(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 June 2006 (08.06.2006)

PCT

(10) International Publication Number
WO 2006/060237 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/US2005/042314

(22) International Filing Date:
17 November 2005 (17.11.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/004,710 3 December 2004 (03.12.2004) US

(71) Applicant *(for all designated States except US)*: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95052 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **POISNER, David** [US/US]; 205 Penry Square, Folsom, California 95630 (US). **STEVENS, William** [US/US]; 109 Prisser Way, Folsom, California 95630 (US).

(74) Agents: **VINCENT, Lester, J.** et al.; Blakely Sokoloff Taylor & Zafman, 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, 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*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 2006/060237 A2

(54) Title: PREVENTION OF DATA LOSS DUE TO POWER FAILURE

(57) Abstract: In some embodiment, an arrangement is provided to prevent a loss of data in a memory due to a power failure for a computing system. When the power failure occurs, any pending memory write operations may be completed and dirty cache lines may be flushed back to the memory. Subsequently, the computing system may be put into a loss-prevention state, under which power may be turned off for all components in the computing system except the memory. The memory is powered by a battery pack which includes batteries and is in a self refresh state. When the power returns, applications and operating systems running in the computing system may resume what is left out when the power supply failure occurs, based at least in part on data retained in the memory. Other embodiments are described and claimed.

PREVENTION OF DATA LOSS DUE TO POWER FAILURE

BACKGROUND

1. FIELD

The present disclosure relates generally to data preservation in a computing system and, more specifically, to prevention of data loss due to power failure.

2. DESCRIPTION

Most desktop computers and servers use a power supply with an alternating current (AC) input source and a direct current (DC) output. When AC power fails, and hence the DC power fails, data that is in most varieties of dynamic and static memory will be lost unless steps are taken to quickly store the data in a non-volatile memory. A variety of schemes have been developed to handle AC power failures. One scheme is to use an uninterruptible power supply (UPS) which continues supplying DC power to a computer when its AC power fails. However, a UPS requires a fairly large volume, large weight, and extra USB or serial-port cables to report when the battery is reaching depletion. Also the cost of a UPS may make it unattractive to many individual personal computer (PC) users.

Another scheme attempts to build a smaller and cheaper "UPS" inside the box. Such an inside-box "UPS" is smaller and cheaper compared to a traditional UPS because it can skip the AC-to-DC stage in the traditional UPS, and it does not need an extra cable, a box, or voltage regulators. However, this scheme requires the capacity of an inside-box "UPS" be large enough so that a computer may be able to copy all data in volatile memory devices (such as disks) to non-volatile memory devices after AC power fails. The inside-box "UPS" need also support a very high current drain. Inexpensive batteries are not optimized for both high drain and high capacity.

Yet another scheme may be using a non-volatile memory for main memory, but there are no existing technologies that can make this scheme close

in performance or cost to using a volatile memory (e.g., dynamic random access memory (DRAM)) for main memory. Yet another scheme may be to immediately copy content of a volatile main memory to a non-volatile main memory when AC power fails. This scheme, however, would almost double memory cost because both a volatile main memory and a non-volatile main memory are required. All of these schemes do not satisfy individual PC users because of cost, performance, and weight concerns.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present disclosure will become apparent from the following detailed description of the present disclosure in which:

Figure 1 illustrates a general computing system which may use a battery pack to prevent data loss due to power failure;

Figure 2 illustrates an example structure of a hardware virtualization environment;

Figure 3 illustrates main functional components in a computing system which may work together to prevent data loss due to power failure; and

Figure 4 illustrates a flowchart of an example process of preventing data loss due to power failure for a computing system.

DETAILED DESCRIPTION

When an AC power failure occurs to a computing system (e.g., a desktop computer and/or a server), the inductance and capacitance of the AC power supply can typically maintain the DC output power for the computing system for a short period of time after such a failure is detected and before the DC power supply becomes invalid. During this short period of time, any pending memory write operations may be completed and "dirty" cache lines, those cache lines that do not match their corresponding values in main memory, may be flushed to the main memory of the computing system. Subsequently, the computing system

may be put in a loss-prevention state under which power all components except the main memory in the computing system may be turned off. Typically, the main memory comprises volatile memory, such as DRAM, in which data needs to be periodically refreshed to prevent data losses. In the loss-prevention state, the DRAM requires a low-level of power and is able to perform a self-refresh operation to avoid loss of data.

A battery pack may be used to provide power when the computing system is in the loss-prevention state. The battery pack includes two or three AA-size batteries or batteries of other sizes/types. The battery pack may include batteries that are rechargeable or non-rechargeable. If the computing system has a write-back cache in a processor or a disk drive, data in the write-back cache may also be refreshed using power supplied by the battery pack under the loss-prevention state. When the AC power supply returns, and hence the DC power returns, the computing system may resume working based on data stored in the main memory. If the battery pack is made using non-rechargeable batteries, a warning may be given out when the batteries are near depletion. The user of the computing system may then replace the batteries. The user would also be advised not to enter any new data into the computer until the batteries had been replaced, as the data would not be adequately protected against loss in case of an AC power failure.

Reference in the specification to "one embodiment" or "an embodiment" of the disclosed techniques means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the disclosed techniques. Thus, the appearances of the phrase "in one embodiment" appearing in various places throughout the specification are not necessarily all referring to the same embodiment.

Figure 1 illustrates a general computing system 100 which may use a battery pack to prevent data loss due to power failure. Note that the details shown in the figure are not required and systems with different details may be used. Although not shown, the computing system 100 is envisioned to receive electrical power from an alternating current (AC) source (e.g., by connecting to an electrical outlet). The computing system comprises one or more processors 110

coupled to a bus 115. Processor 110 may comprise a variety of different types (e.g., For Pentium® family processors).

