

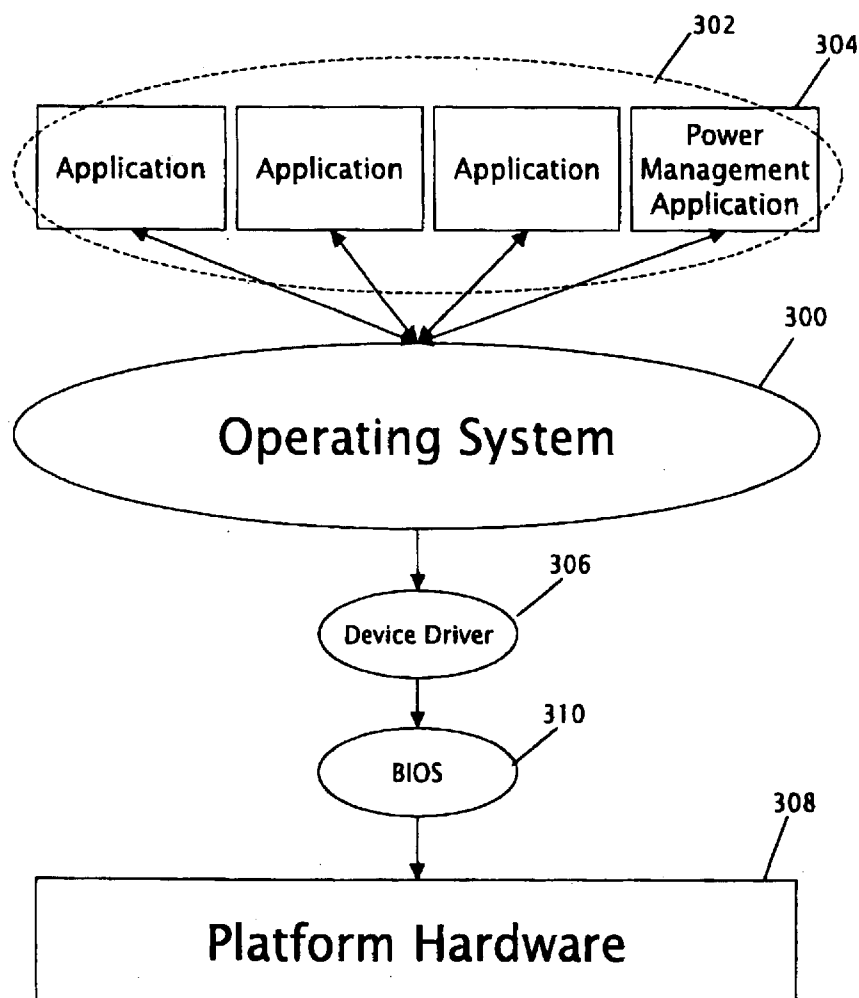


US 20050108587A1

(19) **United States**(12) **Patent Application Publication** (10) **Pub. No.: US 2005/0108587 A1**
(43) **Pub. Date: May 19, 2005**(54) **DEMAND-BASED METHOD AND SYSTEM
OF CPU POWER MANAGEMENT****Publication Classification**(76) Inventors: **Barnes Cooper**, Beaverton, OR (US);
Jay Arjangrad, Portland, OR (US)(51) **Int. Cl.⁷** **G06F 1/32**
(52) **U.S. Cl.** **713/320**(57) **ABSTRACT**

Correspondence Address:
BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030 (US)

A demand-based method and system of central processing unit power management. The utilization of a central processing unit (CPU) during a sampling time interval is determined by measuring a time quantum within the sampling time interval during which a central processing unit clock signal is active within a processor core of the CPU. The total number of cycles of the central processing unit clock signal that are applied to the processor core and the period of the central processing unit clock signal are used to determine the time quantum. The utilization may then be expressed in terms of a ratio of the time quantum to the total time interval and used to select a processor performance mode. The CPU is then operated in the selected processor performance mode.

(21) Appl. No.: **11/006,872**(22) Filed: **Dec. 7, 2004****Related U.S. Application Data**(63) Continuation of application No. 09/751,759, filed on
Dec. 30, 2000, now Pat. No. 6,829,713.

