



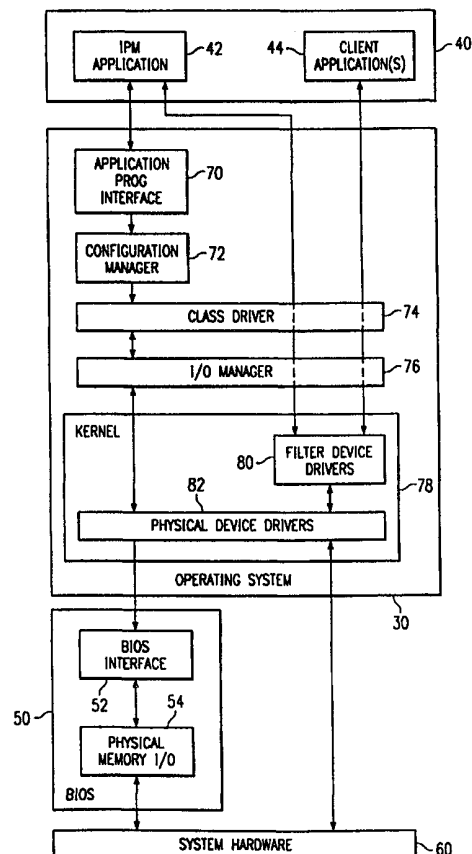
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 1/32	A1	(11) International Publication Number: WO 00/26756 (43) International Publication Date: 11 May 2000 (11.05.00)
(21) International Application Number: PCT/US99/26187 (22) International Filing Date: 4 November 1999 (04.11.99) (30) Priority Data: 09/185,674 4 November 1998 (04.11.98) US (71) Applicant: PHOENIX TECHNOLOGIES LTD. [US/US]; 411 E. Plumeria Drive, San Jose, CA 95134 (US). (72) Inventor: BARBEE, Ronald; 5 Dolphine Drive, Bayou Vista, TX 77563 (US). (74) Agent: PERKINS, Jefferson; Foley & Lardner, 330 North Wabash Avenue, Suite 3300, Chicago, IL 60611 (US).		(81) Designated States: CN, JP, KR, SG, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: METHOD AND APPARATUS FOR PROVIDING INTELLIGENT POWER MANAGEMENT

(57) Abstract

The present invention is an apparatus and method for providing power management apparatus for a circuit in a processor-based system. The apparatus comprises a memory to store instruction sequences by which the processor-based system is processed and a processor coupled to the memory. The stored instruction sequences cause the processor to: (a) determine a system access time of the circuit; (b) determine if the system access time is less than a first predetermined value, if so, increasing an accessibility period of the circuit, during which the circuit is active. Various embodiments are described.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

METHOD AND APPARATUS FOR PROVIDING INTELLIGENT POWER MANAGEMENT

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates in general to processor-based systems, and more particularly to an apparatus and method for providing intelligent power management in processor-based systems.

2. Description of the Related Art

Power management is implemented in processor-based systems to conserve power or to reduce the power consumption of the system. Power management is typically implemented by powering down one or more circuits in the system upon detection of a period of non-use or inactivity. The period of non-use or inactivity, typically termed the "time-out" period, is generally fixed. The user typically has to enter a setup mode for a particular application and select or enter a time-out value for the application. Selection of the time-out value is also based on the user's perception of system performance versus battery life expectancy, and thus may not be optimized.

Accordingly, there is a need in the technology for an apparatus and method for providing power management for a number of applications in a processor-based system, which facilitates conservation of power in the system, while optimizing system performance.

BRIEF SUMMARY OF THE INVENTION

The present invention is an apparatus and method for providing power management apparatus for a circuit in a processor-based system. The apparatus comprises a memory to store instruction sequences by which the processor-based system is processed and a processor coupled to the memory. The stored instruction sequences cause the processor to: (a) determine a system access time of the circuit; (b) determine if the system access time is less than a first predetermined value, if so, increasing an accessibility period of the circuit, during which the circuit is active. Various embodiments are described.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates one embodiment of a power state profile of a device.

Figure 2 is a system block diagram of an exemplary processor system in which the apparatus and method of the present invention is used.

5 Figure 3 is an overall functional block diagram illustrating the architecture of an operating system which utilizes the apparatus and method of the present invention.

Figure 4 is a top-level flow chart of one embodiment of a general process that implements the Intelligent Power Management (IPM) technique of the present invention.

Figure 5 is a flow chart that illustrates one embodiment of the IPM process of the present invention.

Figure 6 is a flow chart that illustrates a second embodiment of the IPM process of the present invention.

