Although implementations for power management in a multi host computing system have been described in language specific to structural features and/or methods, it is to be understood that the appended claims are not necessarily limited to the specific features or methods described. Rather, the specific features and methods are disclosed as exemplary implementations for application sharing.

We claim:

1. A multi-host platform management unit (MHPMU) (126) for power management in multi-host computing system (100), the MHPMU (126) comprising:

an arbitration and interpreter (AI) module (128) configured to

receive power management commands, which conform to at least one of advanced configuration and power interface (ACPI) specification and advanced power management (APM) specification, from a plurality of operating systems running on a multi-host computing system (100);

arbitrate the received power management commands from the plurality of operating systems; and

generate a system response, based on the received power management commands, the system response comprising at least low level instruction, wherein the at least one low level instruction, when executed, implements at least one power management scheme in at least one component of the multi-host computing system (100), and wherein the power management scheme regulates the supply of power to the at least one component of the multi-host computing system (100).

2. The MHPMU (126) as claimed in claim 1, wherein the AI module (128) is further configured to:

determine whether a peripheral device is shared between a plurality of hosts of the multi-host computing system (100);

ascertaining whether the peripheral device is in use by at least one of the plurality of hosts on determining the peripheral device to be shared between the plurality of hosts;

executing the received power management commands based on the ascertaining.

3. The MHPMU (126) as claimed in claim 1, wherein the AI module (128) further comprises an operating system (OS) aware routing module (504) configured to

track a power related event associated with each of the plurality of operating systems for facilitating arbitration of the received power management commands, wherein the power related event comprise at least one of a shutdown event, standby event, sleep event, hibernate event and a restart event.

4. The MHPMU (126) as claimed in claim 1, wherein the AI module (128) further comprises an operating system (OS) aware routing module (504) configured to track status indicators, wherein the status indicators comprise at least one of a battery level indicator, and a human interface device (HID).
5. The MHPMU (126) as claimed in claim 1, wherein the arbitration is based on at least one of a round robin mode and a user defined mode.
6. The MHPMU (126) as claimed in claim 1, wherein the AI module (128) further comprises at least one hand-off block (508) configured to
buffer the power management commands from the plurality of operating systems;
and
convert each of the power management commands to a hardware level instruction so as to generate the system response.
7. The MHPMU (126) as claimed in claim 1, wherein the AI module (128) further comprises at least one hand-off block (508) configured to obtain a power status of a peripheral communicatively coupled to the multi-host computing system (100).
8. The MHPMU (126) as claimed in claim 1, wherein the MHPMU (126) further comprises a platform controller (130) configured to execute the system response generated by the AI module (128).
9. The MHPMU (126) as claimed in claim 8, wherein the platform controller (130) further comprises a power event handler configured to control at least one regulator, wherein the at least one regulator controls a supply of power to at least one component of the multi-host computing system (100).
10. A multi-host computing system (100) comprising:
a plurality of processors; and
the MHPMU (126) as claimed in any of the preceding claims; wherein the MHPMU (126) is communicatively coupled to at least one of the plurality of processors.
11. A multi-host computing system (100) comprising:
a first host (132) configured to manage power in the multi-host computing system (100), wherein the first host (132) further comprises:

a set of ACPI drivers (406) configured to communicate with at least one operating system running on the multi-host computing system (100) to implement ACPI compliant power management schemes; and

a ACPI server module (410), communicatively coupled to the set of ACPI drivers (406), wherein the ACPI server module (410) is configured to generate a virtual copy of at least one feature of the ACPI compliant power management schemes.

12. The multi-host computing system (100) as claimed in claim 11 further comprising:

a second host (134) configured to run at least one operating system, second host (134) comprising:

at least one ACPI filter (412) communicatively coupled to the ACPI server module (410), wherein the at least one ACPI filter (412) is configured to:

filter ACPI commands received from the ACPI server module (410); and

execute the filtered ACPI commands so as to implement at least one feature of the ACPI compliant power management schemes.

13. The multi-host computing system (100) as claimed in claim 11, wherein the second host (134) further comprises:

a APM driver (314) configured to communicate with at least one operating system running on the multi-host computing system (100) to implement APM compliant power management schemes; and

an APM basic input output system (BIOS) (316) communicatively coupled to the APM driver (314) configured to execute power management commands to implement at least one feature of the APM compliant power management schemes.

