Title: FAST PLATFORM HIBERNATION AND RESUMPTION FOR COMPUTING SYSTEMS

Abstract: Fast platform hibernation and resumption for computing systems. An embodiment of an apparatus includes a volatile system memory, a nonvolatile memory, and a processor to operate according to an operating system, the processor to transition the apparatus to a first reduced power state upon receipt of a request, the transition to the first reduced power state including the processor to store context information for the computer in the volatile system memory. The apparatus further includes logic to transition the apparatus to a second reduced power state, the logic to copy the context data from the volatile system memory to the nonvolatile memory for the transition to the second reduced power state, where copying of the context data includes the logic to scan the volatile system memory to locate non-active memory elements in the volatile system memory, eliminate the non-active memory elements from the volatile system memory to generate compressed context data, and store the compressed context data in the nonvolatile memory.
FAST PLATFORM HIBERNATION AND RESUMPTION OF COMPUTING SYSTEMS

TECHNICAL FIELD

[0001] Embodiments of the invention generally relate to the field of computing systems and, more particularly, to fast hibernation and resumption of computing systems.

BACKGROUND

[0002] Computing system may be transferred between various power states. In general, each power state provides for the powering down of certain elements of the computing system during period of inactivity. Lower states generally provide further power savings, but also required additional time to return to operation.

[0003] For example, power states may include state referred to as S-states, including S3 and S4. S3, sometimes referred to as Standby, Sleep, or Suspend to RAM, is a sleep state in which the operating system (OS) of a computing system saves the context of the system into physical memory (dynamic random access memory (DRAM)) and puts the computing system into a suspend state. In this operation, open documents and programs (applications) (or a portion thereof) that were used at the time of entering into S3 are also saved in DRAM during the suspend state. Further, contents of some chipset registers may also be written to DRAM. The physical memory DRAM is may be referred to as main memory or system memory. During the S3 state, power is removed from the platform hardware, with the exception of the DRAM and a small amount of circuitry used to later wake the system. The S3 power state provides a relatively fast suspend and resume (wake) time due to its ability to save and restore OS context and previously used programs and documents from hi-speed DRAM memory.

[0004] S4, sometimes referred to as Hibernate, Safe Sleep, or Suspend to disk, provides that the OS context and open documents and programs (or a portion thereof) are saved on a hard disk drive (HDD) rather than in fast DRAM memory. This allows for higher power savings than the S3 state because the DRAM is not kept powered. However, there are higher latencies due to slow read and write
access times of the HDD. Typical S4 hibernate and resume times are in the order of 10's of seconds.

[0005] Computer platform hibernation and resumption is described in, for example, U.S. Patent No. 7,971,071, "System and Method for Fast Platform Hibernate and Resume".

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Embodiments of the invention are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

[0007] Figure 1 illustrates embodiments of processes for a computer platform to enter into a fast hibernate state and to exit the fast hibernate state;

[0008] Figure 2 is an illustration of the transformation of data by an embodiment of an apparatus, system, or process for fast platform hibernation and resumption;

[0009] Figure 3 is an illustration of an embodiment of a memory scrubbing operation in a fast platform hibernation and resumption apparatus, system, or method;

[0010] Figure 4 is an illustration of an embodiment of memory processing in a fast platform hibernate apparatus, system, or method;

[0011] Figure 5 is an illustration of an embodiment of resumption of a computing system from a fast platform hibernation process;

[0012] Figure 6 is a flowchart to illustrate an embodiment of a process for fast platform hibernation of a computing system;

[0013] Figure 7 is a flowchart to illustrate an embodiment of a process for fast platform resumption of computing system placed in a hibernate state; and

[0014] Figure 8 illustrates an embodiment of a computing system that provides for fast platform hibernation and resumption.
DETAILED DESCRIPTION

[0015] Embodiments of the invention are generally directed to fast platform hibernation and resumption for computing systems.

[0016] In some embodiments, an apparatus, system, or method (such as a computing apparatus, system, or method) provides an enhanced process for fast platform hibernation and resumption. In some embodiments, an apparatus, system, or process includes mechanism to improve the platform hibernation through efficient handling of context data, including elimination of zero page data in hibernation. In some embodiments, an apparatus, system, or process further improves platform resumption through zeroing of memory to return zero pages in the reloading of context data. In some embodiments, the zeroing of memory including use of a hardware element to provide improved initialization of zeros in memory.

