RTmpls copy

BATTERY OPERATED COMPUTER OPERABLE UNDER A PLURALITY OF OPERATING SYSTEMS

DESCRIPTION

This invention relates to the field of data processing, and, more particularly to method and apparatus for initialization of a power management system in a battery operable computer that can operate under a plurality of operating systems.

RELATED APPLICATION

Attention is directed to our copending European patent application filed concurrently herewith and claiming priority from Application serial number 647,118, filed in the US on 25 January 1991.

The above related application is directed to a battery operated computer having a sophisticated power /management system. Such computer has two processors, a main processor and a power management processor (PMP). The main processor operates under the control of an operating system and provides overall control of the computer for executing application programs. The main processor also assists in power management by executing power management interrupt handlers and controlling a power control register to turn power on and off to various devices. The power management processor monitors ambient temperature and humidity, and battery conditions, and generates interrupts as a result of predetermined changes. Such processor also controls charging of the battery. Logic means are responsive to predetermined conditions to also generate interrupts.

An interrupt controller is connected to the main processor and receives the power management interrupts (PMIs). When such a computer is turned on or is rebooted, it is necessary to initialize the system and enable the PMIs. This enablement links the appropriate interrupt handlers to

THIS SHEET HAS BEEN REPLACED

the PMIs. In a prior art battery operated computer, specifically the IBM model PC-Convertible computer, the initialization including PMI enablement was done during a power on self test (POST). This method can not be used with OS/2 because a protect mode interrupt handler is not loaded until after the OS/2 kernel loads. The prior art technique would allow power management interrupts to occur during the OS/2 load process and since no handler is available at that time, such interrupt would cause the processor to become unstable and crash.

According to the present invention there is provided in a battery operated computer having a main processor, a main memory for storing application and operating system programs for execution by said processor, and a memory device for storing a basic input output system and power management interrupt handlers, the improvement comprising: initialization means operated in response to turning on said computer and rebooting said computer after it has been turned on, to sequentially load into said main memory a preselected one of a real mode operating system or a protected mode operating system, and to thereafter selectively utilize said basic input output system to enable power management interrupts in accordance with which one of said operating systems is loaded into said main memory.

Further according to the present invention there is provided a method of initializing a battery operated computer having a main processor, a main memory for storing application and operating system programs for execution by said processor, and a memory device for storing a basic input output system and power management interrupt handlers, said method comprising the steps of: executing a power on self test program; upon completion of said test loading into said main memory a preselected one of a real mode operating system or a protected mode operating system; and thereafter selectively utilizing said basic input output system to enable power management interrupts accordance with which one of said operating systems is loaded into said main memory.

In the drawings:-

This Sheet Has Been Replaced



Fig. 1 is a block diagram of a battery operated computer embodying the invention; and

Fig. 2 is a flow chart illustrating operation of the invention.

Referring now to the drawings, Fig. 1 generally shows a computer 10 the details of which are more fully disclosed in the above identified application to which reference may be had for further details not disclosed herein. Computer 10 comprises a main processor 12 connected to a system bus 14. A main memory 16 is connected to bus 14 for storing application operating system and application programs for execution by processor 12. A read only memory (ROM) 18 is also connected to bus 14 and comprises or stores a POST program 20 and basic I/O operating system (BIOS) programs 22. Such BIOS programs include a power management (PM) BIOS comprising various interrupt handlers for handling the various PMIs.

Computer 10 is selectively operable with power from a battery 11 or from an ac power source (not shown). Battery 11 is connected to a power distribution system (PDS) 24 which operates under the control of settings of a power control register (PCR) 26. PCR 26 is connected to bus 14 and is set or controlled by processor 12 executing PM BIOS.

A power management processor (PMP) 28 is connected by a bus 30 to bus 14 and together with processor execute and perform the power management functions for operating computer 10. PMP 28 is also connected to a keyboard 32 and in addition to performing the PM functions PMP 28 manages the keyboard.