14. The multi-host computing system (100) as claimed in claim 11, wherein a power management scheme implemented by the multi-host computing system (100) is at least one of the ACPI power management scheme, the APM power management scheme, and an user defined power management scheme.

15. A method for power management in a multi-host computing system (100), the method comprising:

receiving at least one power management command from at least one of an application specific power option and an user selectable OS power option;

invoking at least one service to transit the multi-host computing system (100) to a power state, based on the at least one power management command, wherein the at least one service conform to at least one of advanced configuration and power interface (ACPI) specification and advanced power management (APM) specification; and

generating at least one low level instruction to control a supply of power to at least one component of the multi-host computing system (100) based on the at least one service.

16. The method for power management in a multi-host computing system (100) as claimed in claim 15, wherein the receiving further comprises obtaining an input to change the power state of a first host (132) of the multi-host computing system (100) to a current power state.

17. The method for power management in a multi-host computing system (100) as claimed in claim 16, wherein the method further comprises:

retrieving a previous power state of the first host (132) from a power state table;

generating at least one instruction to initiate the change of power state of the first host (132) from the previous power state to the current power state; and

changing an activity state of an operating system running on the first host (132) based on the received input.

18. The method for power management in a multi-host computing system (100) as claimed in claim 15, wherein the method further comprises:

receiving an input to detach a first host (132) of the multi-host computing system (100) from a second host (134) of the multi-host computing system (100);

decoupling at least one peripheral device from the multi-host computing system (100) based on the input;

disconnecting a power supply from the second host (134) to the first host (132); and

generating a notification for a user of the multi-host computing system (100), wherein the notification indicates the first host (132) can be safely detached from the second host (134).

19. The method for power management in a multi-host computing system (100) as claimed in claim 18, wherein the method further comprises configuring a first battery (226-1) to provide power supply to the first host on disconnecting the power supply of the first host (132) from the second host (134).

5 20. The method for power management in a multi-host computing system (100) as claimed in claim 18, wherein the method further comprises:

configuring a first battery (226-1) of the first host (132) and a second battery (226-2) of the second host (134) as a single battery to power the multi-host computing system (100); and

10 charging the single battery using a second battery charger (228-2) of the second host (134).