[0017] In some embodiments, an apparatus, system or method may include: detecting and calculating OS active DRAM memory physical pages; building a filter (referred to as a zero page filter)/bitmap to identify non-active (zero) DRAM pages or other memory elements; initialization of DRAM to zeros via a hardware or software mechanism; efficient processing of direct memory access stream using the zero page filter/bitmap; and application of runtime OS mechanism to flush OS dirty DRAM pages.

[0018] Conventional computing systems face additional issues the amount of memory, and thus the amount of context data to be saved, increases. Such systems do not address the performance implication that results for large DRAM memory size. In some embodiments, an apparatus, system, or method provides a smart mechanism to trim the size of DRAM pages that needs to be saved to and restored from nonvolatile memory, thus providing significant improvement to the performance of hibernate/shutdown and resume processes.

[0019] In some embodiments, an apparatus, system, or method includes two parts: A first mechanism or process to transparently and efficiently store system memory data onto a persistent storage medium; and a second mechanism or process to efficiently recover the system memory data from the storage medium and replace it back into the system memory.
In some embodiments, an apparatus, system, or process provides for a fast platform hibernate and resume apparatus or process, including the copying of context information from volatile system memory to nonvolatile memory (which may be referred to as NVRAM, or nonvolatile random-access memory) as a part of the transition to the lower power states. In some embodiments, the apparatus or process includes the copying of the stored context information back to volatile system memory upon a transition back to an operational state from the hibernate state. In some embodiments, the apparatus, system, or process provides for determining the location of each portion of a memory that is unused and contains zero information. In some embodiments, the device or system provides for elimination of all zero elements of the memory in the copying of context information.

In some embodiments, an apparatus, system, or process includes a memory scrubber, wherein the memory scrubber operates to clear or zero elements of a DRAM memory. In some embodiments, the memory scrubber allows for zeroing elements of a memory more quickly than software can provide a zeroing function. In some embodiments, the memory scrubber is utilized in the resumption of operation of the system or system from a hibernate state to zero out all elements of initialized volatile memory prior to restoring the platform context from nonvolatile memory. In some embodiments, an apparatus, system, or method allows for improved operation in fast platform hibernate and resume by reducing the amount of data that is required to be stored and recovered, by the elimination of zero data elements and by the zeroing of the volatile memory on resume to allow for quicker reloading of zero data elements into the volatile memory.

In some embodiments, a device or system further includes a section of memory for quick access in a hibernate state. In some embodiments, a section of memory may include, for example, a calendar or data that may require quick access when the system or device is in a low power state.

In some embodiments, a storage process is initiated by power management logic or BIOS. In some embodiments, the storage process may be independently initiated or be initiated at the request of the operating system. In some embodiments, a hibernation process for a platform includes:
(a) The operating system initiates a transition to a first reduced power state, such as a Sleep (S3) transition, by saving the current state of the volatile system DRAM.

(b) In some embodiments, as a part of a fast hibernation system, an operating system agent, such as a fast hibernation service or a driver, may optionally flush parts of OS memory to disk, thus making the OS DRAM active memory footprint as small as possible.

(c) A hardware processor, such as the central processing unit (CPU) or an embedded processor (EP), or system software, such as the binary input-output system (BIOS) or firmware (FW), copies OS DRAM memory contents to nonvolatile storage location, either directly or via indirect means such as device direct memory access (DMA).

(d) In some embodiments, upon receiving an OS low power entry request (such as Standby-S3/S0ix), a user level application or a driver operates to flush dirty and partial active memory pages to disk while converting freed up DRAM memory pages to zeros. In some embodiments, this operation provides for decreasing the overall size of DRAM active (non-zero) memory pages, while increasing the size of zero page pool.

(e) The operating system may indicate an intention to transition to a second reduced power state, such as a hibernate state, where a computing system in the second reduced power state consumes less power than the computing system in the first reduce power state. For example, the operating system may operate to set certain bits (the SLP_TYP and SLP_EN bits) in the chipset to indicate to the chipset hardware that the wants to transition to hibernate state.

(f) In some embodiments, the operating system may use a mechanism provided by system hardware to enter into the first reduced power state, such as Standby S3/S0ix state.

(g) Upon receiving the sleep event, system hardware may pass control to power management logic, BIOS, or firmware, which then scans the system memory for active and inactive memory, and builds a zero page filter (such as a bitmap) of non-active (zero) pages or other memory elements.
[0031] (h) In some embodiments, the power management logic, BIOS, or firmware then uses the zero page filter to discard unused DRAM memory pages from the DMA stream, while copying the active (non-zero) pages to nonvolatile memory.