15 Figure 7 is a flow chart that illustrates a third embodiment of the IPM process of the present invention.

Figure 8 is a flow chart that illustrates a fourth embodiment of the IPM process of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED INVENTION

Principles of Operation

20 The present invention is an apparatus and method for altering the time period prior to turning off a device based on system-level demands for that device. Figure 1 illustrates one embodiment of a power state profile of a device. As shown, the device transitions from an in-use period, to an inactive period, an off period and then back to an in-use period.

The following reference designations are used in Figure 1:

T represents a period of time beginning from the time a device is turned off until it is again accessed;

30 T[to] represents an inactivity time-out period prior to turning off the device; this value changes as a result of the intelligent power management scheme of the present invention;

$T[\text{delta}]$ represents the total time during which a device is turned off, and occurs after the device inactivity period and prior to the time when the device is accessed by the system;

$T[\text{delta_inc}]$ represents a stored time interval corresponding to each power-managed device, that is used to determine if the device's time-out period $T[\text{to}]$ should be increased. In one embodiment, $T[\text{delta_inc}]$ is approximately equal to $1/3\{T[\text{delta}]\}$;

$T[\text{delta_dec}]$ represents a stored time interval corresponding to each power-managed device, that is used to determine if the device's time-out period $T[\text{to}]$ should be decreased. In one embodiment, $T[\text{delta_dec}]$ is approximately equal to $T[\text{delta}] + 1/3\{T[\text{delta}]\}$.

Other reference designations that are used in the following sections include the following:

$T[\text{ito}]$ represents an initial inactivity time-out value;

DIR represents a flag that indicates the current mode of the "time-out" change;

INDEX represents an integer that is used to control the amount of increase or decrease to the time-out value $T[\text{to}]$.

In general, the intelligent power management technique of the present invention determines if the system accesses of a power-managed device are occurring too quickly, too slowly, or if the system accesses are just about right relative to a predetermined time-out inactivity value. The time-out inactivity value corresponds to a predetermined period of time during which a power-managed device is inactive. Upon reaching the end of the time-out inactivity period, the power-managed device is turned off.

If system accesses are occurring faster than a first predetermined time-out value, the device time-out inactivity period $T[\text{to}]$ is increased. If system accesses are occurring slower than a second predetermined time-out value, $T[\text{to}]$ is decreased. If the accesses are occurring within a range that is acceptable relative to the time-out inactivity value, i.e., $T[\text{to}]$ occurs between the first and second predetermined value, no changes are implemented. In one embodiment, $T[\text{to}]$ may

be increased or decreased at a predetermined rate to accelerate or decelerate the increase or decrease of $T[to]$ by a predetermined number.

In addition, to prevent $T[to]$ from forever increasing without a limit, which would undermine the responsiveness of the system, a boundary condition is set up. A maximum limit $T[to_max]$, where $T[to_max] = K * T[to_max]$, is imposed to ensure that $T[to]$ would not be subject to increasing without limitation. In one embodiment, $K = 16$. In alternate embodiments, K may be any user selected number.

Implementation

The present embodiment is described in reference to a processor system 10. Figure 2 illustrates an exemplary processor system 10 which implements the processes of the present invention. The processor system 10 comprises a CPU 12 and a memory module 14. The memory module 14 includes random access memory (RAM) 14a and read-only memory (ROM) 14b. In one embodiment, the memory module 14 also includes a main memory or a dynamic random access memory (DRAM). The CPU 12 and memory module 14 are coupled to a system bus 16. The processor system 10 may also include various I/O and peripheral modules (MISC I/O #1, #2, ... #N) which are coupled along an I/O bus 20 that is in turn coupled to the system bus 16 via a bus bridge 22. Examples of the peripheral modules include a console, a printer and a mouse.

The present invention is also described with reference to an operating system installed on the processing system 10. Figure 3 is an overall functional block diagram illustrating the architecture of a processing system 10 that utilizes the apparatus and method of the present invention. The processing system 10 comprises an operating system 30 which supports user applications 40, Basic Input/Output System ("BIOS") 50 and system hardware 60. The user applications 40 include an intelligent power management (IPM) application 42 and client application(s) 44. The BIOS 50 is a collection of drivers, or software interfaces for hardware devices such as the console (keyboard and display), a generic printer, the auxiliary device (serial port), the computer's clock and the boot disk device. The BIOS 50 is typically embedded in programmable, read only memory (PROM). An

operating system typically makes no use of the BIOS 50 after the operating system has been booted and is running. The kernel level drivers in the operating system 30 may interface directly with the system hardware 60.