The computing system 100 may also include a chipset 120 coupled to the bus 115. The chipset 120 may include one or more integrated circuit packages or chips. The chipset 120 may comprise one or more device interfaces 135 to support data transfers to and/or from other components 160 of the computing system 100 such as, for example, BIOS firmware, keyboards, mice, storage devices, network interfaces, etc. The chipset 120 may be coupled to a Peripheral Component Interconnect (PCI) bus 170. The chipset set 120 may include a PCI bridge 145 that provides an interface to the PCI bus 170. The PCI Bridge 145 may provide a data path between the CPU 110 as well as other components 160, and peripheral devices such as, for example, an audio device 180 and a disk drive 190. Although not shown, other devices may also be coupled to the PCI bus 170.

Additionally, the chipset 120 may comprise a memory controller 125 that is coupled to a main memory 150. The main memory 150 may store data and sequences of instructions that are executed by the processor 110 or any other device included in the system. The memory controller 125 may access the main memory 150 in response to memory transactions associated with the processor 110, and other devices in the computing system 100. In one embodiment, memory controller 150 may be located in processor 110 or some other circuitries. The main memory 150 may comprise various memory devices that provide addressable storage locations which the memory controller 125 may read data from and/or write data to. The main memory 150 may comprise one or more different types of memory devices such as Dynamic Random Access Memory (DRAM) devices, Synchronous DRAM (SDRAM) devices, Double Data Rate (DDR) SDRAM devices, or other memory devices.

The main memory 150 may comprise volatile memory devices such as DRAM-based devices. A volatile memory device needs to be refreshed periodically to prevent loss of data stored therein. Thus, when the computing system 100 loses the AC power supply, measures need to be taken to prevent loss of data stored in a volatile memory device. The power supply (not shown in the figure) provides an indication when the AC power is failing. Such an

indication is sent to a power failure handling mechanism 130. Since the inductance and capacitance of the AC power supply may maintain the DC power supply for a short period of time after an AC power failure occurs, the power supply needs to report the AC power failure as soon as the failure occurs so that the power failure handling mechanism may take corresponding measures immediately to prevent any data loss.

Once the power failure handling mechanism 130 is notified of an AC power failure by the power supply, the mechanism may complete any pending memory write operations and flush dirty cache lines back to the main memory 150 within that short period of time during which power is maintained by the inductance and capacitance of the AC power supply. Subsequently, the power failure handling mechanism may put the computing system into a loss-prevention state, under which power to all devices except the main memory may be turned off. The main memory may be powered by a battery pack. In one embodiment, chipset 120 may have logic capable of detecting an AC power failure and notifying the power handling mechanism of the AC power failure.

Although the power failure handling mechanism 130 is shown to be inside the chipset 120, the mechanism may also involve components and/or functions of other devices (e.g., processor 110) in the computing system 100. For example, the power failure handling mechanism 130 may complete any pending memory write operations through a state machine in the processor 110. The power failure handling mechanism may flush dirty cache lines back to the main memory though system management interrupt (SMI) handlers. After the dirty cache lines is flushed back to the main memory, the SMI handlers may put the computing system into the loss-prevention state. Under the loss-prevention state, power to the main memory may be supplied by the battery pack and the memory may perform a self-refresh to maintain data. When the AC power returns, the processor 110 may resume work interrupted by the AC power failure, based on data stored in the main memory.

The computing system 100 as shown in Figure 1 may be configured to provide a hardware virtualization environment for an operating system (OS). Figure 2 illustrates an example structure of a hardware virtualization environment 200. Note that the details shown in the figure are not required and systems with

different details may be used. The platform hardware 240 comprises hardware in a computing system including devices such as chipset, memory, disk drives, and I/O devices. The platform hardware 240 is capable of executing an OS or a virtual machine monitor (VMM) such as a VMM 220. The VMM 220 provides a software layer to run on the platform hardware to enable the operation of multiple virtual machines (VMs) 210 (e.g., 210A, ..., 210N). Each VM behaves like a complete physical machine that can run its own OS, for example, guest OS 204 (e.g., 214A and 214N). One or more applications (e.g., 212A and 212N) may run on top of each guest OS. Usually, each VM is given the illusion that it is the only physical machine.

The VMM takes control of the system whenever a VM attempts to perform an operation that may affect the operations of other VMs or the hardware (e.g., a system call). The VMM will affect the operation for the VM to ensure the whole computer system is not disrupted. The VMM also has the knowledge of states of components of the platform hardware, and stores the hardware component states in the main memory (e.g., main memory 150 as shown in Figure 1). Different operating systems, or separate instances of the same operating system, may execute in each VM. Since VMs are usually isolated from each other, an OS crashing in one VM usually does not affect the other VMs. Although Figure 2 shows only one VMM, there may be other alternative implementations employing more than one VMM, for example, a VMM may be run within, or on top of, another VMM.

The VMM 220 may utilize aspects of a basic input/output system (BIOS) 230 in a computing system. The BIOS 230 comprises firmware that, when executed, controls various functions (keyboard, disk drives and display screen functions, for example) of the computing system at a basic level. In response to the computing system booting up, a processor of the computing system executes the BIOS to perform a power on self-test to locate, initialize and test devices of the computing system. The BIOS is responsible for loading the operating system. Certain BIOS functions are also used during the operation of the computing system. The BIOS image (i.e., the program code and parameter space that define the BIOS) is stored in a memory that does not lose its stored contents when power to the computing system is removed. For example, the BIOS image

may be stored in a FLASH memory, an erasable programmable read only memory (EPROM) that may be rapidly updated. Moreover, functions of the BIOS may be extended to an extensible firmware framework known as the extensible firmware interface (EFI). The EFI is a public industry specification that describes an abstract programmatic interface between platform firmware and shrink-wrap operating systems or other custom application environments. The EFI framework standard includes provisions for extending BIOS functionality beyond that provided by the BIOS code stored in a platform's BIOS device (e.g., flash memory). More particularly, EFI enables firmware, in the form of firmware modules and drivers, to be loaded from a variety of different resources, including primary and secondary flash devices, option ROMs (Read-Only Memory), various persistent storage devices (e.g., magnetic disks, optical disks, etc.), and from one or more computing systems over a computing system network.