[0032] (i) In some embodiments, the zero page filter is then stored in the nonvolatile memory for use in resumption of operation.

[0033] (j) In some embodiments, the power management logic, BIOS, or firmware puts the system into an S4/S5 sleep state. In some embodiments, the power management logic or BIOS can optionally put the system in a complete mechanical off (G3) state.

[0034] Upon receiving a wake event, such as a power button signal, a real-time clock (RTC) alarm, or other wake event, the power management logic or BIOS initiates a restore process. In some embodiments, a restore process for a platform includes:

[0035] (a) Upon the occurrence of a wake event, power management logic, BIOS, or firmware powers on prior to other elements of the system, and operates to initializes DRAM, and starts scrubbing the entire DRAM with zeros. Upon the entire DRAM memory having been scrubbed with zeros, the power management logic, BIOS, or firmware commences to restore the active (non-zero) DRAM pages from the NVRAM using previously stored zero page filter.

[0036] (b) The power management logic, BIOS, or firmware proceeds to perform a system restore (such as an S3 resume) according to normal procedure.

[0037] In some embodiments, an apparatus, system, or method provides for fast platform hibernation in which a system context is written into a nonvolatile memory (a smaller nonvolatile memory or NVRAM), which has a smaller capacity than another nonvolatile memory (the larger nonvolatile memory or hard disc drive) that is used to store at least one operating system, programs, and data. In a resume from fast hibernate, the system context is read from the smaller nonvolatile memory and used to restore operation of a computer system. In some embodiments, the system context is transferred entirely from DRAM system memory to the smaller nonvolatile memory and on resume transferred entirely from the smaller nonvolatile memory to the DRAM system memory. In other embodiments, the system context
can come partly from other memory such as chipset registers and be written back directly to those registers rather than go indirectly through the DRAM.

[0038] In some embodiments, in the fast hibernate process, the transfer from the DRAM system memory to the smaller nonvolatile memory occurs even after processor cores and other system components such as a hard drive and display screen are powered down. This allows the user of the computer to have the perception that the computer system is shut down quickly, even though the transfer of the system context has not been completed. Accordingly, embodiments improve the user perception of the system responsiveness. In some embodiments, the system responsiveness is further improved by the reduction in memory storage through the elimination of zero pages from storage, such that the memory shuts down more quickly.

[0039] In some embodiments, a fast hibernate process utilizing an operating system that is designed for S3 (a first reduced power state) and S4 (a second reduced power state) states, but not specifically designed for the fast hibernate process. This may occur by having the BIOS and/or another mechanism respond to a suspend to RAM (S3) command by putting the processor into a system management mode (SMM), and controlling the transfer from the DRAM system memory to the smaller nonvolatile memory and then changing a sleep type to hibernate state. In these embodiments, the process may be transparent to the operating system.

[0040] Figure 1 illustrates embodiments of processes for a computer platform to enter into a fast hibernate state and to exit the fast hibernate state. In this illustration, the fast hibernate state entry 110 commences with an operating system at an operational (SO) 115 state transitioning to a sleep (S3) state, including the storing of context data to volatile DRAM memory 120. In some embodiments, the fast hibernate process includes identification of zero data pages. The fast hibernate process further includes copying non-zero data pages of the context information from the DRAM, with the non-zero data and a zero page filter stored in NVRAM 125. The storage to nonvolatile memory, allows for transition of the platform to a hibernate (S4) state or, optionally, a zero power state 130.
In some embodiments, the fast hibernate exit process 160 begins with S4 or zero power state 165. In some embodiments, upon waking of logic such as power management logic or BIOS, the DRAM 175 is scrubbed with zero values, and non-zero pages are restored from the NVRAM 170 to the DRAM 175 using the stored zero page filter. The loading of the context into the DRAM 175 then puts the data in the proper location for restoring the context using the sleep state (S3) processes, thereby returning the operating system 180 to the operational (SO) state.

Figure 2 is an illustration of the transformation of data by an embodiment of an apparatus, system, or process for fast platform hibernation and resumption. In some embodiments, in a fast hibernate process, a non-filtered FFS (flash file system) DMA table 220, representing a system context stored to DRAM for transition to a sleep state, may be scanned for zero pages. The scanning results in a zero page filter 250 representing the locations of the active/non-zero pages and the non-active/zero pages. In some embodiments, the zero pages are then discarded 255. In some embodiments, the fast hibernate process then results in a filtered FFS DMA table 270, where the filtered table includes non-zero pages and empty entries.