The operating system 30 includes an Application Program Interface (API) 70 which interfaces with the IPM Application 42, and a Configuration Manager 74 which allocates resources installed on the operating system 30 based on requests from the IPM Application 42. The operating system 30 further includes a class driver 74 which interfaces with the user applications 40 (either directly, as in the case of the client applications 44, or via the API 70 and Configuration Manager 72), and an I/O Manager 76. The I/O Manager 76 converts I/O requests (made via the class driver 74) into properly sequenced calls to various driver routines located in the kernel 78. In particular, when the I/O Manager 42 receives an I/O request, it uses the function codes of the request to call one of several dispatch routines in a driver located in the kernel 78. The kernel 78 provides hardware-independent functions, called system functions, that are accessed by means of a software interrupt. The functions provided by the kernel 78 include file and directory management, memory management, character device input/output and time and date support, among others. In one embodiment, the operating system is the Windows NT operating system. In alternate embodiments, the operating system 30 includes the Solaris or the AIX operating systems or other operating systems based on demand-paged virtual memory subsystems.

The present invention provides the IPM Application 42 which monitors the I/O activities of I/O devices such as MISC I/O #1, ..., MISC I/O #N (Figure 2) through the use of one or more Filter Device drivers 80, located within the kernel 74. Based on information provided by the filter device drivers 80, the IPM Application 42 identifies the most suitable power state for power-managed devices installed on the system 10 and/or for the system 10 using the knowledge-based technique of the present invention. The power state of the system hardware 60 can then be efficiently controlled and managed. In one embodiment, the IPM Application 42 controls the power state of system hardware through the use of the Configuration Manager 72, which provides the necessary pointers to the physical

device drivers 82 located within the kernel 78. The physical device drivers 82 can either directly control system hardware 60 or interface with the system hardware 60 through the BIOS interface 52 and Physical Memory and I/O circuit 54 of BIOS 50.

5 In one preferred embodiment, the IPM Application 42 and the Filter Device Drivers 78 comprise source code written in the C language. It is understood that other assembly languages may be utilized in implementing the functions of the IPM Application 42 and Filter Device drivers 78. The BIOS data and addresses are typically located in physical memory 50 (typically in RAM 14a; see Figure 1) and
10 are accessed by the physical device drivers 82.

Figure 4 is a top-level flow chart of one embodiment of a general process that implements the IPM technique of the present invention. In one embodiment, after initialization of the IPM environment, the system process is configured to run at a predetermined interval, e.g., every 10 seconds. This interval was established as
15 a trade-off between the granularity of monitoring and controlling power-managed devices and the minimization of the IPM process's consumption of power. The predetermined interval may be increased or decreased depending on system requirements and user preference.

The system process 100 of the present invention which implements the IPM
20 process (generally represented by process blocks 106-108) proceeds from a start state to process block 102, where the IPM environment is initialized. In particular, various circuits such as timers or variables required for the use of the system process 100 are initialized. The process 100 then advances to process block 104, where I/O accesses from the Filter Device Drivers 80 (see Figure 3) are read. The
25 I/O accesses are then applied by the process 100 to identify and control the power states of one or more power-managed device(s), as shown in process block 106. The process 100 subsequently outputs one or more power state requests for controlling the power-managed device(s) and/or the system 10, as shown in process block 108. In one embodiment, such requests are serviced by the
30 Configuration Manager 72 (Figure 3).

The process 100 then proceeds to decision block 110, where it determines if the system is still active. If not, the process 100 is terminated (process block 112). Otherwise, the IPM process 100 is suspended for a predetermined period of time. In one embodiment, the IPM process 100 is suspended for 10 seconds. In particular, a timer is set to a predetermined period of time. The process 100 determines if the predetermined period has lapsed, as shown in decision block 118. If not, the process 100 continues to monitor the timer event. Otherwise, it proceeds to process block 120, where the operating system places the IPM process into the run state and returns to process block 104.

Figure 5 is a flow chart that illustrates one embodiment of the IPM process of the present invention. In one embodiment, the IPM process of Figure 5 may be implemented for device management of I/O devices such as hard disk drives, CD ROMs, and Modems. The IPM process 200 begins from a start state and proceeds to process block 202, where it determines if T, the system access time of a managed device, is less than a first predetermined value, T(Inc). If so, the process 200 proceeds to decision block 204, where it determines if the flag, DIR FLAG, which indicates the current mode of the time-out inactivity change, has been set to indicate that T[to] should be increased. If not, the process proceeds to process block 206, where the DIR FLAG is set to INC, indicating that T[to], the time-out inactivity period for the managed device, should be increased. In addition, INDEX, the rate of change of T[to], is set to 1. The process 200 then terminates.