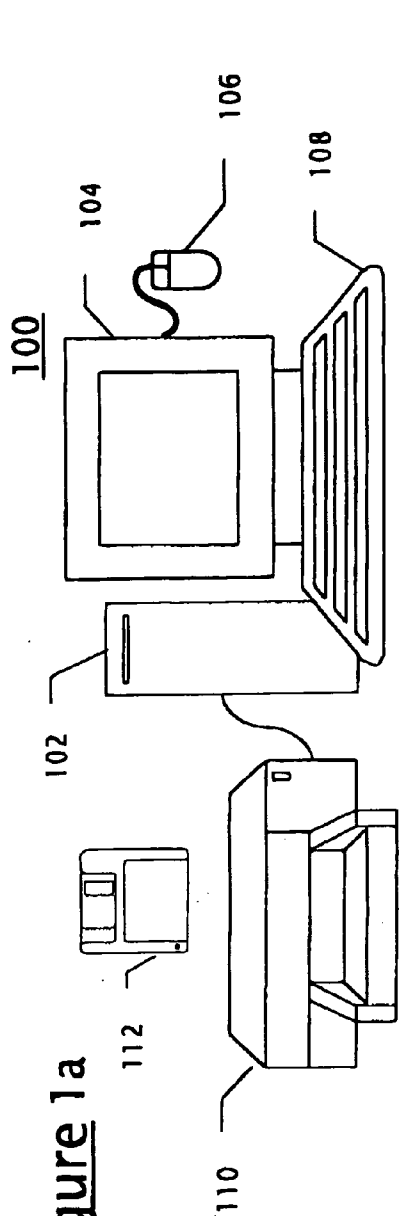


Figure 1a

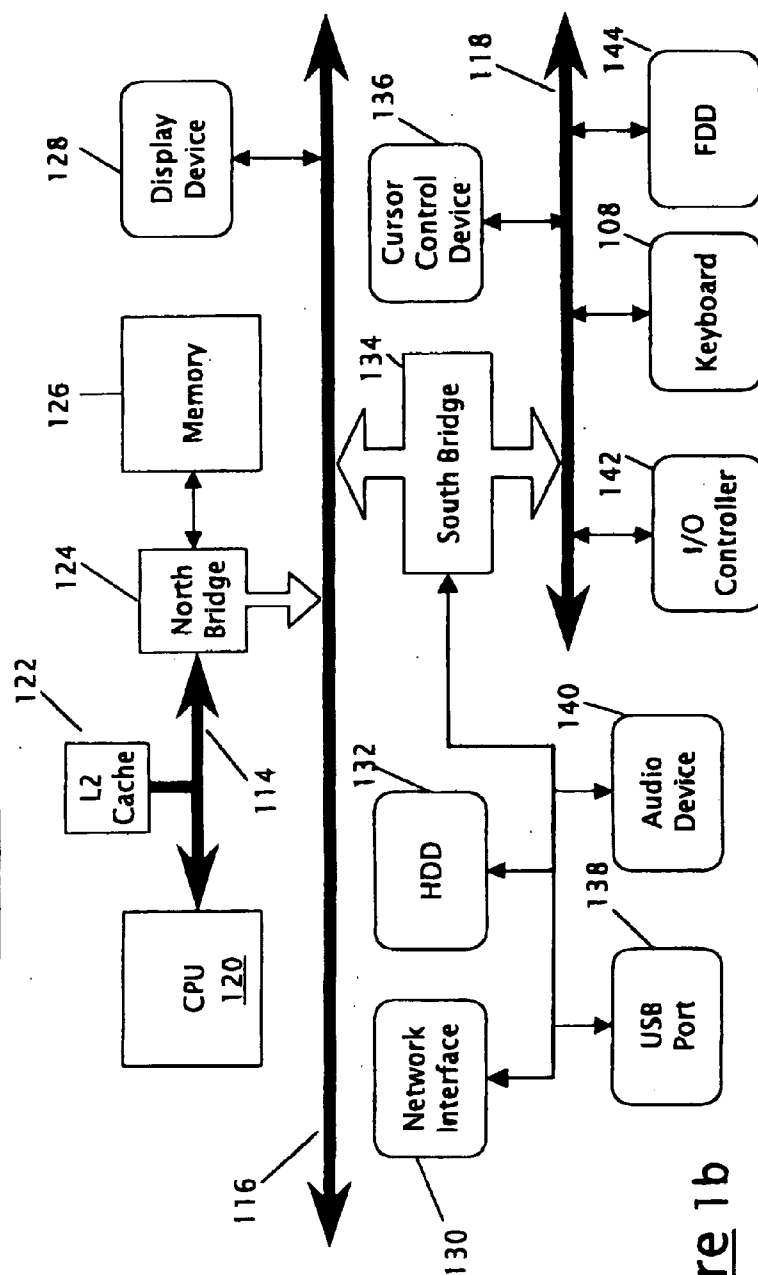
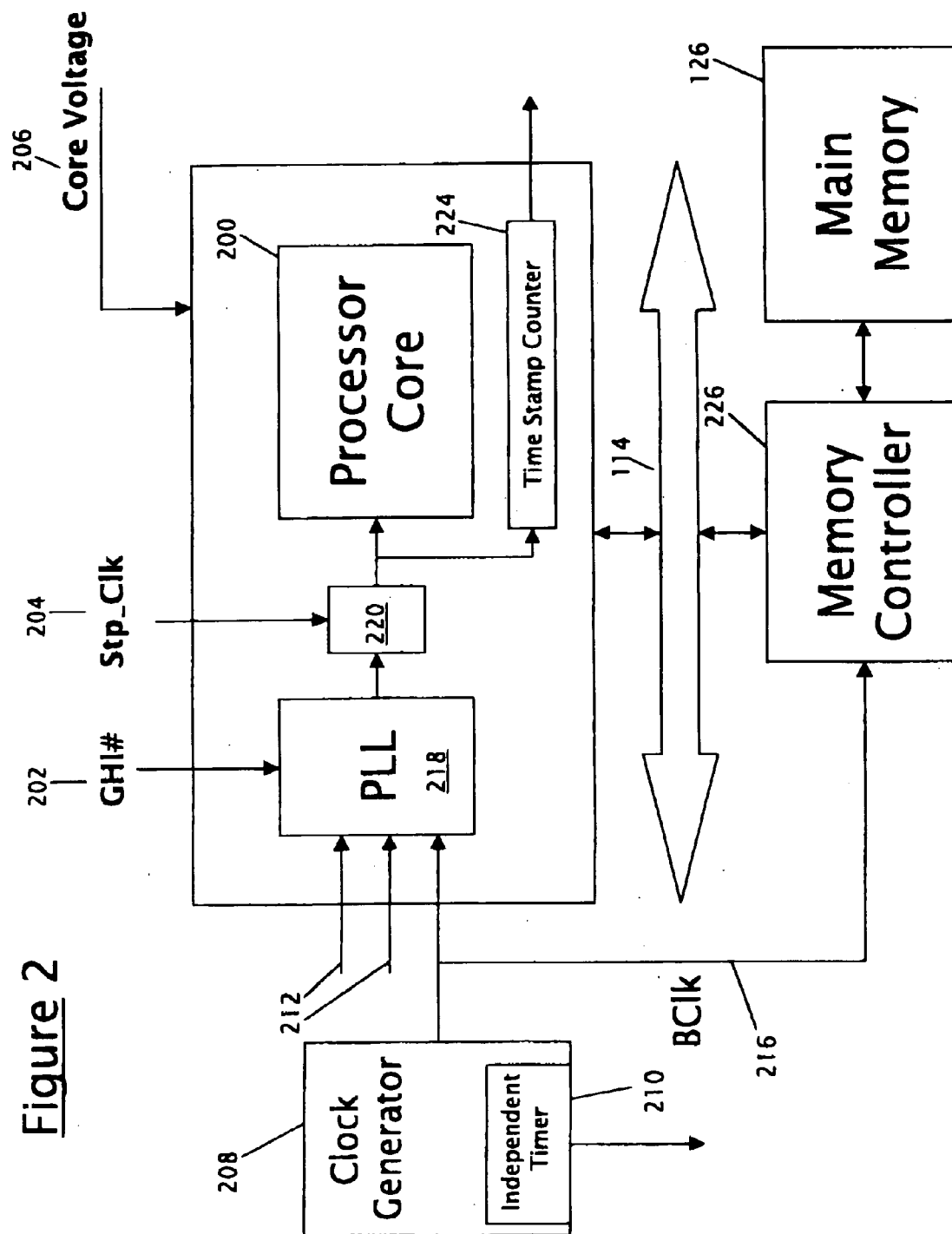