Figure 3 is an illustration of an embodiment of a memory scrubbing operation in a fast platform hibernation and resumption apparatus, system, or method. In some embodiments, an exit from fast hibernation includes initiation of the DRAM prior to re-loading the data from the nonvolatile memory. However, the DRAM memory 320 will contain random one and zero values as the states of memory elements when returned to power are generally unstable. The restoration of data from the nonvolatile memory will only restore the values of non-zero elements, and thus the zero elements will contain random data if no action is taken.

In some embodiments, prior to the restoration of data to the DRAM from nonvolatile memory, the DRAM is scrubbed by a hardware scrubbing element 350, where the hardware element 350 is capable of storing zeros in each bit of the computer DRAM significantly more quickly than this process may be accomplished using software commands. In some embodiments, the result is scrubbed memory 370 that contains zero elements in every bit. In some embodiments, the scrubbed memory 370 then may be utilized to hold the restored data.
system context from nonvolatile memory, where the non-zero pages may be entered
according to a zero filter page, such as zero filter page 250 in Figure 2, thereby
returning the DRAM to a state for transition by the operating system back to an
operational (SO) state.

[0045]  **Figure 4** is an illustration of an embodiment of memory
processing in a fast platform hibernate apparatus, system, or method. In some
embodiments, in FFS DMA processing 400, unfiltered DMA table entries 405 are
processed by a zero page filter 410 to identify zero pages in the table entries. In
some embodiments, the processing results in a table of filtered DMA table entries
415 (such as table entries 270 illustrated in Figure 2) containing non-zero pages and
empty entries.

[0046] In some embodiments, the filtered table entries are compacted,
shown by the process of a table compactor 420, resulting in the compacted DMA
entries 425, which generally are significantly reduced from the original table entries.
In some embodiments, the compacted DMA entries 425 are processed by a
hardware DMA engine 430 for storage in nonvolatile memory, shown as storage in
a solid state drive (SSD) 435.

[0047]  **Figure 5** is an illustration of an embodiment of resumption of a
computing system from a fast platform hibernation process. In some embodiments,
an apparatus, system, or method provides for resumption from fast hibernation,
where the resumption may be described as a two phase process. In some
embodiments, a first phase 510 provides for the preparation of the DRAM for
loading of context data.

[0048] In some embodiments, upon the computing platform receiving a
power on event, the power management logic or BIOS is powered on operates to
initialize the DRAM memory 515. However, upon initialization, the bits of DRAM
memory will generally be random. In order to provide for the loading of context
data, the memory is scrubbed to provide a zero in every bit of the memory. In some
embodiments, the DRAM 515 is scrubbed by a hardware scrubber 520, where the
hardware scrubber can quickly zero out the DRAM to produce scrubbed memory
525, containing zero in data bits of the memory.
In some embodiments, a second phase 560 involves obtaining the context data stored in SSD memory 575, where compacted DMA entries 570 (the non-zero pages) were stored by FFS DMA table processor 565 together with the zero page filter (such as element 250 illustrated in Figure 2) in the fast hibernation process. In some embodiments, the DMA engine 580 operates to obtain the data from the SSD, which further involve obtaining the zero page filter from the SSD. In some embodiments, the DMA engine 580 operates to store the compacted data in the scrubbed DRAM 585 according to the zero page filter, thereby resulting in putting the DRAM in a state for transition to an operational power state by the operating system.

Figure 6 is a flowchart to illustrate an embodiment of a process for fast platform hibernation of a computing system. In this illustration, a computing system may be started 600, thus placing the computing system in an operational (SO) power state. The computer system may continue with various computing system operation 605. Upon detecting an event calling for the computing system to be placed in a first reduced power state referred to here as a sleep state (which may be an S3 state, or other similar state) 610, the operating system take such actions need to enter into the sleep state, including storing current context data 616 to DRAM system memory 660.

In some embodiments, the control is passed to logic such as BIOS, power management logic, or firmware (with the BIOS case illustrated in Figure 6). In some embodiments, the BIOS operates to scan the DRAM system memory for active and inactive pages, and to generate a zero page filter of the non-active pages 620. In some embodiments, the BIOS uses the zero page filter to eliminate inactive DRAM memory pages and thus compress the context data 625.