If, at process block 204, the process 200 determines that the flag, DIR FLAG, which indicates the current mode of the time-out inactivity change, has been set to indicate that T[to] should be increased, the process proceeds to decision block 208, where it determines if the rate of increasing T[to] is a maximum value. If not, the process 200 returns to the system process 100 flow. Otherwise, the process proceeds to process block 210, where it increases INDEX, the rate of changing T[to] by setting $INDEX = A * INDEX$, where A is a predetermined number. In one embodiment, $A = 2$. The process then proceeds to process block 212, where the time-out inactivity period T[to] is increased by setting $INDEX = INDEX * T[ito]$,

where $T[ito]$ is the initial time out inactivity value. The process 200 then returns to the main process 100 flow.

If, at decision block 202, the process determines that the system access time T is not less than the first predetermined value, $T(Inc)$, the process 200 proceeds to process block 214, where it determines if the system access time is greater than a second predetermined period, $T(Dec)$. If so, the process proceeds to decision block 216, where it determines if the DIR FLAG has been set to indicate that $T[to]$ should be decreased. If so, the process proceeds to process block 218, where it sets the DIR FLAG to indicate that $T[to]$ should be decreased. In one embodiment, DIRFLAG is set to a value of "2", indicating that $T[to]$ should be decreased. The process 200 then returns to the main process 100 flow. If, at decision block 216, the process determines that the DIR FLAG has been set to indicate that $T[to]$ should be decreased, it proceeds to decision block 220, where it queries if $T[to]$ is equal to $T[ito]$, the initial time-out inactivity value. If so, the process 200 returns to the main process 100 flow. Otherwise, it proceeds to process block 224, where it increases the rate of change, INDEX by setting $INDEX = B * INDEX$, where B is a constant. In one embodiment, $B = 2$. The process 200 then proceeds to process block 226, where it decreases the time out inactivity period $T[to]$ by setting $T[to] = T[to] - (INDEX * T[ito])$. The process 200 then returns to the main process 100 flow.

If, at decision block 214, the process 200 determines that T is not greater than a second predetermined value, $T(Dec)$, the process 200 proceeds to decision block 228, where it queries if T is within an acceptable range. In one embodiment, the process queries if $T(Inc) < T < T(Dec)$. If so, the process proceeds to process block 230, where it determines if the DIR FLAG has been set to indicate that $T[to]$ should not be changed. If not, the process 200 proceeds to process block 232, where it sets the DIR FLAG to indicate that $T[to]$ should not be changed. In one embodiment, DIRFLAG is set to a value of "0", indicating that $T[to]$ should not be changed. The process 200 then returns to the main process 100 flow. If, at decision block 230, the process 200 determines that the DIR FLAG has been set to indicate that $T[to]$ should be decreased, it returns to the main process 100 flow. In addition, if at decision block 228, the process determines that T is not within the predetermined

acceptable range, i.e., T is not greater than $T(\text{Inc})$ and not less than $T(\text{Dec})$, the process returns to the main process 100 flow.

Figure 6 is a flow chart that illustrates a second embodiment of the IPM process of the present invention. In one embodiment, the IPM process 300 may be implemented for the management of display devices such as video circuits. The IPM process 300 begins from a start state and proceeds to decision block 302, where the process 300 determines if $T[\text{to}]$, the time out inactivity period of a device is greater than $T[\text{inactive_time}]$ a predetermined inactivity period. In one embodiment, $T[\text{inactive_time}]$ is the inactivity period of an I/O device such as a keyboard or a pointing device such as a mouse. In another embodiment, $T[\text{inactive_time}]$ is 10 seconds. If so, the process 300 proceeds to process block 304, where it turns off the device. The process 300 then proceeds to process block 306, where it starts a device interaction timer, which runs for a predetermined period of time, such as 15 seconds. The process 300 then queries if the managed device was accessed during the device interaction timer period, as shown in decision block 310. If so, the process 300 proceeds to process block 312, where it activates the managed device. The process 300 also proceeds to set $T[\text{inactive_time}] = C * T[\text{inactive_time}]$, where C is a predetermined number. In one embodiment, $C = 4$. The process 300 then returns to the main process 100 flow. If, at decision block 310, the process determines that the device was not accessed during the device interaction timer period, the process 300 advances to process block 316, where the interaction timer is reset. The process then returns to decision block 302. If at decision block 302, the process 300 determines that $T[\text{to}]$ is not greater than $T[\text{inactive_time}]$, the process 300 returns to the main process 100 flow.