Figure 1b

Figure 2



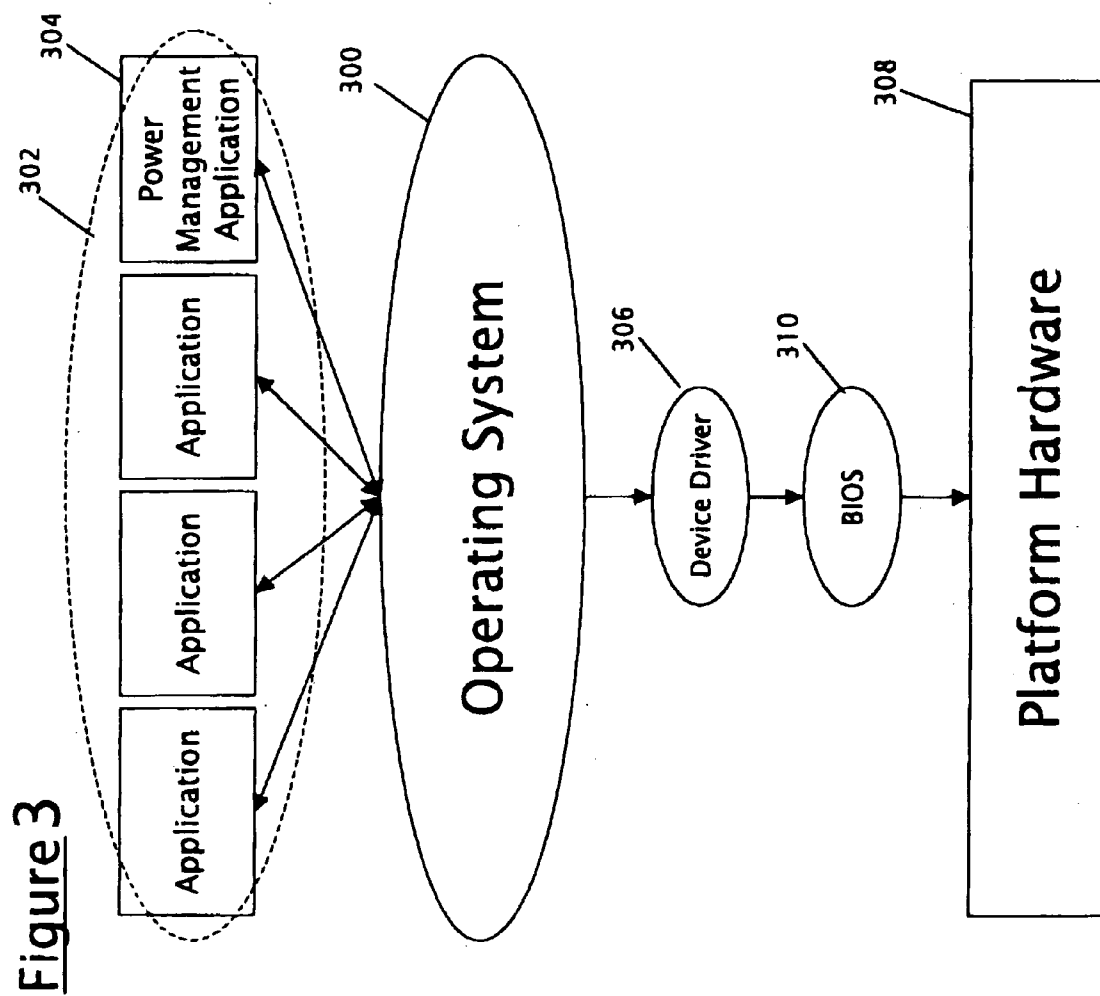


Figure 4

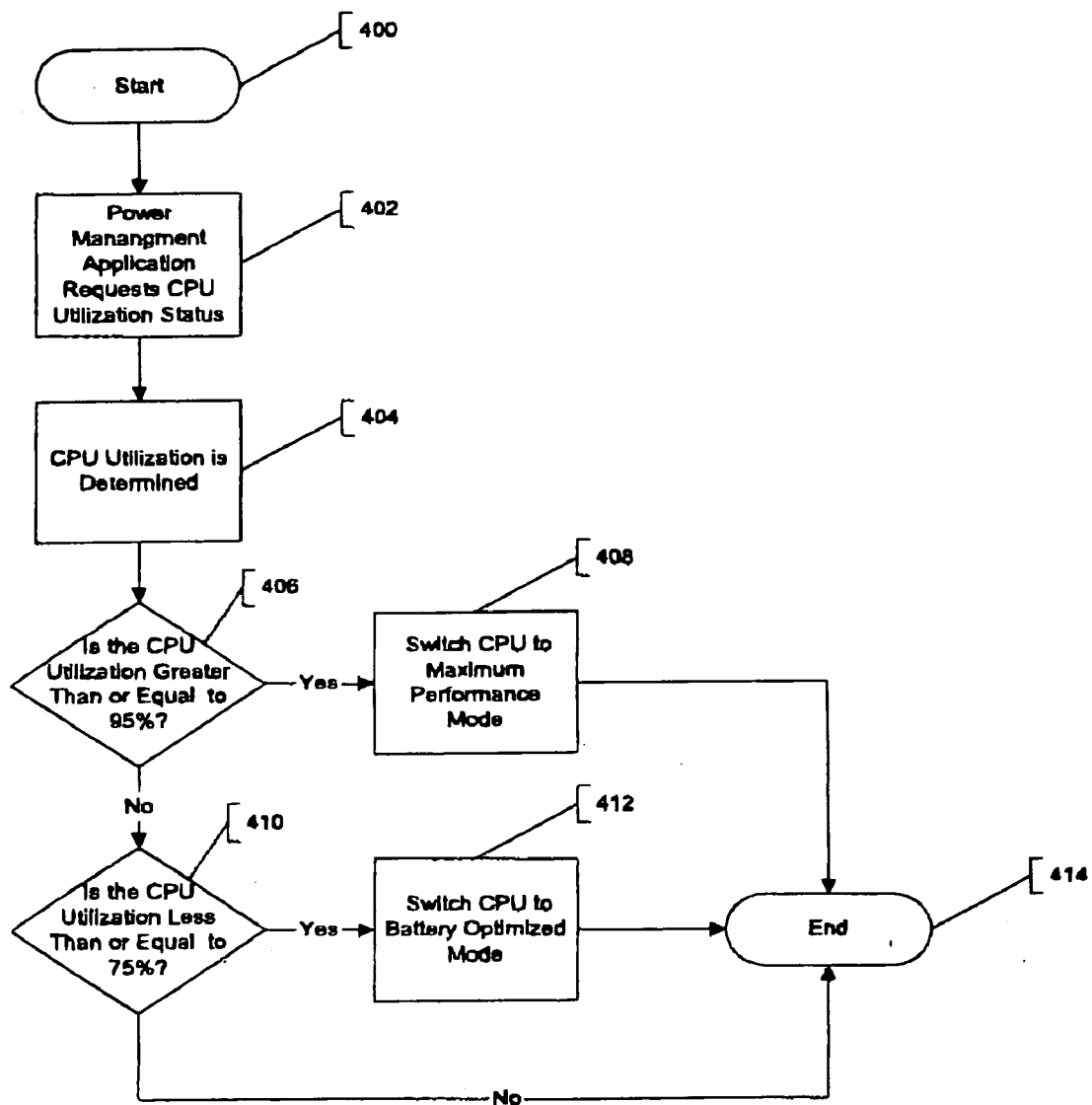


Figure 5