In some embodiments, the BIOS operates to store the compressed context data and the zero page filter 630 in nonvolatile memory (NVRAM) 670. In some embodiments, the storage in NVRAM may include certain data that may be accessible quickly upon initiation of the computing system, such as, for example, calendar or schedule data. In some embodiments, the BIOS then shuts down power to the DRAM, and proceeds to transition the system to a second reduced power state.
referred to here as a hibernate state (which may be an S4 state, or other similar state) 635, leaving the computing system in the hibernate state 640.

[0053] **Figure 7** is a flowchart to illustrate an embodiment of a process for fast platform resumption of computing system placed in a hibernate state. In this illustration, a computing system may initially be in a second reduced power state (a hibernate state) 700. Upon detection of some wake event 705 (such as a power button signal, a real-time clock alarm, or other wake event), logic such as the power management logic, BIOS, or firmware (wherein the example of BIOS is provided in Figure 7) powers on prior to other elements of the system 710. In some embodiments, the system may optionally provide for access to a data image for a certain application 715 to allow a user with quick access to this data.

[0054] In some embodiments, the BIOS operates to initiate the DRAM (volatile) system memory of the system 720, which generally will result in largely random data contained in the DRAM. In some embodiments, the BIOS operates to scrub or zero out the DRAM memory 760, where the operation utilizes a hardware scrubber that provides for fast zeroing of the memory 725.

[0055] In some embodiments, the BIOS obtains the stored compressed context data and zero page filter 730 from the NVRAM 770. The BIOS then proceeds to the context data from the NVRAM 770 into the DRAM system memory 780 according to the zero page filter 735, thereby recreating the data prior to elimination of zero page data in the hibernation process, and placing the data in a form for a first reduced power state (sleep state).

[0056] In some embodiments, the operating system may then proceed to restore the operational state using the recreated context data that is stored in the DRAM system memory 740, thereby resulting in transitioning the computing system from the sleep state into an operational state 750.

[0057] **Figure 8** illustrates an embodiment of a computing system that provides for fast platform hibernation and resumption. In this illustration, certain standard and well-known components that are not germane to the present description are not shown. Under some embodiments, the computing system 800 comprises an interconnect or crossbar 805 or other communication means for transmission of data. The computing system 800 may include a processing means
such as one or more processors 810 coupled with the interconnect 805 for processing information. The processors 810 may comprise one or more physical processors and one or more logical processors. The interconnect 805 is illustrated as a single interconnect for simplicity, but may represent multiple different interconnects or buses and the component connections to such interconnects may vary. The interconnect 805 shown in Figure 8 is an abstraction that represents any one or more separate physical buses, point-to-point connections, or both connected by appropriate bridges, adapters, or controllers.

[0058] In some embodiments, the computing system 800 includes a hardware memory scrubber 812, which may be utilized for the scrubbing of memory to replace data with zeros after the initiation of such memory in a resumption of operation after the system was placed in a hibernate state. In some embodiments, the computing system 800 further comprises a random access memory (RAM) or other dynamic storage device or element as a main memory 814 for storing information and instructions to be executed by the processors 810. RAM memory includes dynamic random access memory (DRAM), which requires refreshing of memory contents, and static random access memory (SRAM), which does not require refreshing contents, but at increased cost. DRAM memory may include synchronous dynamic random access memory (SDRAM), which includes a clock signal to control signals, and extended data-out dynamic random access memory (EDO DRAM). In some embodiments, the memory scrubber 812 may access the main memory 814, thus allowing the main memory 814 to be scrubbed prior to the transfer of data into the main memory upon the resumption of operations of the computing system 800. In some embodiments, memory of the system may include certain registers or other special purpose memory. The computing system 800 also may comprise a read only memory (ROM) 816 or other static storage device for storing static information and instructions for the processors 810. The computing system 800 may include one or more nonvolatile memory elements 818 for the storage of certain elements. In some embodiments, the nonvolatile memory elements 818 include nonvolatile memory for the storage of context data in a fast hibernation process.
[0059] The computing system 800 may also be coupled via the interconnect 805 to an output display 840. In some embodiments, the display 840 may include a liquid crystal display (LCD) or any other display technology, for displaying information or content to a user. In some environments, the display 840 may include a touch-screen that is also utilized as at least a part of an input device. In some environments, the display 840 may be or may include an audio device, such as a speaker for providing audio information. The computer system 800 may further include one or more input devices 842, such as a keyboard, mouse or other pointing device, a microphone for audio commands, and other input devices.

[0060] One or more transmitters or receivers 845 may also be coupled to the interconnect 805. In some embodiments, the computing system 800 may include one or more ports 850 for the reception or transmission of data. The computing system 800 may further include one or more antennas 855 for the reception of data via radio signals.