When a computing system (e.g., the computing system 100 as shown in Figure 1) loses its AC power, any pending memory write operations may be completed and dirty cache lines may be flushed back to the main memory. Power to the main memory may be maintained using power supplied by a battery pack and the main memory be placed in a low-power self-refresh state. When the AC power returns, the hardware states before the AC power failure occurred may be retrieved from the main memory, since the VMM 220 retains states of hardware components in the computing system in the main memory. The BIOS, which helps reboot the computing system when the AC power returns, may inform the VMM that the AC power failure has occurred and instruct the VMM to retrieve the hardware states from the main memory because the VMM does not have the knowledge of the AC power failure. The VMM may further provide VMs with the retrieved hardware states. Based at least in part on the retrieved hardware states, guest OS's (e.g., 214A) as well as applications (e.g., 212A) running on top of the guest OS's may resume what processing was left when the AC power failure occurred. If the VMM does not have knowledge of states for any hardware components (e.g., those add-in cards), physical resets may be needed for those components and the VMM should be able to handle such resets.

Figure 3 illustrates main functional components in a computing system 300 which may work together to prevent data loss due to power failure. Note that the

details shown in the figure are not required and systems with different details may be used. Compared to the computing system 100 as shown in Figure 1, the computing system 300 comprises similar components such as one or more processors 110 coupled to a bus 115 and a chipset 120 also coupled to the bus 115. Similarly, the chipset 120 comprises a memory controller 125 and a power failure handling mechanism 130. In addition to these similar components compared to Figure 1, Figure 3 shows more detailed components related to data loss prevention due to power failure. For example, Figure 3 shows that the computing system 300 includes a power supply 330 to supply power to the computing system. The power supply 330 receives AC power through a power cable connecting to an electrical power outlet. The power supply 330 converts the received AC power to direct current (DC) power, regulates the DC voltage, and supplies the DC power to the processor, the chipset, the main memory, and other components in the computing system 300. Additionally, inductance and capacitance associated with circuitry that performs AC-to-DC conversion and DC voltage regulation may help maintain the output DC voltage of the power supply 330 for a short period of time even after the AC power supply is lost. How long this short period of time can be depends on current drawn from the power supply as well as the inductance and capacitance of the power supply 330.

The power supply may comprise power control logic 335 to immediately detect an AC power loss and inform the power failure handling mechanism 130 of the loss. Subsequently, the power handling mechanism may use a state machine in the processor 110 to complete any pending memory write operations. In the meanwhile, the power handling mechanism may trigger an SMI and flush dirty cache lines back to the main memory through an SMI handler or the state machine. Typically, completing pending memory write operations and flushing dirty cache lines to the main memory may be completed during the short period of time when the output DC voltage is maintained by the inductance and capacitance of the power supply 330. After pending memory write operations are completed and dirty cache lines are flushed back to the main memory, the power failure handling mechanism 130 may trigger another SMI to put the computing system in a loss-prevention state.

Under the loss-prevention state, all components in the computing system 300 may be turned off power except the main memory 150, which will be powered by a battery pack 370. The battery pack only needs to supply enough power to keep data in the main memory refreshed so that the data is not lost. The battery pack 370 may include two or three AA-sized batteries. The battery pack may also include batteries of other types or sizes, and may use either rechargeable or non-rechargeable batteries. Output power voltage of the battery pack may be further regulated by a voltage regulator 360. If a disk drive (not shown) or a processor in the computing system 300 includes a write-back cache, the battery pack may also provide limited power to the write-back cache so that data stored therein can be retained while the AC power is not present. The battery pack 370 may be located inside a case of the computing system. A small panel door may be provided on the case so that a user of the computing system may remove depleted batteries and install fresh batteries inside the battery pack. If the battery pack uses rechargeable batteries, batteries inside the battery pack may be recharged, if necessary, whenever the AC power is present.

Additionally, the computing system 300 may comprise a first isolation circuitry 340, a second isolation circuitry 350, and a third isolation circuitry (not shown). When the computing system is in the loss-prevention state, the first isolation circuitry 340 may prevent current from the battery pack 370 from flowing into the power supply 330 while the second isolation circuitry 350 may let the battery pack 370 provide power to the main memory 150. When the computing system is not in the loss-prevention state (e.g., power supply 330 has AC power supply), the first isolation circuitry 340 may let the power supply 330 provide power to the main memory, while the second isolation circuitry 350 may prevent current from the power supply 330 from flowing into the voltage regulator 360 and the battery pack 370. Also when the computing system is not in the loss-prevention state, the second isolation circuitry 350, when informed by the power control logic 335, may prevent the battery pack 370 from providing any power to the main memory 150. The first isolation circuitry 340 may be a part of or separate from the power supply 330, and the second isolation circuitry 350 may be a part of or separate from the voltage regulator 360. The third isolation circuitry (not shown) may be located between the memory controller 125 and the

main memory 150 to prevent the battery pack from supplying power to the memory controller when the system is in the loss-prevention state. The third isolation circuitry may be a part of or separate from the memory controller 125.

The chipset 120 comprises a real-time clock (RTC) well 310 to keep track of the time even when the computing system loses the AC power or is turned off. The RTC well is powered by a RTC battery 320, which is independent from the power supply 330 and the battery pack 370. The RTC well may comprise a counter (not shown) to count the amount of time, such as hours, that the main memory 150 has been powered by the battery pack 370. When the amount of time counted reaches a predetermined threshold, the RTC well may cause an alert sending mechanism 315, coupled to the RTC well, to send out a warning signal so that a user of the computing system may change batteries for the battery pack. The predetermined time threshold depends on specifications of the batteries used. It is estimated that a pack of 3 AA alkaline batteries could provide approximately 75 hours of backup for a 1 GByte memory (typically an AA alkaline battery provides about 2.5 amp-hours of energy and a 512 MByte dual in-line memory module (DIMM) draws about 50 mAmp in its low-power self refresh state). In one embodiment, the alert sending mechanism 315 may be integrated with the RTC well. In another embodiment, the alert sending mechanism 315 may be implemented by a circuitry which is separate from the RTC well.