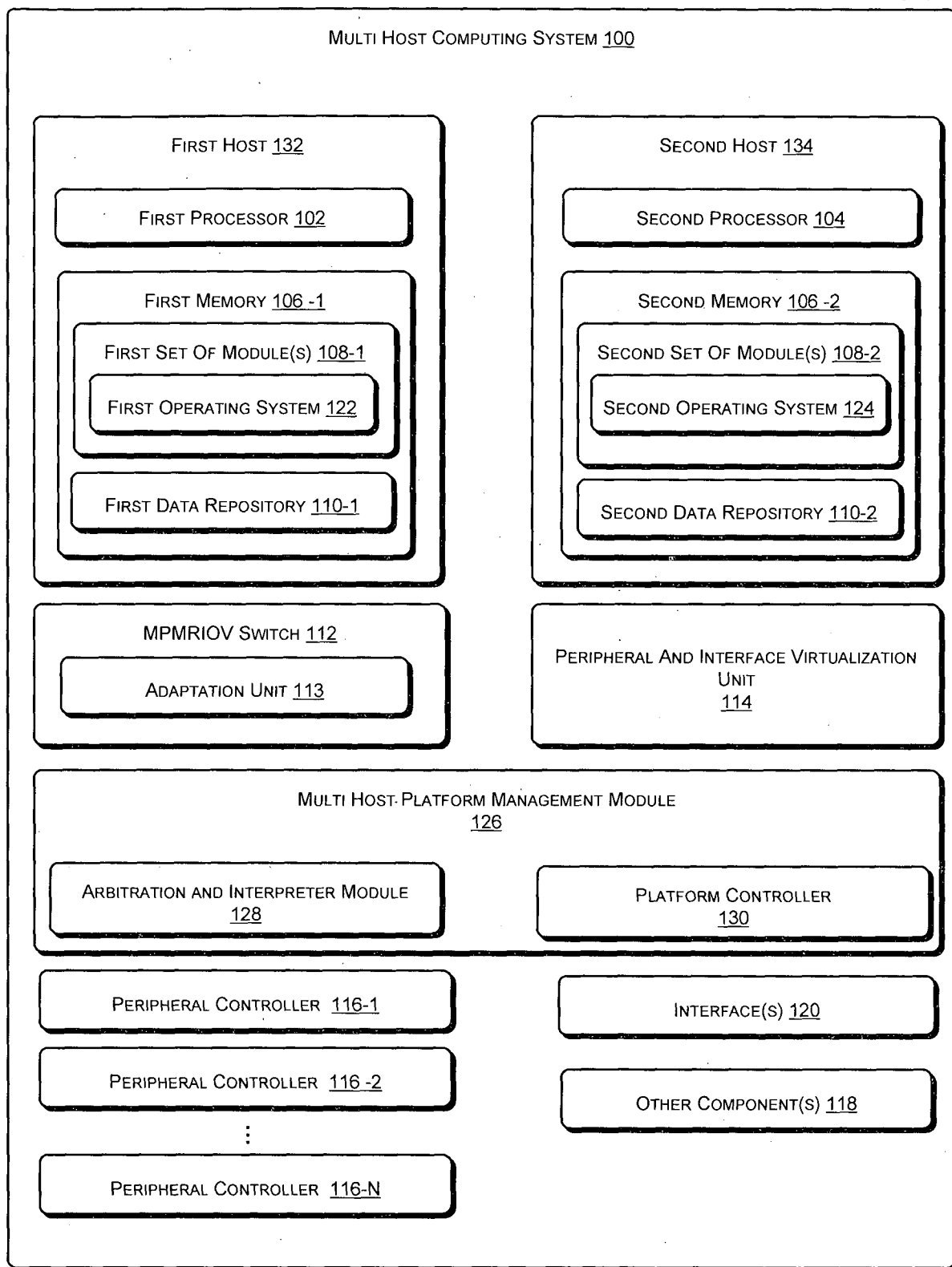


Fig. 1