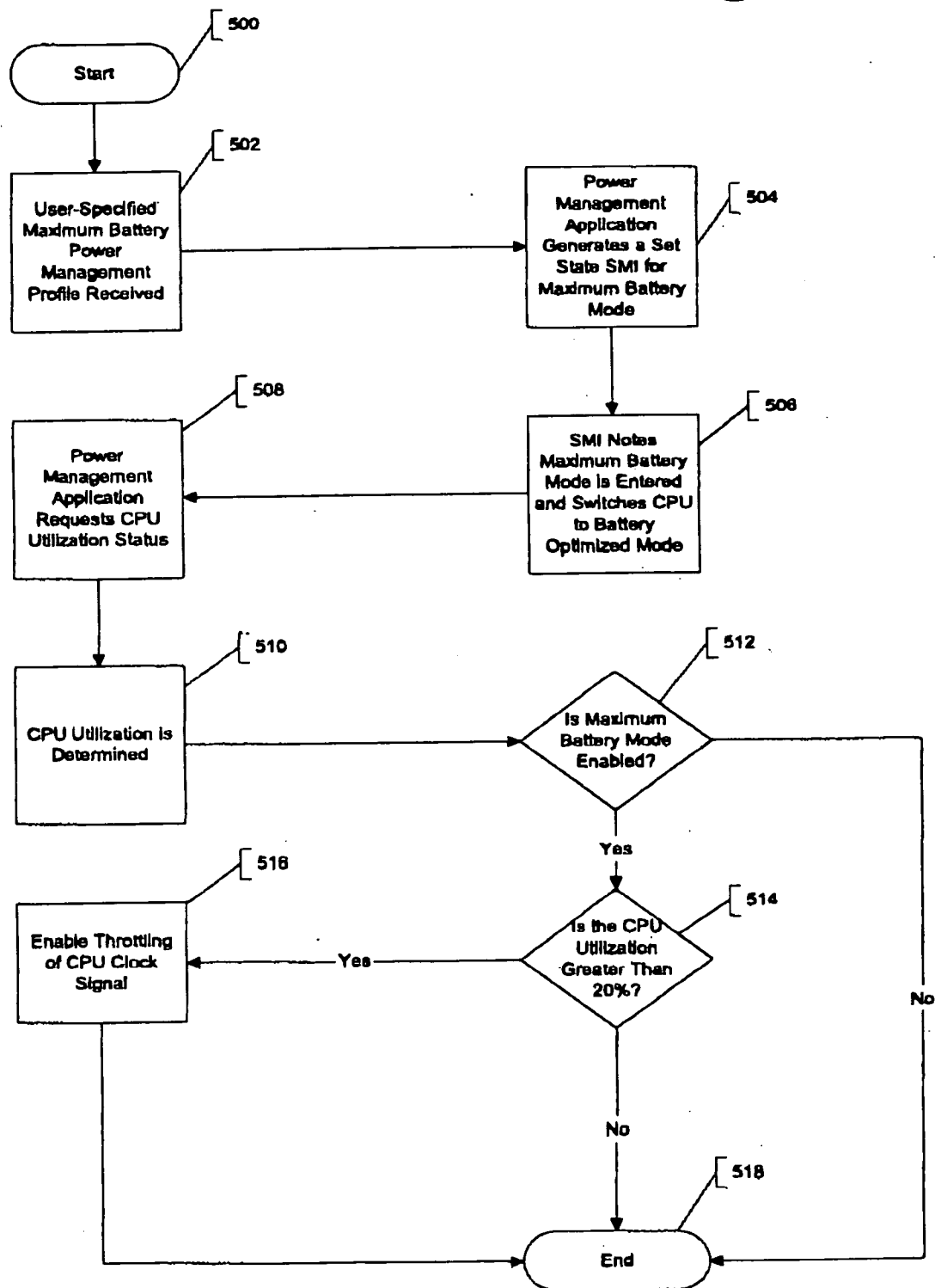


Figure 6

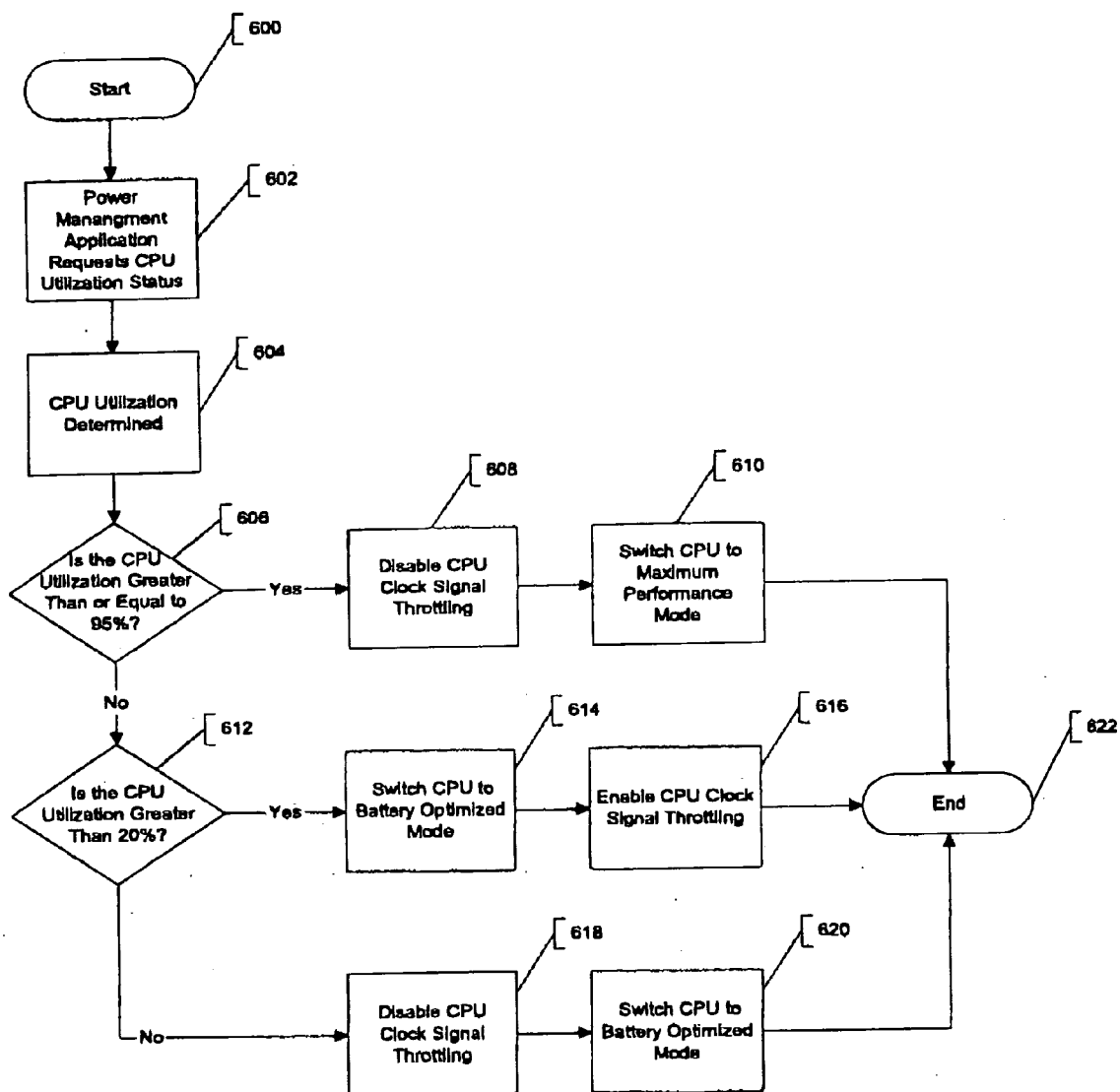
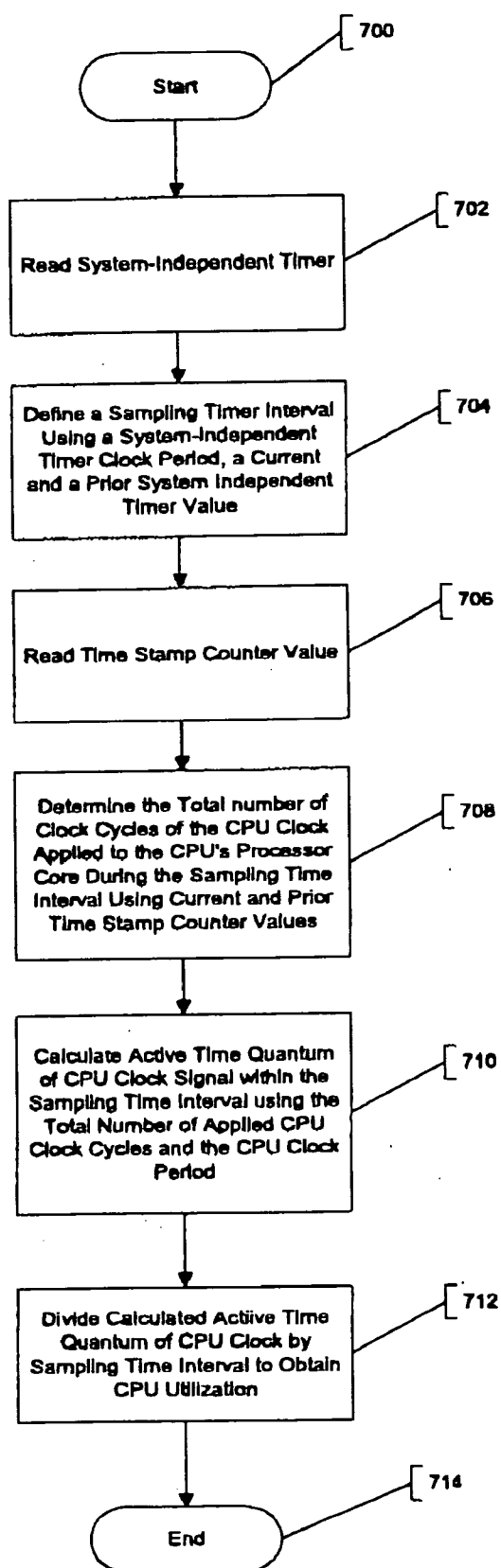