Figure 7 is a flow chart that illustrates a third embodiment of the IPM process of the present invention. In one embodiment, the IPM process 400 may be implemented for the management of processing circuits such as processing units or CPUs. Beginning from a start state, the process 400 proceeds to process block 402, where it reads the profile identification (ID) of the managed device. The profile ID provides the operational mode of the managed device. The process 400 then proceeds to decision block 404, where it queries if the profile ID of the device

indicates that the device is providing word processing or presentation services. If so, the process sets the device clock to min, which is a fraction of the full operational speed of the managed device. In one example, "min" = 25% of the full operational speed of the managed device. The process 400 then returns to the main process 100 flow.

If, at decision block 404, the process determines that the profile ID of the managed device is neither word processing nor presentation services, it proceeds to decision block 408, where it determines if the profile ID of the device indicates that it is involved in providing spreadsheet services. If so, the process 400 sets the device clock to max, as shown in process block 41, which is a predetermined percentage of the full operational speed of the managed device. In one embodiment, "max" is 100% of the full operational speed of the managed device. The process 400 then returns to the main process 100 flow.

If, at decision block 408, the process 400 determines that the profile ID does not indicate that the device is providing spreadsheet services, it proceeds to decision block 412, where it determines if there is any activity from an I/O device such as a keyboard, a mouse or a disk drive, within a predetermined period. In one embodiment, the predetermined period is 1 minute. If not, the process 400 proceeds to process block 414, where it sets the device clock to min. The process then returns to the main process flow. If, at decision block 412, the process 400 determines that there is activity from an I/O device within the predetermined period, the process 400 proceeds to process block 416, where it sets the device clock to $\text{device_clock} = D * \text{device_clock}$, where D is a predetermined number. The process 400 then returns to the main process 100 flow.

Figure 8 is a flow chart that illustrates a fourth embodiment of the IPM process of the present invention. In one embodiment, the IPM process 500 may be implemented for the management of I/O devices such as floppy disks. Beginning from a start state, the process 500 proceeds to process block 502, where it determines if $T[t_o]$ is greater than a predetermined amount, such as 4 seconds. If so, the device is turned off. The process 500 then returns to the main process 100

flow. If T[to] is less than the predetermined amount, the process 500 directly returns to the main process 100 flow.

The present invention provides an apparatus and method for providing power management for a number of applications in a processor-based system, which facilitates conservation of power in the system, while optimizing system performance.

Although the present invention has been described in terms of certain preferred embodiments, other embodiments apparent to those of ordinary skill in the art are also within the scope of this invention. For example, the processes 100, 200, 300, 400 and 500 described above are illustrative and not restrictive. In addition, the IPM processes 200, 300, 400 and 500 may be implemented separately or in combination. Accordingly, the scope of the invention is intended to be defined only by the claims which follow.

CLAIMS:

What is claimed is:

1. A power management apparatus for a circuit in a processor-based system, comprising:

5 a memory to store instruction sequences by which the processor-based system is processed; and

a processor coupled to the memory, wherein the stored instruction sequences cause the processor to: (a) determine a system access time of the circuit; and (b) determine if the system access time is less than a first predetermined value, if so, increasing an accessibility period of the circuit, during which the circuit is active.

10 2. The apparatus of Claim 1, wherein the stored instruction sequences further cause the processor to: (c) determine if the system access time is greater than a second predetermined value, if so, decreasing the accessibility period of the circuit.

15 3. The apparatus of Claim 1, wherein the stored instruction sequences further cause the processor to: (d) determine if the system access time is within a range between the first and the second predetermined value, if so, setting a flag to indicate that the accessibility period of the circuit should not be changed.

20 4. The apparatus of Claim 1, wherein (a) comprises:
(a.1) monitoring the circuit to determine the system access time of the circuit; and

(a.2) providing the system access time of the circuit to the processor.

25 5. The apparatus of Claim 1, wherein (b) comprises:

(b.1) determining if the system access time is less than a first predetermined value; and

(b.2) if so, increasing an accessibility period of the circuit at a predetermined rate.

30 6. The apparatus of Claim 2, wherein (c) comprises:

(c.1) determining if the system access time is greater than a second predetermined value; and

(c.2) if so, decreasing the accessibility period of the circuit at a predetermined rate.

7. A power management apparatus for a circuit in a processor-based system, comprising:

5 a memory to store instruction sequences by which the processor-based system is processed; and

a processor coupled to the memory, wherein the stored instruction sequences cause the processor to: (a) determine a system access time of the circuit; and (b) determine if the system access time is greater than a first predetermined value, if so, decreasing the accessibility period of the circuit, during which the circuit is active.