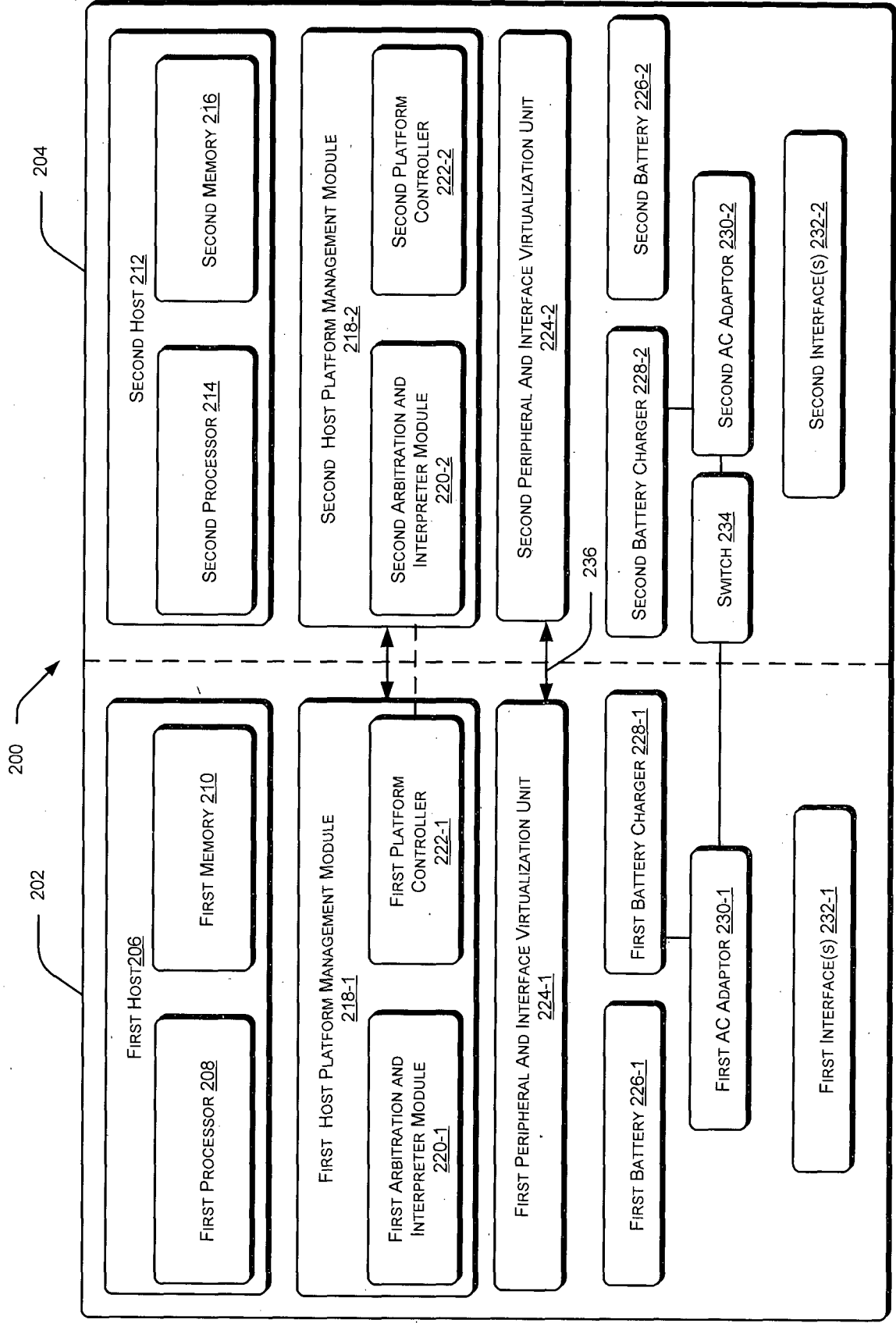


Fig. 2

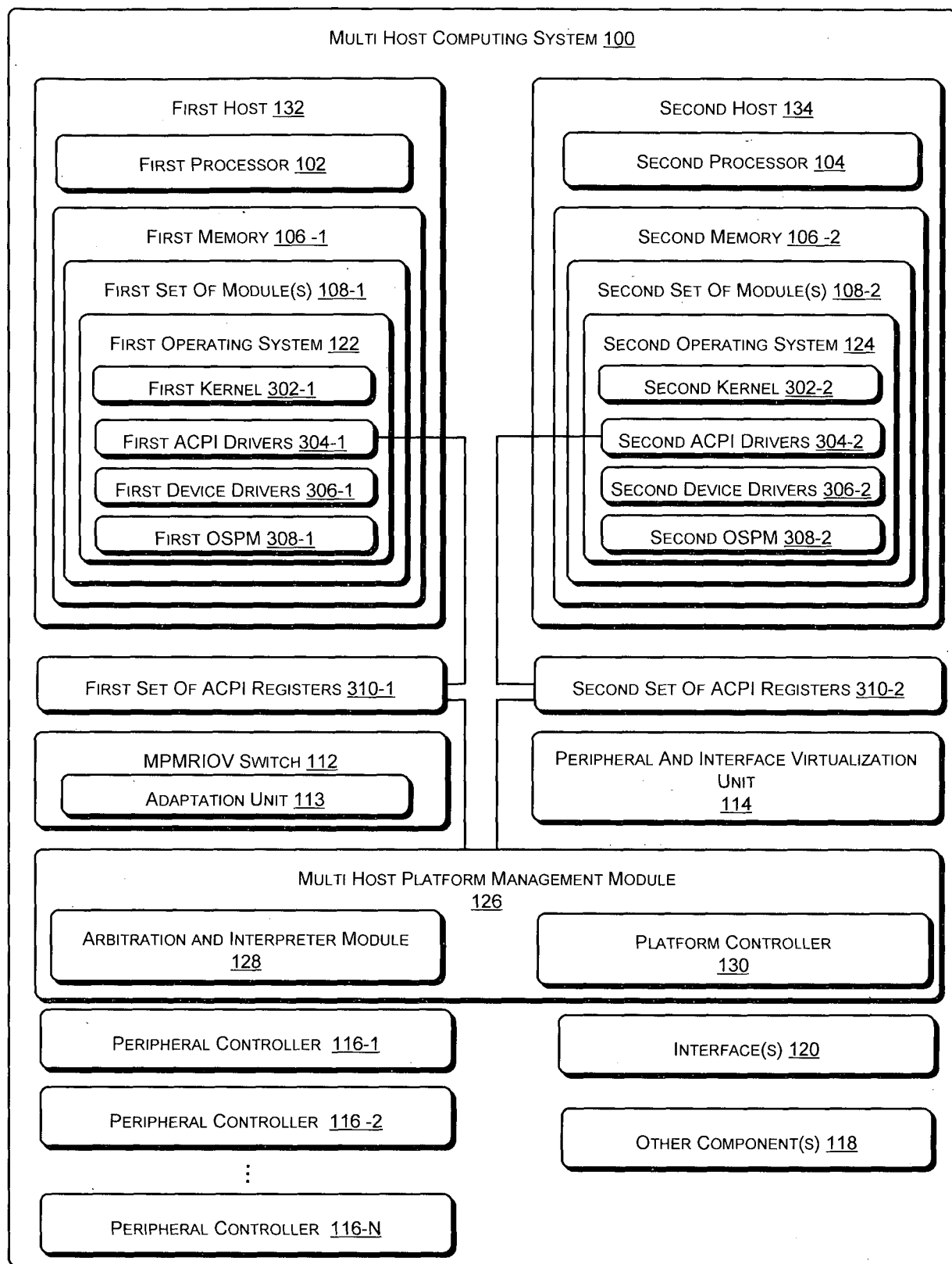