In one embodiment, to avoid any data loss in the main memory when changing batteries in the battery back while the computing system is in the loss-prevention state, the battery back may comprise space for backup batteries so that new batteries may be placed in the battery pack before the depleted batteries are removed. In the same embodiment or in another embodiment, the computer system may provide an indication of the capacity level of the batteries in the battery pack, either on a screen or through lighting signals, when the AC power supply is present. If the indication shows that the capacity of the batteries is low (even though it is not low enough to trigger an alert signal), a user of the computing system may decide to change the batteries if the batteries are non-rechargeable.

Figure 4 illustrates a flowchart of an example process of preventing data loss due to power failure for a computing system. For the convenience of

description, the example process will be described using the computing system 300 as shown in Figure 3, although this example process applies to any other computing systems. The process starts with block 405 where an AC power failure occurs. Assume that a hardware virtualization environment runs on the computing system, states of hardware components in the computing system 300 just before the AC power failure occurs may be stored in the main memory 150. In block 410, the AC power failure may be detected by the power control logic 335. In block 415, a state machine in the processor 110 may be activated and an SMI may be triggered. In block 420, any pending memory write operations may be completed and dirty cache lines may be flushed back to the main memory 150 by the state machine and/or an SMI handler. Subsequently in block 425, the computing system may be put into a loss-prevention state. In block 430, all power planes (for all components in the computing system) may be turned off except the main memory and write-back caches in a processor or a disk drive if such write-back caches exist, which are in self refresh mode and powered by the battery pack 370.

In block 435, the AC power may be monitored to check if it returns. If the AC power returns before the battery pack is depleted, applications and guest OS's running on the computing system may resume what was left when the AC power failure occurred in block 440, based at least in part on the hardware component states retained by the VMM and other data stored in the main memory. If the state of a hardware component is not known to the VMM, this hardware component may be reset when the AC power returns. If the AC power does not return, in block 445 when batteries in the battery pack 370 needs to be changed may be detected. If it is determined that the batteries in the battery pack need to be changed in block 445, an alert signal may be sent out to a user of the computing system to change the batteries to avoid any data loss in the main memory and/or write-back caches in block 450; otherwise, the process goes back to block 435. Fresh batteries may be placed into the battery pack before the depleted batteries are removed to avoid any data loss in the main memory and/or write-back caches while changing batteries.

Although techniques to prevent memory data loss due to AC power failure are described in the context of a hardware virtualization environment, a person of

ordinary skill in the art can readily appreciate that the disclosed techniques will be also be adapted to a computing system without running a hardware virtualization environment.

Although an example embodiment of the disclosed techniques is described with reference to diagrams in Figures 1-4, persons of ordinary skill in the art will readily appreciate that many other structures and methods of implementing the present invention may alternatively be used. For example, the order of execution of the functional blocks or process procedures may be changed, and/or some of the structures, functional blocks or process procedures described may be changed, eliminated, or combined.

In the preceding description, various aspects of the disclosed techniques have been described. For purposes of explanation, specific numbers, systems and configurations were set forth in order to provide a thorough understanding of the present disclosure. However, it is apparent to one skilled in the art having the benefit of this disclosure that the present disclosure may be practiced without the specific details. In other instances, well-known features, components, or modules were omitted, simplified, combined, or split in order not to obscure the present disclosure.

Embodiments of the present techniques described herein may be implemented in circuitry, which includes hardwired circuitry, digital circuitry, analog circuitry, programmable circuitry, and so forth. They may also be implemented in computer programs. Such computer programs may be coded in a high level procedural or object oriented programming language. However, the program(s) can be implemented in assembly or machine language if desired. The language may be compiled or interpreted. Additionally, these techniques may be used in a wide variety of networking environments. Such computer programs may be stored on a storage media or device (e.g., hard disk drive, floppy disk drive, read only memory (ROM), CD-ROM device, flash memory device, digital versatile disk (DVD), or other storage device) readable by a general or special purpose programmable processing system, for configuring and operating the processing system when the storage media or device is read by the processing system to perform the procedures described herein. Embodiments of the disclosure may also be considered to be implemented as a machine-readable

storage medium, configured for use with a processing system, where the storage medium so configured causes the processing system to operate in a specific and predefined manner to perform the functions described herein.

While the disclosed techniques have been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the disclosure, which are apparent to persons skilled in the art to which the disclosure pertains are deemed to lie within the spirit and scope of the disclosure.

CLAIMS

What is claimed is:

1. A method for preventing data loss due to power failure in a computing system, comprising:

detecting a power failure in the computing system;

completing pending memory write operations and flushing dirty cache lines to a memory after the power failure is detected; and

putting the computing system in a loss-prevention state after the pending write operations are completed and the dirty cache lines are flushed to the memory.

2. The method of claim 1, wherein the memory comprises volatile memory, the volatile memory including write-back caches in at least one of a processor and a disk drive.

3. The method of claim 1, wherein the power failure comprises a failure of an alternating current (AC) power;

4. The method of claim 1, further comprising triggering at least one of a state machine and a system management interrupt (SMI) to complete the pending memory write operations and to flush the dirty cache lines to the memory, after the power failure is detected.

5. The method of claim 1, further comprising turning off power supply for all components in the computing system except the memory when the computing system is put in the loss-prevention state.

6. The method of claim 5, further comprising restoring a status of the computing system just before the power failure occurs based on data preserved in the memory when power returns, the preserved data including hardware states just before the power failure occurs.

7. The method of claim 6, wherein the hardware states are retained by a virtual machine monitor (VMM) in the memory, the VMM providing an operating system with information on hardware in the computing system.

8. The method of claim 7, wherein the VMM restores the hardware states to states just before the power failure occurs and provides the restored hardware states to the operating system, when the power returns.

9. The method of claim 1, wherein the loss-prevention state comprises a state under which data in the memory is refreshed using batteries, the batteries comprising at least two AA-sized batteries.