[0061] The computing system 800 may also comprise a power device or system 860, which may comprise a power supply, a battery, a solar cell, a fuel cell, or other system or device for providing or generating power. The power provided by the power device or system 860 may be distributed as required to elements of the computing system 800. In some embodiments, the power system 860 may include or may work with a power management logic, where such power management logic may provide for functions or processes connected with a fast platform hibernate and resumption.

[0062] In the description above, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form. There may be intermediate structure between illustrated components. The components described or illustrated herein may have additional inputs or outputs which are not illustrated or described.

[0063] Various embodiments may include various processes. These processes may be performed by hardware components or may be embodied in
computer program or machine-executable instructions, which may be used to cause a
general-purpose or special-purpose processor or logic circuits programmed with
the instructions to perform the processes. Alternatively, the processes may be
performed by a combination of hardware and software.

[0064] Portions of various embodiments may be provided as a computer
program product, which may include a computer-readable medium having stored
thereon computer program instructions, which may be used to program a computer
(or other electronic devices) for execution by one or more processors to perform a
process according to certain embodiments. The computer-readable medium may
include, but is not limited to, floppy diskettes, optical disks, compact disk read-only
memory (CD-ROM), and magneto-optical disks, read-only memory (ROM),
random access memory (RAM), erasable programmable read-only memory
(EPROM), electrically-erasable programmable read-only memory (EEPROM),
magnet or optical cards, flash memory, or other type of computer-readable medium
suitable for storing electronic instructions. Moreover, embodiments may also be
downloaded as a computer program product, wherein the program may be
transferred from a remote computer to a requesting computer.

[0065] Many of the methods are described in their most basic form, but
processes can be added to or deleted from any of the methods and information can
be added or subtracted from any of the described messages without departing from
the basic scope of the present invention. It will be apparent to those skilled in the
art that many further modifications and adaptations can be made. The particular
embodiments are not provided to limit the invention but to illustrate it. The scope of
the embodiments of the present invention is not to be determined by the specific
examples provided above but only by the claims below.

[0066] If it is said that an element "A" is coupled to or with element
"B," element A may be directly coupled to element B or be indirectly coupled
through, for example, element C. When the specification or claims state that a
component, feature, structure, process, or characteristic A "causes" a component,
feature, structure, process, or characteristic B, it means that "A" is at least a partial
cause of "B" but that there may also be at least one other component, feature,
structure, process, or characteristic that assists in causing "B." If the specification
indicates that a component, feature, structure, process, or characteristic "may", "might", or "could" be included, that particular component, feature, structure, process, or characteristic is not required to be included. If the specification or claim refers to "a" or "an" element, this does not mean there is only one of the described elements.

[0067] An embodiment is an implementation or example of the present invention. Reference in the specification to "an embodiment," "one embodiment," "some embodiments," or "other embodiments" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least some embodiments, but not necessarily all embodiments. The various appearances of "an embodiment," "one embodiment," or "some embodiments" are not necessarily all referring to the same embodiments. It should be appreciated that in the foregoing description of exemplary embodiments of the present invention, various features are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure and aiding in the understanding of one or more of the various inventive aspects. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed invention requires more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed embodiment. Thus, the claims are hereby expressly incorporated into this description, with each claim standing on its own as a separate embodiment of this invention.
CLAIMS

What is claimed is:

1. An apparatus comprising:
   a volatile system memory;
   a nonvolatile memory;
   a processor to operate according to an operating system, the processor to
   transition the apparatus to a first reduced power state upon receipt of
   a request, the transition to the first reduced power state including the
   processor to store context data for the apparatus in the volatile
   system memory; and
   logic to transition the apparatus to a second reduced power state, the logic to
   copy the context data from the volatile system memory to the
   nonvolatile memory for the transition to the second reduced power
   state, wherein the copying of the context data includes the logic to:
   scan the volatile system memory to locate non-active memory
   elements in the volatile system memory,
   eliminate the non-active memory elements from the volatile system
   memory to generate compressed context data, and
   store the compressed context data in the nonvolatile memory.

2. The apparatus of claim 1, further comprising a hardware scrubbing element
   to store zeros in data elements of the volatile system memory prior to the
   logic restoring the context data from the nonvolatile memory to the volatile
   system memory.

3. The apparatus of claim 1, wherein the logic is further to restore the context
   data from the nonvolatile memory to the volatile system memory upon a
   wake event for the apparatus.