Fig. 3a

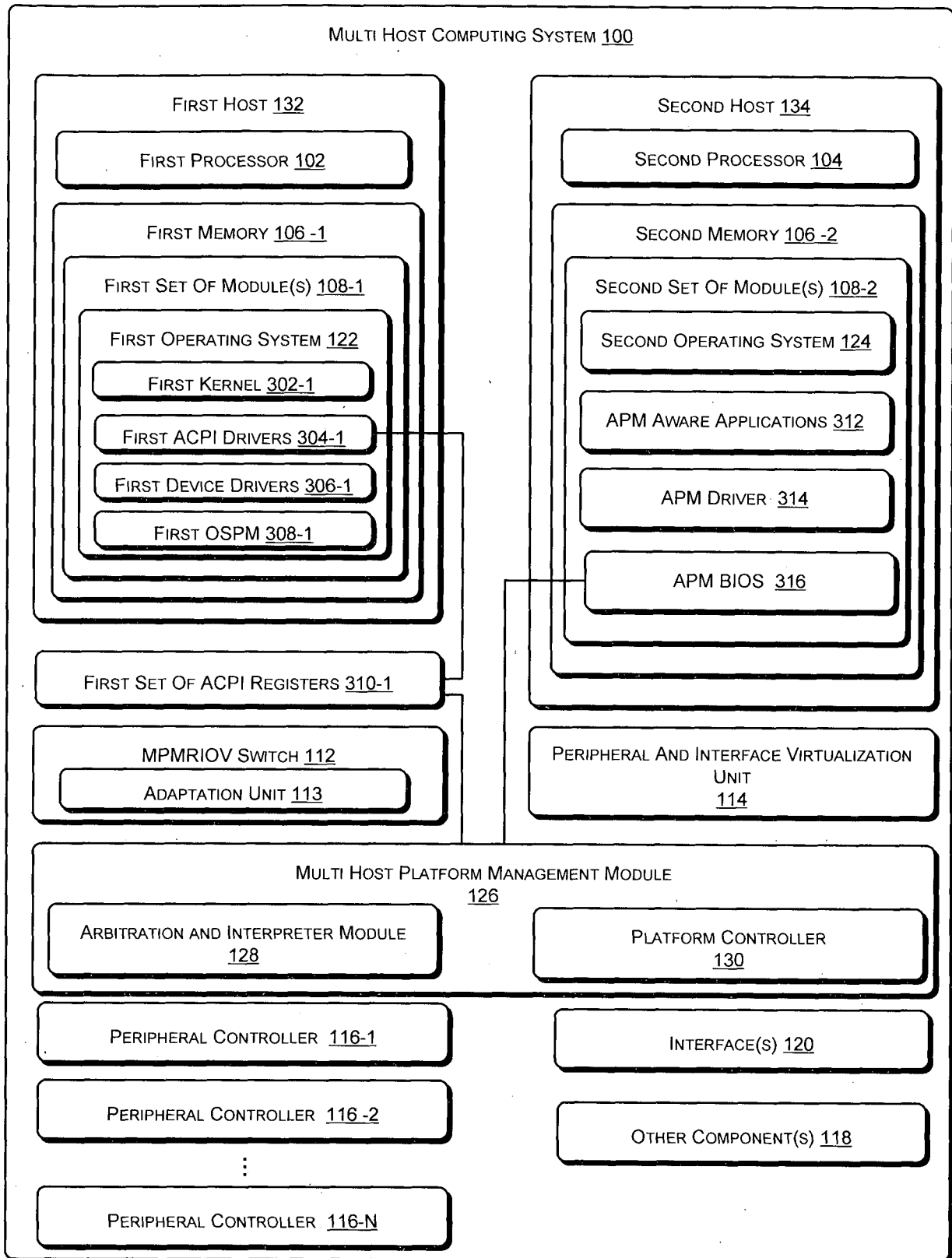