10. The method of claim 9, further comprising:

detecting when the batteries need to be changed; and

sending an alert to a user of the computing system to change the batteries when the batteries need to be changed.

11. An apparatus for preventing data loss due to power failure in a computing system, comprising:

a memory to store data accessible by components in the computing system;

power control logic to detect a power failure;

a power failure handling mechanism to complete pending memory write operations and to flush dirty cache lines to the memory, when informed by the power control logic of the power failure; and

a battery pack to supply power to the memory to refresh the data in the memory to prevent data loss in the memory, after the pending memory write operations are completed and the dirty cache lines are flushed to the memory.

12. The apparatus of claim 11, wherein the memory comprises volatile memory, the volatile memory including write-back caches in at least one of a processor and a disk drive.

13. The apparatus of claim 11, wherein the battery pack comprises at least two AA-sized batteries.

14. The apparatus of claim 11, wherein the power failure comprises a failure of an alternating current (AC) power.

15. The apparatus of claim 11, wherein the battery pack supplies power only to the memory to refresh the data in the memory, after the pending memory write operations are completed and the dirty cache lines are flushed to the memory.

16. The apparatus of claim 11, further comprising:

a memory controller, coupled to the memory, to handle data traffic to and from the memory;

a first isolation circuitry to prevent current from the battery pack from flowing into a power supply when the battery pack is used to supply power to the memory after the power failure occurs, the power supply including an alternating current (AC) power supply;

a second isolation circuitry to prevent current from the power supply from flowing into the battery pack when the power supply is used to supply power to the memory and to prevent the battery pack from providing power to the memory when the power supply is present; and

a third isolation circuitry to prevent the battery pack from supplying power to the memory controller, when the battery pack is used to supply power to the memory.

17. The apparatus of claim 11, further comprising a voltage regulator to regulate output power voltage provided by the battery pack.

18. The apparatus of claim 11, further comprising:

a real-time clock well, the real-time clock well including a counter to count time during which the memory is powered by the battery pack; and

an alert sending mechanism, coupled to the real-time clock well, to send out a warning of replacing batteries in the battery pack when the counted time exceeds a pre-determined limit, the pre-determined limit set based on specifications of the batteries used in the battery pack.

19. The apparatus of claim 11, wherein the power failure handling mechanism completes the pending write operations and flushes the dirty cache lines to the memory by using at least one of a state machine and at least one system management interrupt (SMI) handler.

20. The apparatus of claim 11, wherein the power failure handling mechanism puts the computing system in a loss-prevention state after completing the pending write operations and flushing the dirty cache lines to the memory.

21. The apparatus of claim 20, wherein the loss-prevention state comprises a state under which data in the memory is refreshed using power provided by the battery pack.

22. The apparatus of claim 20, wherein the power failure handling mechanism turns off power to all components in the computing system except the memory, when the computing system is put in the loss-prevention state.

23. A computing system, comprising:

at least one processor;

a power supply to supply power to components in the computing system;

a memory to store data accessible by the at least one processor; and

a battery pack to supply power to the memory to refresh the data in the memory after a power failure occurs.

24. The computing system of claim 23, wherein the memory comprises volatile memory, the volatile memory including write-back caches in at least one of a processor and a disk drive.

25. The computing system of claim 23, wherein the battery pack comprises at least two AA-sized batteries.

26. The computing system of claim 23, wherein the power supply comprises an alternating current (AC) power supply.

27. The computing system of claim 23, further comprising:

power control logic to detect the power failure; and

a power failure handling mechanism to complete pending memory write operations and to flush dirty cache lines to the memory by using at least one of a state machine and at least one system management interrupt (SMI) handler, when informed by the power control logic of the power failure.

28. The computing system of claim 27, wherein the power failure handling mechanism puts the computing system in a loss-prevention state after completing the pending write operations and flushing the dirty cache lines to the memory, the loss-prevention state comprising a state under which data in the memory is refreshed using power provided by the battery pack.

29. The computing system of claim 27, wherein the power failure handling mechanism turns off power to all components in the computing system except the memory, when the computing system is put in the loss-prevention state.

30. The computing system of claim 23, further comprising:

a memory controller, coupled to the memory, to handle data traffic to and from the memory;

a voltage regulator to regulate output power voltage provided by the battery pack;

a first isolation circuitry to prevent current from the battery pack from flowing into the power supply when the battery pack is used to supply power to the memory after the power failure occurs;

a second isolation circuitry to prevent current from the power supply from flowing into the battery pack when the power supply is used to supply power to the memory and to prevent the battery pack from providing power to the memory when the power supply is present; and

a third isolation circuitry to prevent the battery pack from supplying power to the memory controller, when the battery pack is used to supply power to the memory.

31. The computing system of claim 23, further comprising:

a real-time clock well, the real-time clock well including a counter to count time during which the memory is powered by the battery pack; and

an alert sending mechanism, coupled to the real-time clock well, to send out a warning of replacing batteries in the battery pack when the counted time exceeds a pre-determined limit, the pre-determined limit set based on specifications of the batteries used in the battery pack.

32. The computing system of claim 23, further comprising a virtual machine monitor (VMM) to provide at least one operating system with information on hardware in the computing system, and to retain hardware states in the memory.

33. The computing system of claim 32, wherein the VMM restores the hardware states to states just before the power failure occurs and provides the restored hardware states to the operating system, when the power returns.

34. The computing system of claim 32, wherein the computing system resumes what processing is left when the power failure occurs based at least in part on data preserved in the memory during the power failure, the preserved data including the hardware states just before the power failure occurs.

35. An article comprising a storage medium having stored thereon instructions that, when accessed by a processor result in the following:

detecting a power failure in the computing system;

completing pending memory write operations and flushing dirty cache lines to a memory after the power failure is detected; and

putting the computing system in a loss-prevention state after the pending write operations are completed and the dirty cache lines are flushed to the memory, the loss-prevention state comprises a state under which data in the memory is refreshed using batteries, the memory including volatile memory, the volatile memory including write-back caches in at least one of a processor and a disk drive.