4. The apparatus of claim 3, wherein the logic is to generate a filter to indicate
   locations of the non-active memory elements.
5. The apparatus of claim 4, wherein the logic to copy the context data further includes the logic to store the filter in the nonvolatile memory.

6. The apparatus of claim 5, wherein to restore the context data includes the logic to obtain the filter from the nonvolatile memory and to store the context data to the volatile system memory according to the filter.

7. The apparatus of claim 3, wherein the logic is to store zeros in data elements of the volatile system memory prior to restoring the context data from the nonvolatile memory to the volatile system memory.

8. The apparatus of claim 1, wherein the first reduced power state is a sleep state and the second reduced power state is a hibernate state, the apparatus to consume less power in the second reduced power state than the first reduced power state.

9. The apparatus of claim 8, wherein the logic to transition to the second reduced power state includes the logic to store data for a certain application in a location in the nonvolatile memory, and wherein the logic is further make the data for the application available prior to restoring the context data from the nonvolatile memory to the volatile system memory.

10. A method comprising:
    transitioning a computing system to a first reduced power state upon receipt of a request, the first reduced power state including the storage of context information for the computing system in a volatile system memory; and
    transitioning the computing system to a second reduced power state, including copying the context data from the volatile system memory to a nonvolatile memory in the transition to the second reduced power state, wherein the copying of the context data includes:
        scanning the volatile system memory to locate non-active memory elements in the volatile system memory,
eliminating the non-active memory elements from the volatile system memory to generate compressed context data, and storing the compressed context data in the nonvolatile memory.

11. The method of claim 10, further comprising:
    detecting a wake event for the computing system; and
    restoring the context data from the nonvolatile memory to the volatile system memory.

12. The method of claim 11, further comprising storing zeros in data elements of the volatile system memory prior to restoring the context data from the nonvolatile memory to the volatile system memory.

13. The method of claim 12, wherein storing the zeros in the data elements of the volatile system memory includes utilizing a hardware scrubbing element to scrub the volatile system memory.

14. The method of claim 10, further comprising generating a filter to indicate locations of the non-active memory elements.

15. The method of claim 14, further comprising storing the filter in the nonvolatile memory.

16. The method of claim 15, wherein restoring the context data from the nonvolatile memory to the volatile system memory includes:
    obtaining the filter from the nonvolatile memory; and
    storing the context data to the volatile system memory according to the filter.

17. The method of claim 10, wherein transitioning to the second reduced power state includes storing data for a certain application in a location in the nonvolatile memory, and further comprising making the data for the application available prior to restoring the context data from the nonvolatile memory to the volatile system memory.
18. A system comprising:
   a DRAM (dynamic random access memory) element;
   a nonvolatile memory element;
   a processor to operate according to an operating system, the processor to
   transition the system to a first reduced power state upon receipt of a
   request, the first reduced power state including the storage of context
   information for the system in the volatile system memory;
   logic to transition the system between the first reduced power state and a
   second reduced power state, the logic to copy the context data from
   the DRAM element to the nonvolatile memory element in a transition
   from the first reduced power state to the second reduced power state
   and to restore the context information from nonvolatile memory
   element to the DRAM element in a transition from the second
   reduced power state to the first reduced power state; and
   a hardware memory scrubber to store zeros in elements of the DRAM
   element prior to the restoration of the context information from
   nonvolatile memory element to the DRAM element.

19. The system of claim 18, wherein the logic to copy of the context data from
    the DRAM element to the nonvolatile memory element includes the logic to:
    scan the volatile system memory to locate non-active memory elements in
    the volatile memory and generate a filter of the non-active memory
    elements;
    eliminate the non-active memory elements from the volatile system memory
    to generate compressed context data; and
    store the compressed context data and the filter in the nonvolatile memory.

20. The system of claim 18, wherein the logic to restore the context data
    includes the logic to obtain the filter from the nonvolatile memory and store
    the context data to the volatile system memory according to the filter.
21. A computer-readable medium having stored thereon data representing sequences of instructions that, when executed by a processor, cause the processor to perform operations comprising:

transitioning a computing system to a first reduced power state upon receipt of a request, the first reduced power state including the storage of context information for the computing system in a volatile system memory;

transitioning the computing system to a second reduced power state, including copying the context data from the volatile system memory to a nonvolatile memory in the transition to the second reduced power state, wherein the copying of the context data includes:

scanning the volatile system memory to locate non-active memory elements in the volatile system memory,

eliminating the non-active memory elements from the volatile system memory to generate compressed context data, and

storing the compressed context data in the nonvolatile memory.