Fig. 3b

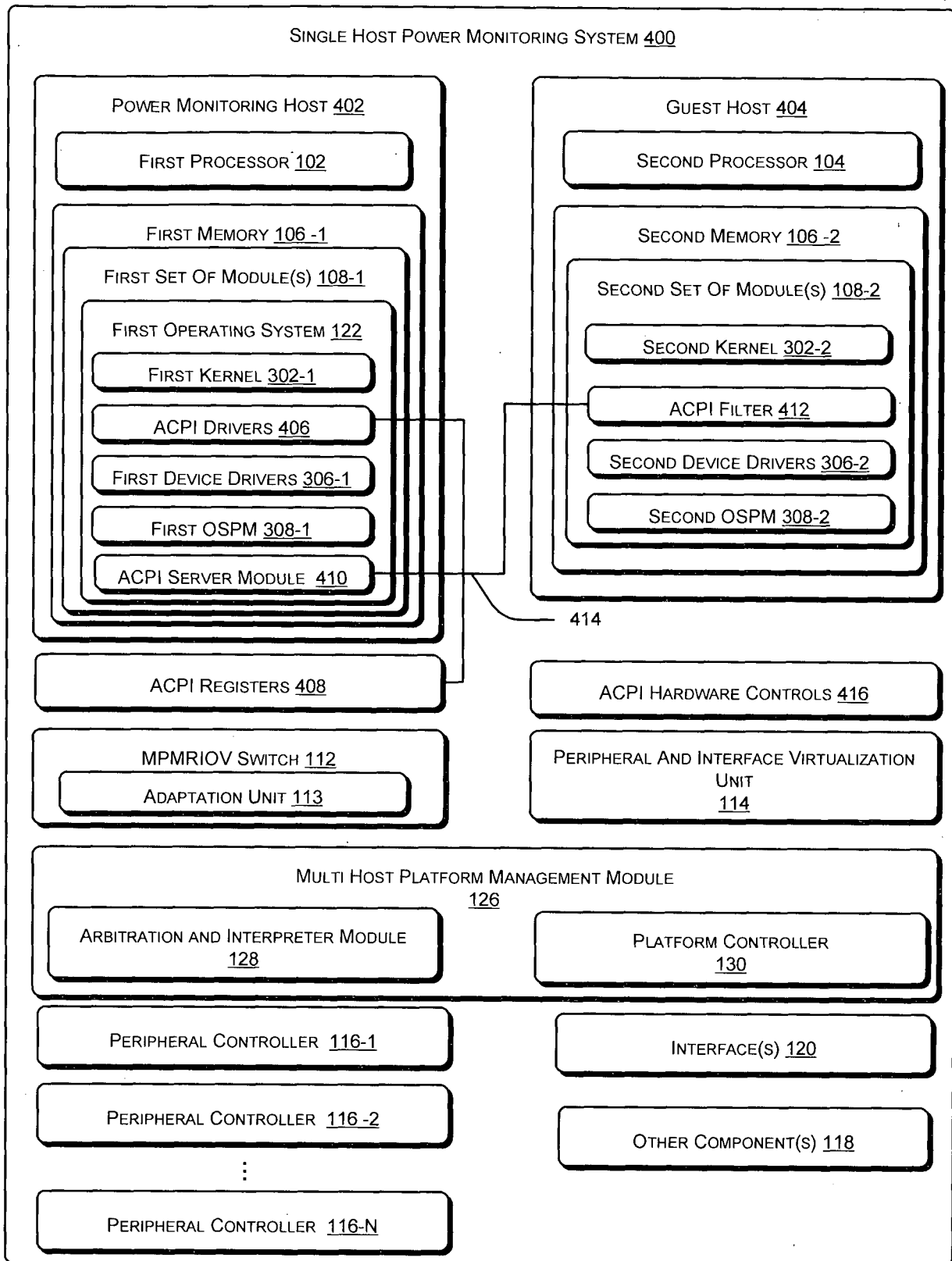