8. The apparatus of Claim 7, wherein the stored instruction sequences further cause the processor to: (c) determine if the system access time is less than a second predetermined value, if so, increasing an accessibility period of the circuit.

15 9. The apparatus of Claim 7, wherein (b) comprises:

(b.1) determining if the system access time is greater than a first predetermined value; and

(b.2) if so, decreasing the accessibility period of the circuit at a predetermined rate.

10. The apparatus of Claim 8, wherein (c) comprises:

20 (c.1) determining if the system access time is less than a second predetermined value; and

(c.2) if so, increasing an accessibility period of the circuit at a predetermined rate.

11. A power management apparatus for a circuit in a processor-based system, comprising:

25 a memory to store instruction sequences by which the processor-based system is processed; and

a processor coupled to the memory, wherein the stored instruction sequences cause the processor to: (a) determine if an accessibility period of the circuit is greater than a first predetermined value, if so, terminating power supply

5

10

15

20

25

30

to the circuit a system access time of the circuit; and (b) determine if the circuit is accessed within a predetermined time interval, if so, activating the circuit.

12. The apparatus of Claim 11, wherein the instruction sequences further cause the processor to: (c) increase the first predetermined value by a second predetermined value.

13. A power management apparatus for a circuit in a processor-based system, comprising:

a memory to store instruction sequences by which the processor-based system is processed; and

a processor coupled to the memory, wherein the stored instruction sequences cause the processor to: (a) determine an operational mode of the circuit; (b) operating the circuit at a first predetermined level based on the operational mode of the circuit.

14. The apparatus of Claim 13, wherein the stored instruction sequences further cause the processor to (c) determine if there is a predetermined activity level from an input/output device coupled to the circuit within a predetermined period, if so, operating the circuit at a second predetermined level based on the activity level, otherwise operating the circuit at a third predetermined level.

15. A method for providing power management for a circuit in a processor-based system, comprising:

determining a system access time of the circuit; and

determining if the system access time is less than a first predetermined value, if so, increasing an accessibility period of the circuit during which the circuit is active.

16. The method of Claim 15, further comprising determining if the system access time is greater than a second predetermined value, if so, decreasing the accessibility period of the circuit.

17. The method of Claim 16, further comprising determining if the system access time is within a range between the first and the second predetermined value, if so, setting a flag to indicate that the accessibility period of the circuit should not be changed.

18. The method of Claim 15, wherein (a) comprises:

(a.1) monitoring the circuit to determine the system access time of the circuit; and

(a.2) providing the system access time of the circuit to the processor.

5 19. The method of Claim 15, wherein (b) comprises:

(b.1) determining if the system access time is less than a first predetermined value; and

(b.2) if so, increasing an accessibility period of the circuit at a predetermined rate.

10 20. The method of Claim 16, wherein (c) comprises:

(c.1) determining if the system access time is greater than a second predetermined value; and

(c.2) if so, decreasing the accessibility period of the circuit at a predetermined rate.

15 21. A method for providing power management for a circuit in a processor-based system, comprising:

(a) determining a system access time of the circuit; and

(b) determining if the system access time is greater than a first predetermined value, if so, decreasing the accessibility period of the circuit during which the circuit is active.

20 22. The method of Claim 21, further comprising: (c) determining if the system access time is less than a second predetermined value, if so, increasing an accessibility period of the circuit.

23. The method of Claim 21, wherein (b) comprises:

25 (b.1) determining if the system access time is greater than a first predetermined value; and

(b.2) if so, decreasing the accessibility period of the circuit at a predetermined rate.

24. The apparatus of Claim 22, wherein (c) comprises:

30 (c.1) determining if the system access time is less than a second predetermined value; and

(c.2) if so, increasing an accessibility period of the circuit at a predetermined rate.

25. A method for providing power management for a circuit in a processor-based system, comprising:

5 (a) determining if an accessibility period of the circuit is greater than a first predetermined value, if so, terminating power supply to the circuit a system access time of the circuit; and

(b) determining if the circuit is accessed within a predetermined time interval, if so, activating the circuit.

10 26. The method of Claim 25, further comprising: (c) increasing the first predetermined value by a second predetermined value.

27. A method for providing power management for a circuit in a processor-based system, comprising:

(a) determining an operational mode of the circuit; and

15 (b) operating the circuit at a first predetermined level based on the operational mode of the circuit.