A plurality of sensors 34 are connected to PMP 28 to monitor or measure environmental and battery conditions including ambient temperature and humidity, and battery voltage, current, and temperature. In response to such measurements, PMP 28 determines the energy level of battery 11 and generates a low battery signal on line 36 when the battery becomes discharged and needs replacement or recharging. PMP 28 also generates a

PM software interrupt (PMSI) signal on line 34 when there is a watchdog timeout, extreme temperature, or low battery condition. The PMSI signal is fed into a multiplexer or MUX 40. Various other conditions produce signals that are analyzed by logic 38 to produce suspend and resume (S/R) interrupts on lines 44, the signals being a LID POSITION signal indicating when the computer lid has been opened or closed, a NOT DOCKED signal indicating the computer is not connected to a docking station, a TOD ALARM signal generated at a preset time of day, and a MODEM RING signal generated when a modem rings for transmitting information to computer 10.

MUX 40 has two additional input lines 46 and 48. Line 46 receives a signal SYSTEM VOLTAGE when the system voltage is at or above a predetermined level (5 volts). It is used when a discharged battery is replaced by a charged battery and the system voltage becomes normal upon connection to the charged battery. It is also used when the computer is in a suspend mode and is then attached to an ac power source. Line 48 transmits an A/M signal when the position of a switch 50 is changed. Switch 50 has two settings or positions, an "automatic" and a "manual" which allow the user to select a preset speed at which to run processor 12 or to allow the computer to automatically select the processor running speed.

Thus, MUX 40 receives a plurality of interrupts or requests therefor. MUX 40 polls the input lines on a round robin basis and when it notes an active signal on one of the lines, it outputs onto lines 52 a power management interrupt (PMI) along with a three-bit code identifying the specific interrupt. Lines 52 are connected to interrupt controller 54. The interrupt request from MUX 40 is transmitted on a specific one of plural interrupt request lines into controller 54. In response to receiving the interrupt from MUX 40, controller 54 transmits an interrupt request to processor 12 on one of lines 56. When processor 12 is able to handle the interrupt, it acknowledges such request and then controller 54 transmits to the processor a vector identifying the specific PM interrupt handler for processing such interrupt. Processor

12 thereupon accesses the desired handler in BIOS 22 and executes it accordingly. Obviously, the above process presupposes that PM interrupts have been enabled.

Processor 12 receives two types of interrupts, maskable and non-maskable. A PM interrupt is a maskable interrupt. When the computer is initially turned on, or when it is subsequently rebooted, the computer is initialized and one step of such initialization involves enabling PM interrupts. Thus, with reference to Fig. 2, when the computer is powered on or rebooted in step 62, POST routine 20 is first executed. In accordance with the above mentioned prior art, the PC Convertible prior art POST routine also performed step 63 to thereupon enable PM interrupts during POST. However, such step precluded being able to load and execute OS/2. Therefore, step 63 is eliminated from the POST routine 20 of the invention. Continuing with the operation of the invention, upon completion of POST 20, an operating system is loaded in accordance with the system setup. Thus, assuming DOS is the desired operating system, step 64 loads it into main memory for execution by processor 12. After DOS is loaded, a DOS prompt appears on the display screen (not shown) and the computer in step 66 waits for a key stroke from keyboard 32. When a key stroke occurs, step 82 processes the stroke and upon completion branches to step 66 to set up a loop for processing subsequent keystrokes. Had the PM interrupts been enabled by step 63 in accordance with the prior art, such processing would have occurred immediately following step 66. However, in accordance with the invention, a plurality of other steps occur between step 66 and step 82.

Step 66 is processed by immediately making a BIOS call in step 68 to issue a "wait for key" call via interrupt 16 and setting processor register AH to 0. PM BIOS 22 is then entered at step 70 by setting AH=41 and issuing an int=15 call. Step 70 then routes or directs that step 72 be the next step. Step 72 decides if this is the first time this string of code or routine is being executed since turning the power on or rebooting the system. If so, step 74 enables power management interrupt in controller 54 by setting an input gate (not shown) connected to line

52. If step 72 results in a negative determination or upon completion of step 74, step 76 then stops processor clocks by shutting off power to the processor until an interrupt occurs. Then step 78 exits PM BIOS 22 and returns to step 68. If the interrupt was caused by a keystroke, step 68 branches along line 80 to step 82 whereupon the keystroke can be processed. If the interrupt was not caused by a keystroke, step 68 enters step 70 to repeat the process until a keystroke occurs.