Fig. 4

126 →

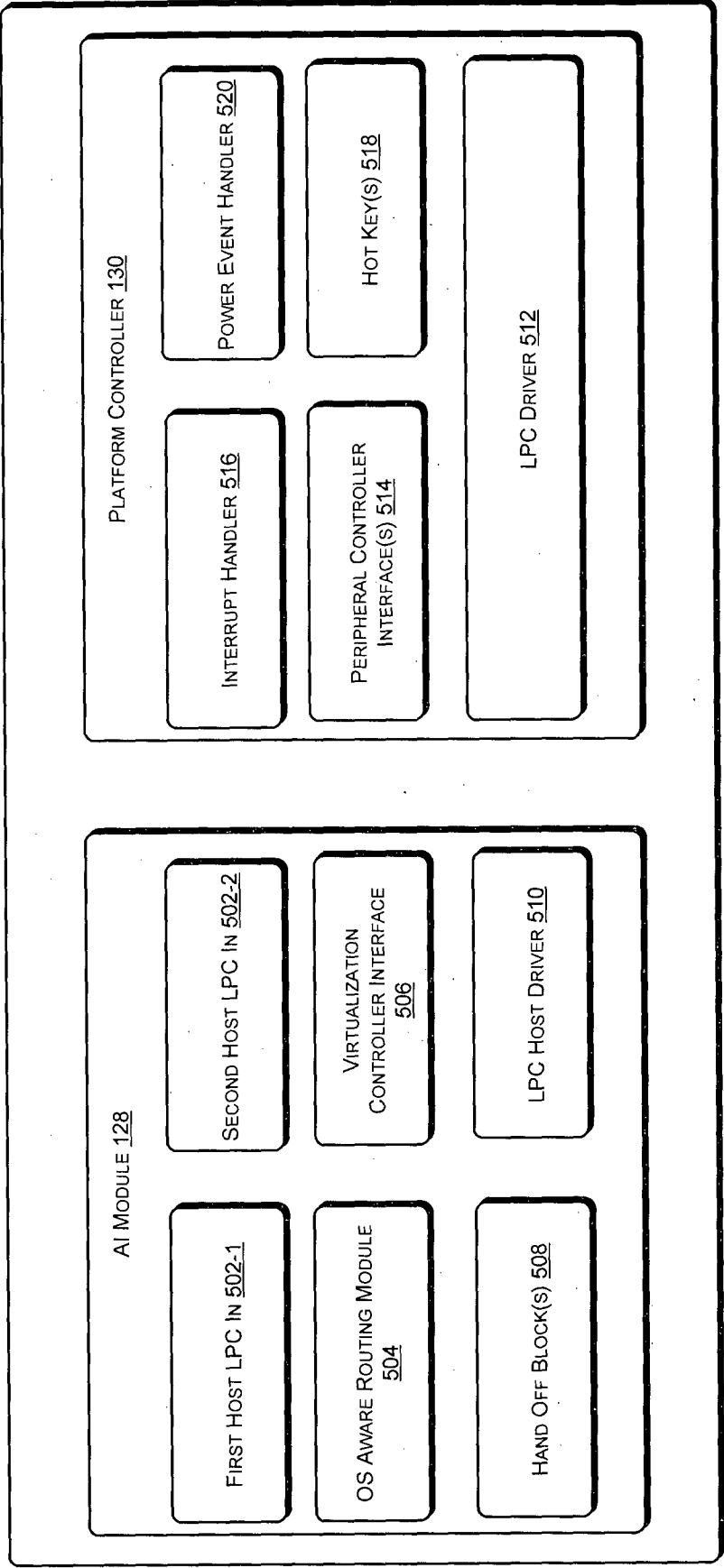


Fig. 5