Figure 7



DEMAND-BASED METHOD AND SYSTEM OF CPU POWER MANAGEMENT

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. patent application Ser. No. 09/751,759 entitled "Demand-Based Method and System of CPU Power Management," assigned to the assignee of the present invention and filed Dec. 30, 2000.

FIELD OF THE INVENTION

[0002] The field of the invention relates generally to central processing units (CPUs). More particularly the field invention relates to CPU power management. Still more particularly, the field of the invention relates to a demand-based method and system of CPU power management.

BACKGROUND OF THE INVENTION

[0003] As battery-dependent portable computing devices (notebook computers, personal digital assistants, etc.) have become more prevalent, the conservation of battery power or "power management" has become more and more important. In many power management systems, some or all system components may be deactivated or "powered down" to conserve power. This method however, requires that the devices powered down be inactive or unused for a sufficiently long period of time to justify the latency associated with their re-activation. Therefore, a number of methods have been implemented to decrease device power consumption within the active or "powered on" state. Since, the power dissipated by a device is dependent both on its applied voltage and on the frequency with which device transitions or "switching" occurs, conventional power management techniques typically focus on one or both of these factors.

[0004] Modern power management systems implement a variety of voltage and frequency reduction or "scaling" techniques. Although substantial power savings can be realized by reducing a device's voltage, special hardware is often required to correctly operate such devices using low and variable voltages. Such voltage reduction techniques also currently limit the maximum frequency at which a device may be operated. Similar power savings may be realized by scaling a device's operating frequency or "clock". In conventional power management systems, a device's operating frequency may be altered in a variety of ways. In one approach, the applied clock signal is periodically stopped and restarted such that the average or effective operating frequency is lowered (throttling). In another approach, a lower frequency clock signal, generated independently or derived from an existing clock, is applied to a device. Although these approaches may be used alone or in combination to reduce a device's or system's power consumption, this frequency scaling technique reduces the operating frequency of the device, and consequently the number of operations or tasks it can perform.

[0005] In the past, several approaches have been taken to control the activation of the above-described power management techniques such as the user selection of a pre-defined power mode, the occurrence of environmental events such as the application or removal of an A/C (alternating current) power source, or the detection of a system or

device temperature. More recently, power management systems have looked to device utilization or "idleness" to trigger the application or removal of such techniques in an effort to conserve power in a more user-transparent manner. When a utilization-based power managed device is idle for a pre-determined period of time, power reduction techniques such as voltage and frequency scaling are applied to decrease the amount of power consumed. The greatest difficulty traditionally associated with such demand-based systems has been in determining a device's current utilization, particularly for processing devices such as the central processing unit (CPU) of a data processing system.

[0006] In a conventional operating system (OS), CPU utilization is determined by accumulating CPU idle time across a sampling interval to determine the percentage of time the processor is inactive. To accomplish this, a list of tasks or threads is maintained by the OS which are ready-to-run, i.e., not waiting for some event to resume execution. When this ready-to-run list is empty, no tasks are being executed and the processor is idle. Accordingly, a CPU-independent timer is read and the processor is placed in a low power state. When a new task is added to the ready-to-run list, the processor is placed in an active state and the timer is read again. The difference between the first and second timer reads (multiplied by the timer's period) then represents the CPU's idle time. The accumulation of this time across a sampling interval is then used to determine the CPU utilization (what percentage of the CPU's time is spent idle). Unfortunately, neither this measure of CPU utilization nor the state of the ready-to-run task list is available outside of the OS through a supported application programming interface (API). Consequently, this OS-generated CPU utilization metric cannot be utilized in a "demand" or utilization-based power management system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which

[0008] FIG. 1a illustrates a conventional data processing system useable with the present invention;

[0009] FIG. 1b illustrates a prior art architecture of the data processing system depicted in FIG. 1a;

[0010] FIG. 2 illustrates a portion of the architecture depicted in FIG. 1b in greater detail;

[0011] FIG. 3 illustrates an architectural system diagram depicting the operation of a data processing system according to the present invention;

[0012] FIG. 4 illustrates a high-level logic flowchart of a first embodiment of the method of the present invention;

[0013] FIG. 5 illustrates a high-level logic flowchart of a second embodiment of the method of the present invention;

[0014] FIG. 6 illustrates a high-level logic flowchart of a third embodiment of the method of the present invention;

[0015] FIG. 7 illustrates a high-level logic flowchart of a method of determining the utilization of a central processing unit according to one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0016] A demand-based method and system of CPU power management is disclosed. In the following detailed descrip-

tion, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one of ordinary skill in the art that these specific details need not be used to practice the present invention. In other circumstances, well-known structures, materials, circuits, processes and interfaces have not been shown or described in detail in order not to unnecessarily obscure the present invention.