36. The article of claim 35, wherein the power failure comprises a failure of an alternating current (AC) power.

37. The article of claim 35, wherein the instructions when accessed by the processor also result in:

triggering at least one of a state machine and a system management interrupt (SMI) to complete the pending memory write operations and to flush the dirty cache lines to the memory, after the power failure is detected; and

turning off power supply for all components in the computing system except the memory when the computing system is put in the loss-prevention state.

38. The article of claim 37, wherein the instructions when accessed by the processor further result in restoring a status of the computing system just before the power failure occurs based on data preserved in the memory when power returns, the preserved data including hardware states just before the power failure occurs.

39. The article of claim 38, wherein the hardware states are retained by a virtual machine monitor (VMM) in the memory, the VMM restoring the hardware states to states just before the power failure occurs and providing the restored hardware states to an operating system, when the power returns.

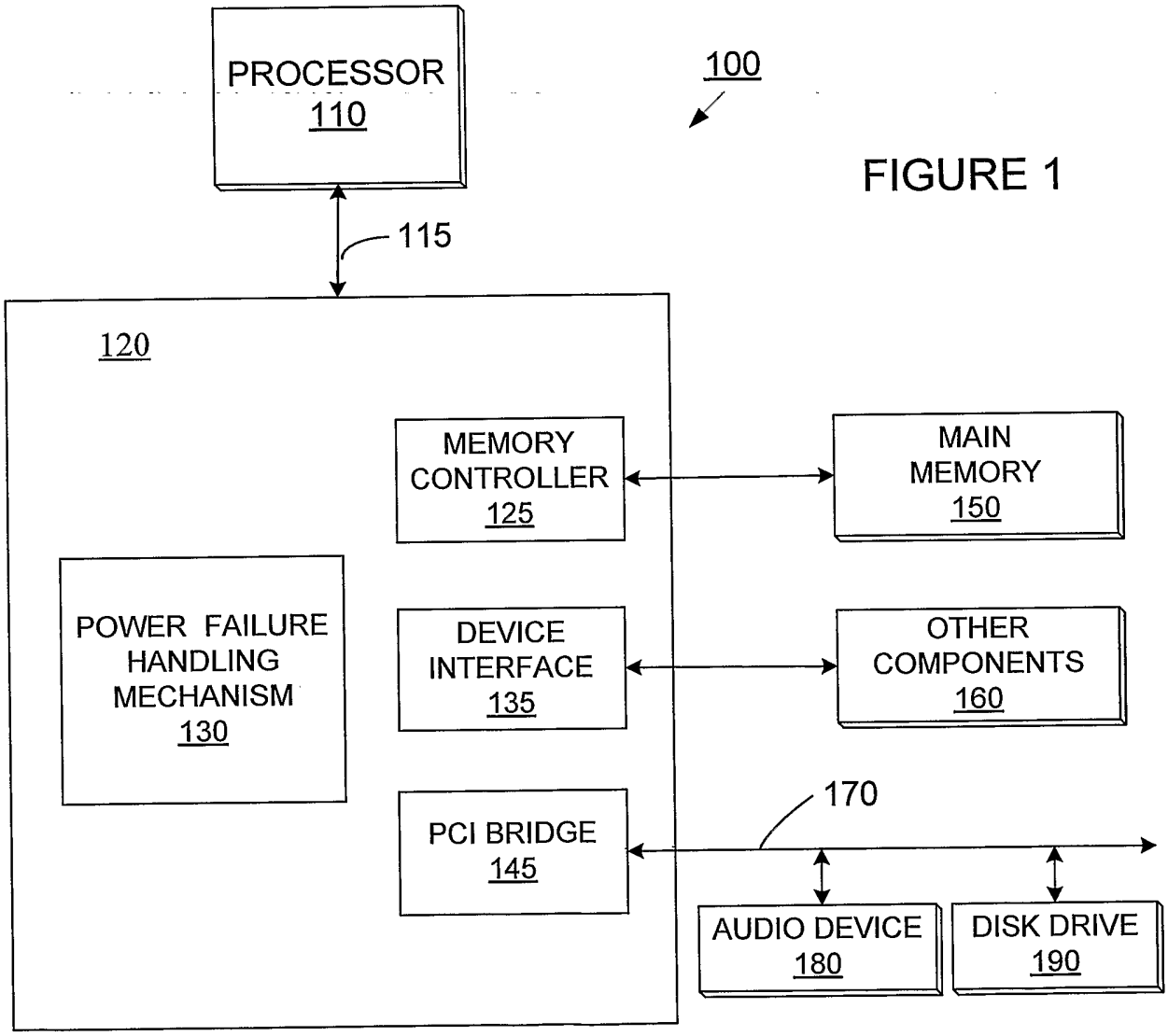

PROCESSOR
110
100
115
120
MEMORY
CONTROLLER
125
MAIN
MEMORY
150
POWER FAILURE
HANDLING
MECHANISM
130
DEVICE
INTERFACE
135
OTHER
COMPONENTS
160
170
PCI BRIDGE
145
AUDIO DEVICE
180
DISK DRIVE
190