28. The method of Claim 27, further comprising: (c) determining if there is a predetermined activity level from an input/output device coupled to the circuit within a predetermined period, if so, operating the circuit at a second
20 predetermined level based on the activity level, otherwise operating the circuit at a third predetermined level.

29. A computer-readable apparatus, comprising:
a computer-readable medium that stores an instruction sequence which when executed by a processor causes said processor to:

25 determine a system access time of the circuit; and
determine if the system access time is less than a first predetermined value, if so, increasing an accessibility period of the circuit during which the circuit is active.

30 30. The computer-readable medium of Claim 29, wherein the instruction sequences further cause the processor to determine if the system access time is

greater than a second predetermined value, if so, decreasing the accessibility period of the circuit.

31. The computer-readable medium of Claim 30, wherein the instruction sequences further cause the processor to determine if the system access time is within a range between the first and the second predetermined value, if so, setting a flag to indicate that the accessibility period of the circuit should not be changed.

32. The computer-readable medium of Claim 28, wherein (a) comprises:
(a.1) monitoring the circuit to determine the system access time of the circuit; and

(a.2) providing the system access time of the circuit to the processor.

33. The computer-readable medium of Claim 29, wherein (b) comprises:
(b.1) determining if the system access time is less than a first predetermined value; and

(b.2) if so, increasing an accessibility period of the circuit at a predetermined rate.

34. The computer-readable medium of Claim 30, wherein (c) comprises:
(c.1) determining if the system access time is greater than a second predetermined value; and

(c.2) if so, decreasing the accessibility period of the circuit at a predetermined rate.

35. A computer-readable apparatus, comprising:
A computer-readable medium that stores an instruction sequence which when execution by a processor causes the processor to:

(a) determine a system access time of the circuit; and

(b) determine if the system access time is greater than a first predetermined value, if so, decreasing the accessibility period of the circuit, during which the circuit is active.

36. The computer-readable apparatus of Claim 35, wherein the stored instruction sequences further cause the processor to: (c) determine if the system access time is less than a second predetermined value, if so, increasing an accessibility period of the circuit.

37. The computer-readable apparatus of Claim 36, wherein (b) comprises:

(b.1) determining if the system access time is greater than a first predetermined value; and

(b.2) if so, decreasing the accessibility period of the circuit at a predetermined rate. 38. The computer-readable apparatus of Claim 37, wherein

(c) comprises:

(c.1) determining if the system access time is less than a second predetermined value; and

(c.2) if so, increasing an accessibility period of the circuit at a predetermined rate.

39. A computer-readable apparatus, comprising:

A computer-readable medium that stores an instruction sequence which when executed by a processor causes said processor to : (a) determine if an accessibility period of the circuit is greater than a first predetermined value, if so, terminating power supply to the circuit a system access time of the circuit; and

(b) determine if the circuit is accessed within a predetermined time interval, if so, activating the circuit.

40. The computer-readable apparatus of Claim 39, wherein the instruction sequences further cause the processor to: (c) increase the first predetermined value by a second predetermined value.

41. A computer-readable apparatus, comprising:

A computer-readable medium that stores an instruction sequence which when executed by a processor causes the processor to: (a) determine an operational mode of the circuit; and (b) operating the circuit at a first predetermined level based on the operational mode of the circuit.

42. The computer-readable apparatus of Claim 41, wherein the stored instruction sequences further cause the processor to (c) determine if there is a predetermined activity level from an input/output device coupled to the circuit within a predetermined period, if so, operating the circuit at a second predetermined level based on the activity level, otherwise operating the circuit at a third predetermined level.