22. The medium of claim 21, further comprising instructions that, when executed by the processor, cause the processor to perform operations comprising:

detecting a wake event for the computing system; and

restoring the context data from the nonvolatile memory to the volatile system memory.

23. The medium of claim 22, further comprising instructions that, when executed by the processor, cause the processor to perform operations comprising:

storing zeros in data elements of the volatile system memory prior to

restoring the context data from the nonvolatile memory to the volatile system memory.
24. The medium of claim 22, wherein storing the zeros in the data elements of the volatile system memory includes directing a hardware scrubbing element to scrub the volatile system memory.

25. The medium of claim 21, further comprising instructions that, when executed by the processor, cause the processor to perform operations comprising:
   generating a filter to indicate locations of the non-active memory elements.

26. The medium of claim 25, further comprising instructions that, when executed by the processor, cause the processor to perform operations comprising:
   storing the filter in the nonvolatile memory.

27. The medium of claim 26, wherein restoring the context data includes:
   obtaining the filter from the nonvolatile memory; and
   storing the context data to the volatile system memory according to the filter.

28. The medium of claim 21, further comprising instructions that, when executed by the processor, cause the processor to perform operations comprising:
   storing data for a certain application in a location in the nonvolatile memory
   as a part of transitioning the computing system to the second reduced power state; and
   making the data for the application available prior to restoring the context data from the nonvolatile memory to the volatile system memory.
Fig. 1
Fig. 2
Fig. 3
Fig. 5
Start 600

Computing system operations 605

Event to place computing system in sleep mode? 610

Yes

OS - Store context data to DRAM 615

BIOS – Scan memory for active and inactive pages and generate zero page filter 620

BIOS – Remove zero pages to compress context data 625

BIOS – Store data and zero page filter in NVRAM 630

BIOS – Shut down DRAM and transition system to hibernation 635

Hibernate state 640

No

DRAM – Context data 660

NVRAM – Non-zero data 670

Fig. 6
Fig. 7

Hibernate state 700

Wake event? 705

No

Power on BIOS 710

Yes

Provide data image of certain application to provide quick access to device 715

BIOS - Initiate DRAM 720

BIOS - Zero out DRAM memory using hardware scrubber 725

BIOS – Obtain compressed context data and zero page filter from NVRAM 730

BIOS – Store context data from NVRAM in DRAM according to the zero page filter 735

OS – Restore operational state using context data stored in DRAM 740

Operational state 750

DRAM – Zero data 760

NVRAM – Stored data 770

DRAM – Restored context data 780
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/06(2006.01)i, G06F 9/22(2006.01)i, G06F 1/32(2006.01)i, G06F 12/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 9/06; G06F 12/02; G06F 1/26; G06F 12/00; G06F 1/00; G06F 013/28; G06F 1/32

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: hibernation, resumption, volatile system memory, transition a computing system;

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 2006-0200691 A1 (TAKASHI YOMO et al.) 07 September 2006&lt;br&gt;See the abstract, claims 1-20 and figures 1-7.</td>
<td>1-28</td>
</tr>
<tr>
<td>A</td>
<td>US 2010-0037076 A1 (REECE DEAN et al.) 11 February 2010&lt;br&gt;See the abstract, claims 1-13 and figures 1-5.</td>
<td>1-28</td>
</tr>
<tr>
<td>A</td>
<td>US 2011-0231595 A1 (WARKAT NIR J. et al.) 22 September 2011&lt;br&gt;See the abstract, claims 1-29 and figures 1-12.</td>
<td>1-28</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search 26 APRIL 2012 (26.04.2012)

Date of mailing of the international search report 04 MAY 2012 (04.05.2012)

Name and mailing address of the ISA/KR
Korean Intellectual Property Office
Government Complex-Daejeon, 189 Cheongsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea
Facsimile No. 82-42-472-7140

Authorized officer
BOK. Jin Yo
Telephone No. 82-42-481-5113

Form PCT/ISA/210 (second sheet) (July 2009)
### INTERNATIONAL SEARCH REPORT

#### Information on patent family members

<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2006-020069 A1</td>
<td>07.09.2006</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2005-0097239 A1</td>
<td>05.05.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2005-0108589 A1</td>
<td>19.05.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 6963934 B2</td>
<td>08.11.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2009-016477 A1</td>
<td>25.06.2009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 7900074 B2</td>
<td>01.03.2011</td>
</tr>
<tr>
<td>US 2011-0231595 A1</td>
<td>22.09.2011</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>