FIGURE 1

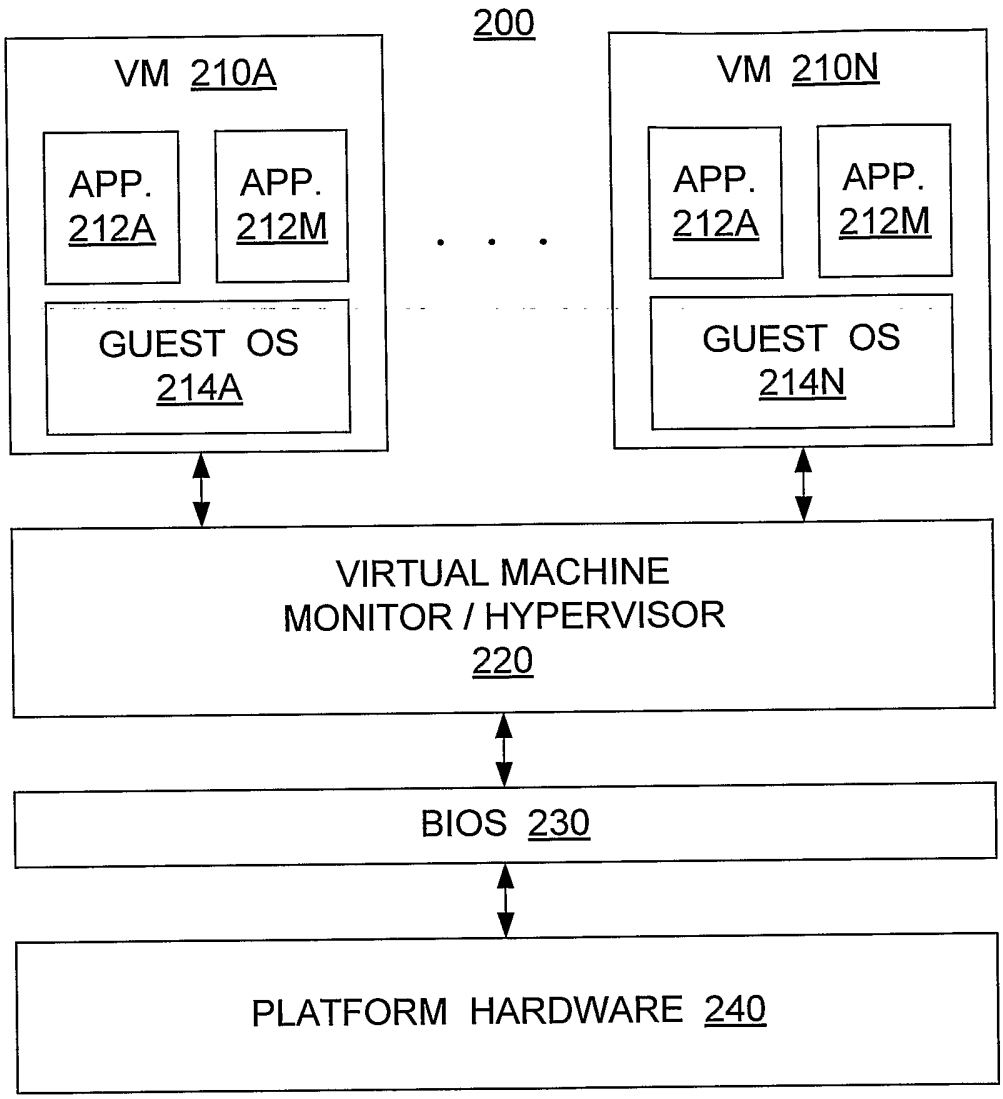
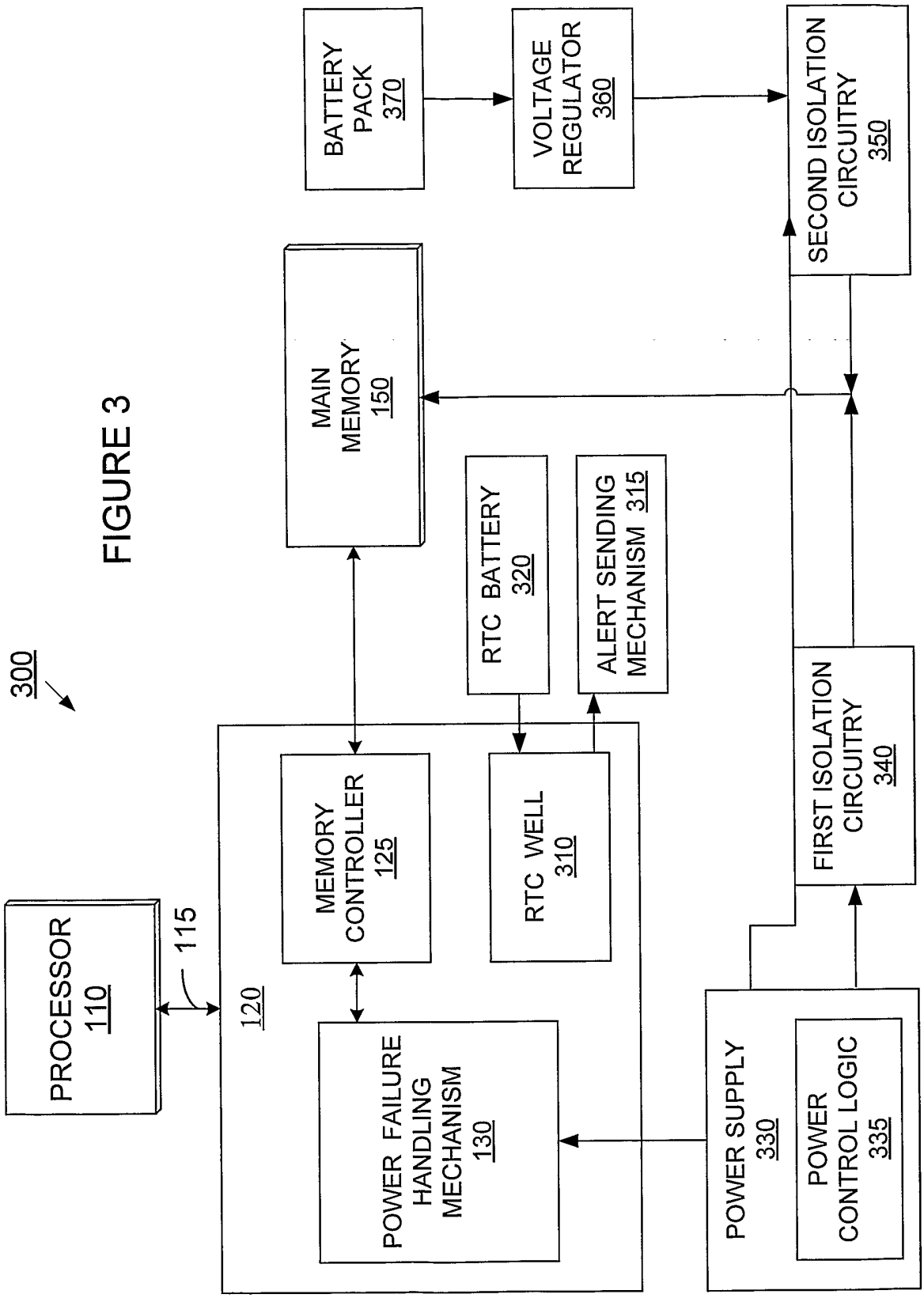