[0017] Referring now to FIG. 1a, a conventional data processing system 100 useable with the present invention is illustrated. Data processing or computer system 100 is comprised of a system unit 102, output devices such as display 104 and printer 110, and input devices such as keyboard 108, and mouse 106. Data processing system 100 receives data for processing by the manipulation of input devices 108 and 106 or directly from fixed or removable media storage devices such as disk 112 and network connection interfaces (not shown). Data processing system 100 then processes data and presents resulting output data via output devices such as display 104, printer 110, fixed or removable media storage devices like disk 112 or network connection interfaces.

[0018] Referring now to FIG. 1b, there is depicted a high-level block diagram of the components of a data processing system 100 such as that illustrated by FIG. 1a. In a conventional computer system, system unit 102 includes a processing device such as central processing unit (CPU) 120 connected to a level two (L2) cache 122 over a processor system bus (PSB) 114. Processor system bus 114 is in turn coupled to an expansion bus such as local bus 116 and a memory 126 via a north bridge circuit 124. Local bus 116 may include a peripheral component interconnect (PCI), Video Electronics Standards Association (VESA) bus or the like, tightly coupled to the processor 120 and the processor system bus 114 to permit high-speed access to select devices such as display device 128.

[0019] Memory 126 may include read-only (ROM) and/or random access (RAM) memory devices such as a synchronous dynamic random access memory (SDRAM) module capable of storing data as well as instructions to be executed by CPU 120. Access to data and instructions stored within memory 126 is provided via a memory controller (not shown) within north bridge circuit 124. L2 cache 122 is similarly used, typically in a hierarchical manner, to store data and instructions for direct access by CPU 120. Display device 128 may include a cathode ray tube (CRT) display such as display 104, liquid crystal display (LCD), or a similar device for displaying various kinds of data to a computer user. For example, image, graphical, or textual information may be presented to the user on display device 128. System unit 102 of data processing system 100 also features an expansion or "compatibility" bus 118 such as the Industry Standard Architecture (ISA) bus, and a south bridge circuit 134 coupling it to local bus 116 to facilitate the attachment of other, relatively slower devices to the system 100. South bridge circuit 134 includes a universal serial bus (USB) port 138 as well as other direct connections for devices such as a network interface card 130, a data storage device, such as a magnetic hard disk drive 132, and an audio device 140 such as a speaker or sound card.

[0020] Other devices not directly coupled to south bridge 134 may be connected to the system 100 via the expansion

bus 118 as illustrated. A floppy disk drive (FDD) 144 providing additional data storage capacity on removable media storage devices such as disk 112, and input devices such as a keyboard 108 and a cursor control device 136 are each coupled to expansion bus 118 in this manner to communicate data, instructions, and/or command selections to central processing unit 120. Cursor control device 136 may comprise a conventional mouse such as mouse 106 of FIG. 1a, a trackball, or any other device capable of conveying desired cursor manipulation. Similarly, expansion bus 118 includes an input/output (I/O) controller having standard serial and parallel port functionality for connecting other I/O devices such as printer 110 to the system.

[0021] The system of the present invention includes software, information processing hardware, and various processing steps, which will be described below. The features and process steps of the present invention may be embodied in machine or computer executable instructions embodied within media such as disk 112. The instructions can be used to cause a general purpose or special purpose processor such as CPU 120, which is programmed with the instructions to perform the described methods of the present invention. Alternatively, the features or steps of the present invention may be performed by specific hardware components that contain hard-wired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

[0022] Referring now to FIG. 2, a portion of the architecture depicted in FIG. 1b is illustrated in greater detail. Processor 120 is shown in communication with memory 126 over the processor system bus 114 utilizing a memory controller 226 of north bridge circuit 124. A common system clock, (BCLK) 216 is generated by a clock generator 208 and applied to a clock control phase lock loop (PLL) 218 of CPU 120 and to memory controller 226. A core voltage 206 is similarly applied to CPU 120 in the illustrated embodiment, providing necessary operating power. While the BCLK signal 216 is applied, accesses to and from memory 126 occur at its frequency of approximately 100 megahertz (MHz). The central processing unit 120 however, is capable of performing tasks at much greater speeds than this and accordingly, a bus ratio or multiplier 212 is selected using a clock control signal, GHI#202 and a higher frequency central processing unit clock signal is generated utilizing PLL 218. So for example, if the system or front side clock 216 has a frequency of 100 MHz, and a ratio 212 of 5 to 1 is selected using the GHI# signal 202, then the generated CPU clock will have a frequency of approximately 500 MHz. Alternatively, a higher multiplier or ratio 212 of say 7 to 1 could be selected, yielding a CPU clock frequency of approximately 700 MHz.

[0023] The generated central processing unit clock signal is then applied to clock throttling logic 220 before being passed to processor core 200. Throttling is a technique by which the CPU clock is deasserted or "gated off" from the processor core to prevent functional units within the core from operating. Throttling logic 220 therefore acts as a switch, actuated by a stop clock (Stp_Clk) control signal 204, between the PLL 218 and the processor core 200. A time stamp counter 224 is also included within the CPU 120 and incremented for each cycle (sometimes called ticks or pulses) of the CPU clock which is "gated through" or applied to the processor core as shown. Because time stamp

counter **224** tracks the number of clock ticks or cycles applied to the functional units of the processor core **200** such as instruction decoders, floating point and integer execution units, etc. it provides an extremely accurate representation of the actual work performed by CPU **120**. One additional chipset architecture component illustrated in **FIG. 2** is independent timer **210**. System independent timer **210** runs independently of CPU **120** and its associated system clock **216**, unaffected by Stp_Clk signal **204** throttling or BClk signal **216** frequency modifications. Using the number of ticks of independent timer **210** elapsed between reads and its fixed frequency, an accurate measure of the passage of time may be obtained. In one embodiment, a Windows™ high performance counter, exported via the Win32 Application Programming Interface (API) as the QueryPerformanceCounter() function can be used as independent timer **210**. In an alternative, Advanced Configuration and Power Interface (ACPI) compliant embodiment, a power management timer may be utilized. Although in the illustrated embodiment independent timer **210** is depicted as being integrated with clock generator circuit **208**, in alternative embodiments the timer **210** may be generated in a separate device or integrated circuit.

[0024] Referring now to **FIG. 3**, an architectural system diagram depicting the operation of a data processing system according to the present invention is illustrated. In the illustrated embodiment, a plurality of application programs **302** such as power management application **304** interact with various platform hardware devices **308** including a CPU **120** via an operating system **300** such as the Windows™ operating system from Microsoft Corporation, one or more device drivers **306**, and basic input/output system (BIOS) code **310**. The illustrated system is interrupt-driven both with respect to the multitasking of the various applications **302** and communication between applications **302** and platform hardware **308**.