If the computer is setup to run under OS/2, then step 84 loads the appropriate portions of OS/2 into main memory. Step 84 then loads the PM driver and enables PM interrupts in a manner the same as that described for step 74. Step 88 then loads any other drivers and the operating system kernel is started in step 90. Step 92 is an idle task which is entered from the kernel. Step 92 looks to see if the processor is busy with other tasks. If the processor is not busy, step 92 requests by step 96 that the processor clocks be stopped as with step 76. When that is done, step 94 returns to step 92. When an interrupt subsequently occurs, the processor is started and the interrupt handled.

Thereafter, when computer 10 is operating under either DOS or OS/2, when a PM interrupt occurs at 98, PM BIOS 22 is entered at step 100 to monitor the hardware activity. Step 102 decides if the hardware has been idle (by looking at waiting interrupts in controller 54,for a period of n seconds where n is a value selected by the user or given a system default value. If the hardware has not been idle, step 106 returns to the application. If the hardware has been idle for the time period, step 104 starts a suspend interrupt handler to switch computer 10 into the suspend mode.

It should be apparent to those skilled in the art that many changes can be made in the details and arrangements of steps and parts without departing from the scope of the invention as defined in the appended claims.

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

CLAIMS

1. In a battery operated computer having a main processor, a main memory for storing application and operating system programs for execution by said processor, and a memory device for storing a basic input output system and power management interrupt handlers, the improvement comprising:

initialization means operated in response to turning on said computer and rebooting said computer after it has been turned on, to sequentially load into said main memory a preselected one of a real mode operating system or a protected mode operating system, and to thereafter selectively utilize said basic input output system to enable power management interrupts in accordance with which one of said operating systems is loaded into said main memory.

2. A computer in accordance with claim 1, wherein:

said real mode operating system is a disk operating system.

3. A computer in accordance with claim 1 wherein:

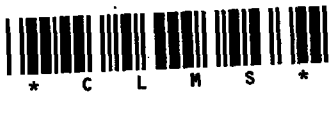
said protected mode operating system is a multitasking operating system.

4. A computer in accordance with claim 3 wherein:

said initialization means loads a power management driver to enable said power management interrupts.

5. A computer in accordance with claim 2 wherein:

said system includes an interrupt controller having selectively enabled interrupt input lines;



This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

and said initialization means enables one of said interrupt input lines for power management interrupts.

6. A computer in accordance with claim 5 wherein:

said initialization means sets up a loop to analyze plural keystrokes, and said power management interrupts are enabled on a first pass through said loop.

7. A computer in accordance with claim 6 wherein said initialization means is operable to load a kernel of said protect mode operating system into main memory, and set up a loop of waiting for said processor to become idle;

said processor being shut off when it becomes idle.

8. A method of initializing a battery operated computer having a main processor, a main memory for storing application and operating system programs for execution by said processor, and a memory device for storing a basic input output system and power management interrupt handlers, said method comprising the steps of:

executing a power on self test program;

upon completion of said test loading into said main memory a preselected one of a real mode operating system or a protected mode operating system;

and thereafter selectively utilizing said basic input output system to enable power management interrupts in accordance with which one of said operating systems is loaded into said main memory.



* C L M S *

THIS SHEET HAS BEEN REPLACED

9. The method in accordance with claim 8, comprising:

storing in said computer, before said test is executed, a plurality of power management interrupt handlers;

said processor being operable upon completion of said enabling power management interrupts, to thereafter execute one of said power management interrupt handlers in response to a power management interrupt.

10. A method in accordance with claim 9 wherein said power management interrupts are enabled by enabling a line into an interrupt controller.

11. A method in accordance with claim 9 wherein said processor is shut off after said power management interrupts are enabled, to conserve power.

This Sheet Has Been Replaced