200
VM 210A
APP. 212A
APP. 212M
GUEST OS 214A
VM 210N
APP. 212A
APP. 212M
GUEST OS 214N
VIRTUAL MACHINE MONITOR / HYPERVISOR 220
BIOS 230
PLATFORM HARDWARE 240

FIGURE 2

FIGURE 3


300
PROCESSOR 110
115
120
MEMORY CONTROLLER 125
RTC WELL 310
POWER FAILURE HANDLING MECHANISM 130
MAIN MEMORY 150
RTC BATTERY 320
ALERT SENDING MECHANISM 315
BATTERY PACK 370
VOLTAGE REGULATOR 360
SECOND ISOLATION CIRCUITRY 350
FIRST ISOLATION CIRCUITRY 340
POWER SUPPLY 330
POWER CONTROL LOGIC 335

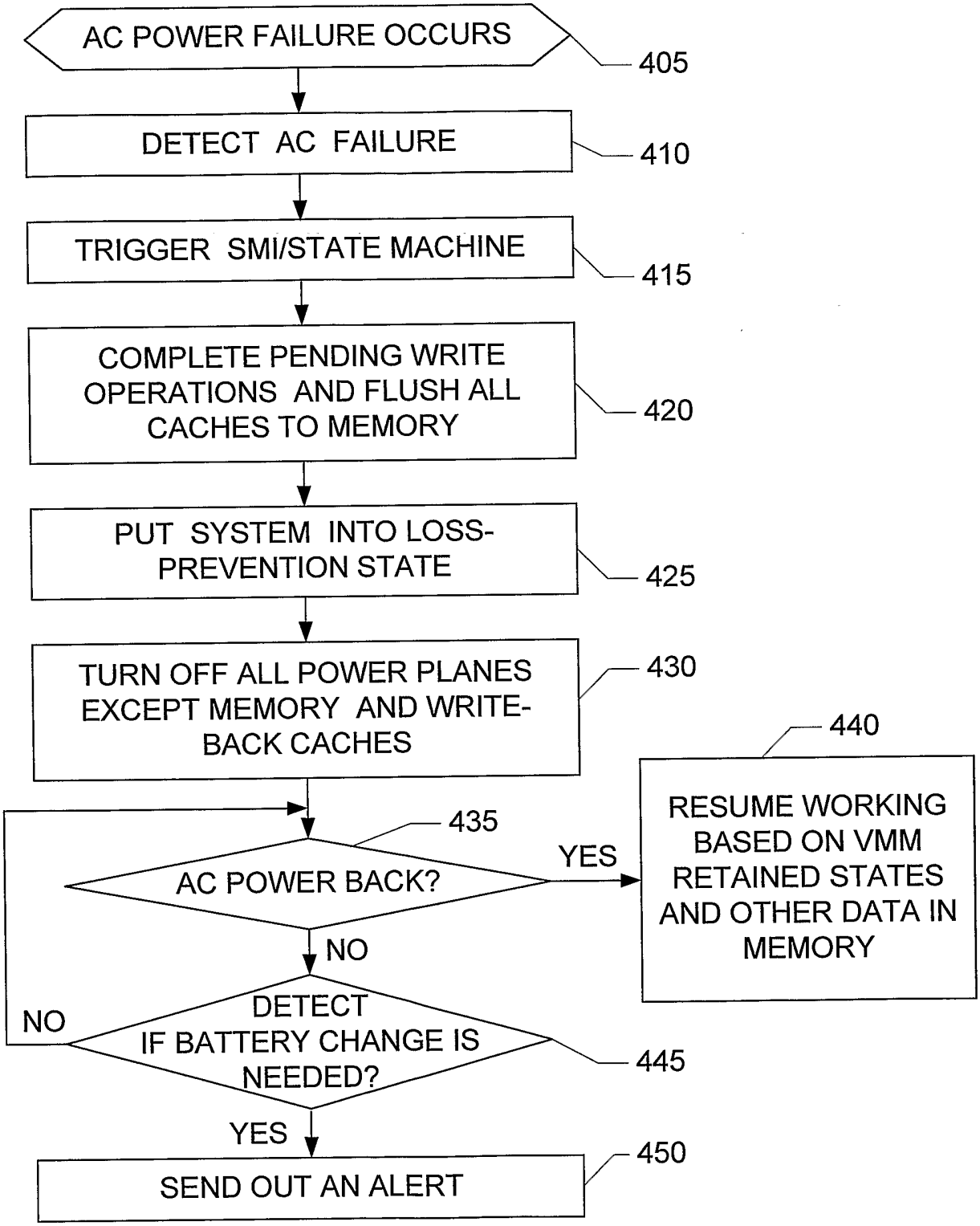

AC POWER FAILURE OCCURS
405
DETECT AC FAILURE
410
TRIGGER SMI/STATE MACHINE
415
COMPLETE PENDING WRITE OPERATIONS AND FLUSH ALL CACHES TO MEMORY
420
PUT SYSTEM INTO LOSS-PREVENTION STATE
425
TURN OFF ALL POWER PLANES EXCEPT MEMORY AND WRITE-BACK CACHES
430
440
RESUME WORKING BASED ON VMM RETAINED STATES AND OTHER DATA IN MEMORY
435
AC POWER BACK?
YES
NO
NO
DETECT IF BATTERY CHANGE IS NEEDED?
445
YES
SEND OUT AN ALERT
450

FIGURE 4