[0025] Accordingly, in one embodiment of the present invention, an application **302** request for a hardware resource from within platform hardware **308** can cause an interrupt, such as a System Control Interrupt (SCI) or a System Management Interrupt (SMI) to be generated and an interrupt handler routine to be responsively executed. Interaction between operating system **300** and platform hardware **308** is then facilitated by a device driver **306** and BIOS **310**. In the illustrated embodiment, BIOS **310** contains information such as physical device addresses of the various devices **308** attached to the data processing system **100** and is useful with respect to the actual transmission of data. By contrast, device driver **306** is typically specific to a particular hardware device and is usually concerned with the translation of data between various device formats.

[0026] Referring now to **FIG. 4**, a high-level logic flowchart of a first embodiment of the method of the present invention is illustrated. In **FIG. 4** there is depicted a technique by which a demand-based transition between two processor performance states is executed. At block **400**, the illustrated process is begun and thereafter a CPU utilization status request is received from a power management application (block **402**). The described utilization request may be periodic or may occur in response to relevant power management events such as thermal or processor workload events, the connection of an alternating current power supply or the like. Once the CPU utilization has been estab-

lished (block **404**), a determination is then made whether the calculated utilization exceeds a utilization threshold (block **406**). In the illustrated embodiment, a relatively high utilization threshold of 95% is selected to identify the execution of demand-intensive applications such as DVD movie players, personal computer games, and performance benchmark tests. It should be readily appreciated however that the various utilization thresholds described herein have been selected for illustrative purposes only and that a wide range of threshold values could be substituted therefore without departing from the spirit and scope of the present invention. If the utilization threshold is exceeded, the CPU is transitioned to a maximum performance processor performance mode (block **408**) and operated at a higher performance level to ensure that the execution performance of such demand-intensive application programs is not degraded.

[0027] If the utilization of the CPU is not above or equal to the 95% utilization threshold, it is then determined whether the CPU's utilization falls at or below a second utilization threshold of, in the illustrated embodiment, 75% (block **410**). The processor performance level may then be matched to its current utilization level by switching the CPU to a battery optimized processor performance mode (block **412**) to conserve power when the utilization level falls below this figure and a decrease in performance will be less noticeable to the end user. Otherwise the process is terminated (block **414**) with the processor performance mode of the central processing unit remaining unchanged. Power may be conserved and the maximum performance mode distinguished from the battery optimized mode by the frequency at which the processor is operated. While numerous other power and performance management techniques are known and within the scope of the present invention, in one embodiment utilization of the maximum performance processor performance mode entails the operation of the central processing unit at an operating frequency of 600 MHz while the battery optimized mode entails the application of a 500 MHz central processing unit clock signal. Following any transition to either maximum performance or battery optimized mode, the process is terminated (block **414**). In an alternative embodiment, factors other than an instantaneous CPU utilization and a utilization threshold may be used to select an appropriate processor performance mode such as the duration of time that the examined CPU remains at a particular utilization level or within a particular range of utilization levels.

[0028] Referring now to **FIG. 5**, a high-level logic flowchart of a second embodiment of the method of the present invention is illustrated. After the process is begun (block **500**) a user-specified power management profile is received (block **502**) in which power conservation and system performance are prioritized generally or a specific, preferred processor performance mode may be designated. In the illustrated embodiment, a maximum battery or ultra battery optimized profile is received conveying that power conservation is to be favored over execution speed. Then an executing power management software or firmware application generates a system management interrupt (SMI) (block **504**) in response to the receipt of the user power management profile which in turn transitions the CPU to battery optimized mode if necessary from whatever prior state the processor was operating in. Subsequently, the power management application issues a request for the current CPU utilization status (block **508**) which is deter-

mined either by the generated SMI or directly by the power management application itself (block 510) by a method which will be described in greater detail with reference to FIG. 7 herein. In alternative embodiment, the described system management interrupt is used only to transition the system from one performance or power mode to another with both CPU utilization detection and other related tasks being performed directly by the power management application.

[0029] The user-specified power management profile is then checked to ensure that maximum battery mode is still currently enabled (block 512). If so, the resolved CPU utilization is examined to determine whether it exceeds a utilization threshold of 20% (block 514) in this embodiment. If not, the process is terminated (block 518). If the current utilization of the CPU exceeds the tuneable threshold, the CPU is transitioned from battery optimized mode to a virtual maximum battery performance mode by engaging throttling of the central processing unit clock signal at a particular frequency (block 516). Otherwise, the process ends (block 518) and the battery optimized performance mode is utilized until another transition-precipitating event occurs. Using the illustrated process allows small, bursty tasks or code segments which can be completed within the sampling time interval of the CPU utilization determination to be executed at the full, battery optimized performance level without enabling CPU clock signal throttling. Such tasks can be completed faster at this non-throttled rate, allowing the system to transition after their completion to an even lower power state than can be achieved with clock throttling, conserving more power overall.

[0030] Referring now to FIG. 6, a high-level logic flowchart of a third embodiment of the method of the present invention is illustrated. The beginning of the process is depicted at block 600 and thereafter a CPU utilization status request is received from a power management application (block 602). Once the CPU utilization has been established (block 604), a determination is made whether the calculated utilization exceeds a utilization threshold (block 606). In the illustrated embodiment, a relatively high utilization threshold of 95% is selected for this first utilization threshold as illustrated. If the utilization threshold is exceeded, any previously applied CPU clock signal throttling is disabled (block 608) and the CPU is transitioned to a maximum performance processor performance mode (block 610) and operated at a higher performance level to ensure that the execution performance of demand-intensive application programs is not degraded.

[0031] If the utilization of the CPU is not above or equal to the 95% utilization threshold, it is then determined whether the CPU's utilization falls at or below a second utilization threshold of, in the illustrated embodiment, 20% (block 612). If the current CPU utilization level is not greater than the 20% utilization threshold, the CPU is operated in battery optimized mode (block 620) and clock throttling is disabled (block 618) such that power saving states such as the C2 and C3 states defined by the well known Advanced Configuration and Power Interface Specification, Revision 2.0, Jul. 27, 2000 (ACPI) can be entered more quickly following completion of the CPU workload. Lastly, for CPU utilizations falling in between the two utilization thresholds, the CPU is transitioned to and operated in maximum battery mode by entering battery opti-

mized mode (block 614) and enabling clock throttling for the applied CPU clock (block 616). Consequently, the performance of CPU workloads having a consistent, intermediate demand intensity is reduced and the completion time is extended in order to reduce the total amount of power consumed. Following any transition to (or retention of) any of the above-described power management performance modes (maximum performance, battery optimized mode, and maximum battery) the process is terminated (block 622).

[0032] Referring now to FIG. 7, a high-level logic flowchart of a method of determining the utilization of a central processing unit according to one embodiment of the present invention is illustrated. FIG. 7 depicts a technique by which a the utilization of a CPU may be determined independently of a data processing system's operating system. In one embodiment, this method is utilized to determine CPU utilization within the various method embodiments of the present invention such as at blocks 404, 510, and 604 of FIGS. 4, 5, and 6, respectively. The process illustrated by FIG. 7 begins at block 700. Thereafter, a system-independent timer such as an ACPI chipset-compliant power management timer or Windows™ performance counter is read. (block 702). Next, a sampling time interval is defined using the independent timer's clock period, as well as currently and previously read system-independent timer values (block 704). A value is then read from a time stamp counter (block 706) which is incremented for each cycle or "clock" of a CPU clock signal which is applied to the processor core 200 of central processing unit 120. Using a previously read time stamp counter value and the currently read value, the total number of CPU clock signal ticks or cycles applied to the CPU's processor core 200 during the sampling time interval may be obtained (block 708). Thereafter, the total amount or "quantum" of time within the sampling time interval during which the CPU clock signal was active within the CPU's processor core 200 can be derived using the accumulated number of CPU clock cycles and the CPU clock signal's period (block 710). CPU utilization may then be expressed as a ratio of this active CPU clock signal time to the sampling time interval (block 712). Thereafter, the process is terminated (block 714).

[0033] Although the present invention is described herein with reference to a specific preferred embodiment, many modifications and variations therein will readily occur to those with ordinary skill in the art. Accordingly, all such variations and modifications are included within the intended scope of the present invention as defined by the following claims.

What is claimed is:

1. A method comprising:

determining a utilization of a central processing unit;

placing said central processing unit in a first processor performance mode in response to a determination that said utilization is greater than a first utilization threshold;

placing said central processing unit in a second processor performance mode in response to a determination that said utilization is less than or equal to said first utilization threshold, wherein said central processing unit

operates at a higher frequency in said first processor performance mode than in said second processor performance mode.

2. The method as set forth in claim 1, further comprising:

placing said central processing unit in a third processor performance mode in response to a determination that said utilization is greater than or equal to a second utilization threshold, wherein said second utilization threshold is greater than said first utilization threshold and said central processing unit operates at a higher frequency in said third processor performance mode than in said second processor performance mode.

3. The method as set forth in claim 1, wherein placing said central processing unit in a first processor performance mode comprises:

receiving a user-specified power management profile; and

placing said central processing unit in said first processor performance mode in response to receiving said user-specified power management profile.

4. The method as set forth in claim 1, wherein placing said central processing unit in a first processor performance mode in response to a determination that said utilization is greater than a first utilization threshold comprises placing said central processing unit in a first processor performance mode in response to a determination that said utilization is greater than 20%.

5. The method as set forth in claim 4, wherein placing said central processing unit in a second processor performance mode in response to a determination that said utilization is less than or equal to said first utilization threshold comprises placing said central processing unit in a second processor performance mode in response to a determination that said utilization is less than or equal to 20%.

6. The method as set forth in claim 2, wherein placing said central processing unit in a third processor performance mode in response to a determination that said utilization is greater than or equal to a second utilization threshold comprises placing said central processing unit in a third processor performance mode in response to a determination that said utilization is greater than or equal to 20%.

7. A data processing system-readable medium having a plurality of instructions executable by a data processing system embodied therein, wherein said instructions when executed cause said data processing system to perform a method comprising:

determining a utilization of a central processing unit;

placing said central processing unit in a first processor performance mode in response to a determination that said utilization is greater than a first utilization threshold;

placing said central processing unit in a second processor performance mode in response to a determination that said utilization is less than or equal to said first utilization threshold, wherein said central processing unit operates at a higher frequency in said first processor performance mode than in said second processor performance mode.

8. The data processing system-readable medium as set forth in claim 7, wherein said method further comprises:

placing said central processing unit in a third processor performance mode in response to a determination that

said utilization is greater than or equal to a second utilization threshold, wherein said second utilization threshold is greater than said first utilization threshold and said central processing unit operates at a higher frequency in said third processor performance mode than in said second processor performance mode.

9. The data processing system-readable medium as set forth in claim 7, wherein placing said central processing unit in a first processor performance mode comprises:

receiving a user-specified power management profile; and

placing said central processing unit in said first processor performance mode in response to receiving said user-specified power management profile.

10. The data processing system-readable medium as set forth in claim 7, wherein placing said central processing unit in a first processor performance mode in response to a determination that said utilization is greater than a first utilization threshold comprises placing said central processing unit in a first processor performance mode in response to a determination that said utilization is greater than 20%.

11. The data processing system-readable medium as set forth in claim 10, wherein placing said central processing unit in a second processor performance mode in response to a determination that said utilization is less than or equal to said first utilization threshold comprises placing said central processing unit in a second processor performance mode in response to a determination that said utilization is less than or equal to 20%.

12. The data processing system-readable medium as set forth in claim 8, wherein placing said central processing unit in a third processor performance mode in response to a determination that said utilization is greater than or equal to a second utilization threshold comprises placing said central processing unit in a third processor performance mode in response to a determination that said utilization is greater than or equal to 95%.

13. A plurality of instructions executable by a data processing system and embodied within a data processing system-readable medium, comprising:

a first code segment to determine a utilization of a central processing unit;

a second code segment to place said central processing unit in a first processor performance mode in response to a determination that said utilization is greater than a first utilization threshold;

a third code segment to place said central processing unit in a second processor performance mode in response to a determination that said utilization is less than or equal to said first utilization threshold, wherein said central processing unit operates at a higher frequency in said first processor performance mode than in said second processor performance mode.

14. The plurality of instructions as set forth in claim 13, wherein said second code segment to place said central processing unit in a first processor performance mode comprises:

a fourth code segment to receive a user-specified power management profile; and

a fifth code segment to place said central processing unit in said first processor performance mode in response to receiving said user-specified power management profile.

15. The plurality of instructions as set forth in claim 13, further comprising:

a sixth code segment to place said central processing unit in a third processor performance mode in response to a determination that said utilization is greater than or equal to a second utilization threshold, wherein said second utilization threshold is greater than said first utilization threshold and said central processing unit operates at a higher frequency in said third processor performance mode than in said second processor performance mode.

16. The plurality of instructions as set forth in claim 13, wherein said second code segment to place said central processing unit in a first processor performance mode in response to a determination that said utilization is greater than a first utilization threshold comprises a code segment to place said central processing unit in said first processor performance mode in response to a determination that said utilization is greater than 20%.

17. The plurality of instructions as set forth in claim 16, wherein said third code segment to place said central processing unit in a second processor performance mode in response to a determination that said utilization is less than or equal to said first utilization threshold comprises a code segment to place said central processing unit in a second processor performance mode in response to a determination that said utilization is less than or equal to 20%.

18. The plurality of instructions as set forth in claim 15, wherein said sixth code segment to place said central processing unit in a third processor performance mode in response to a determination that said utilization is greater than or equal to a second utilization threshold comprises a code segment to place said central processing unit in said third processor performance mode in response to a determination that said utilization is greater than or equal to 20%.

* * * * *