1/6

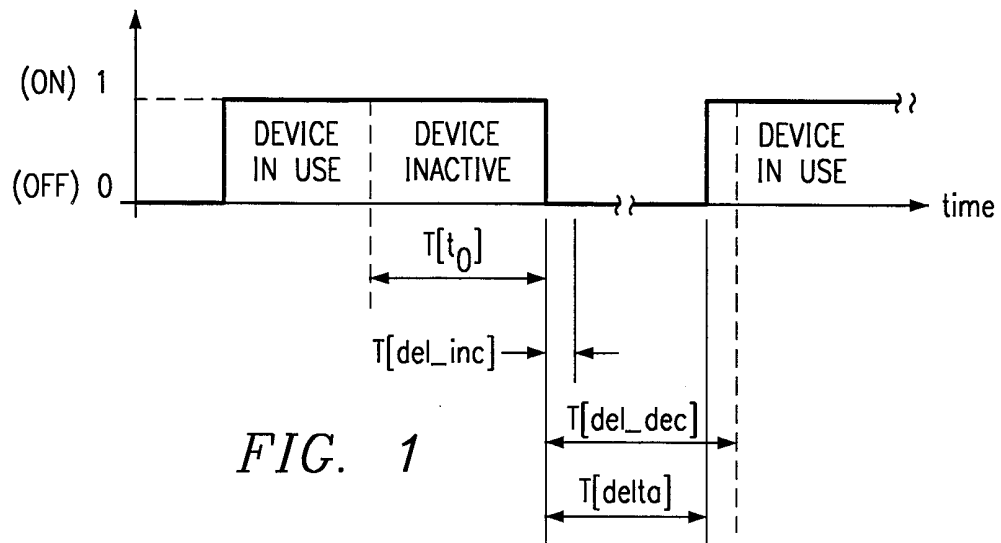


FIG. 1

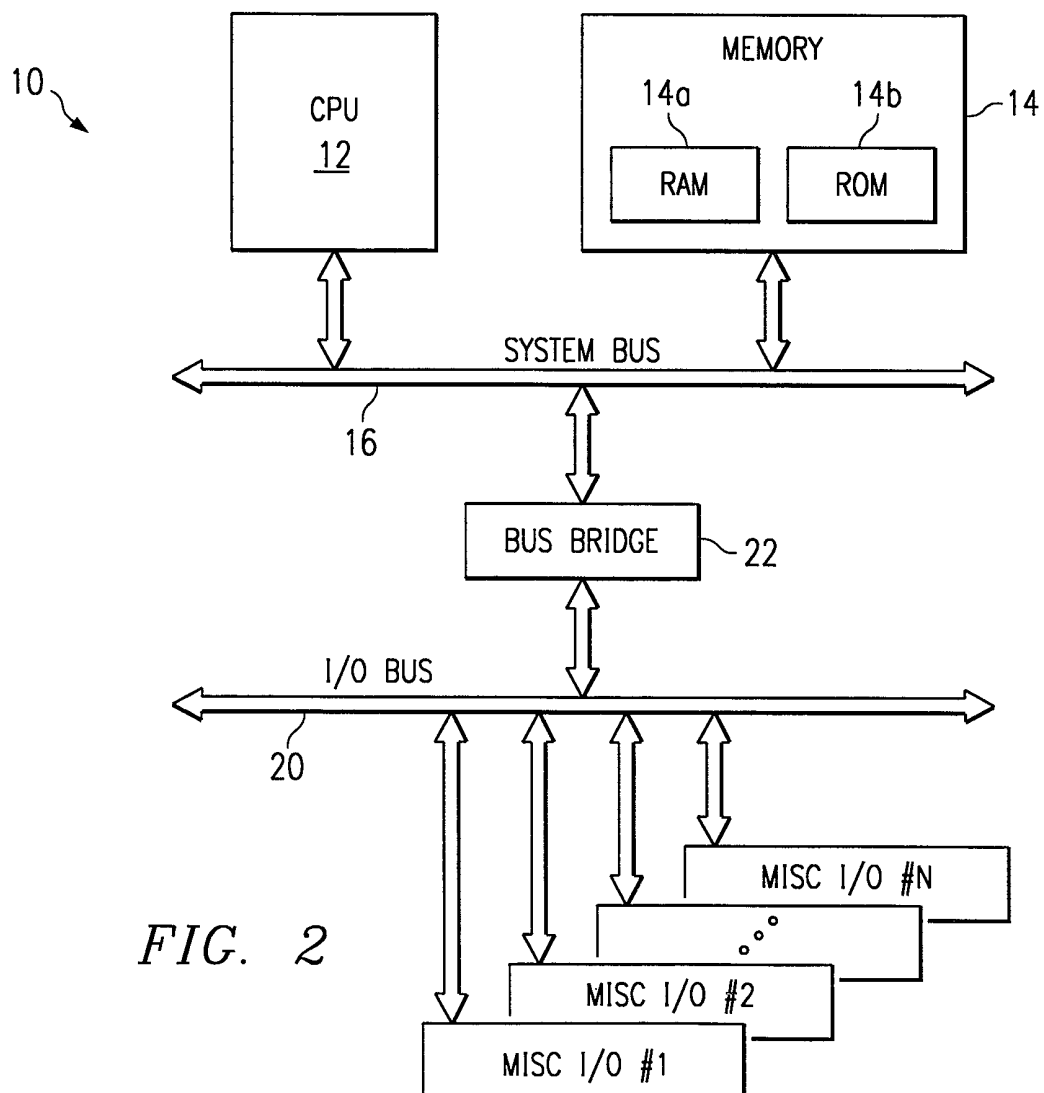


FIG. 2

2/6

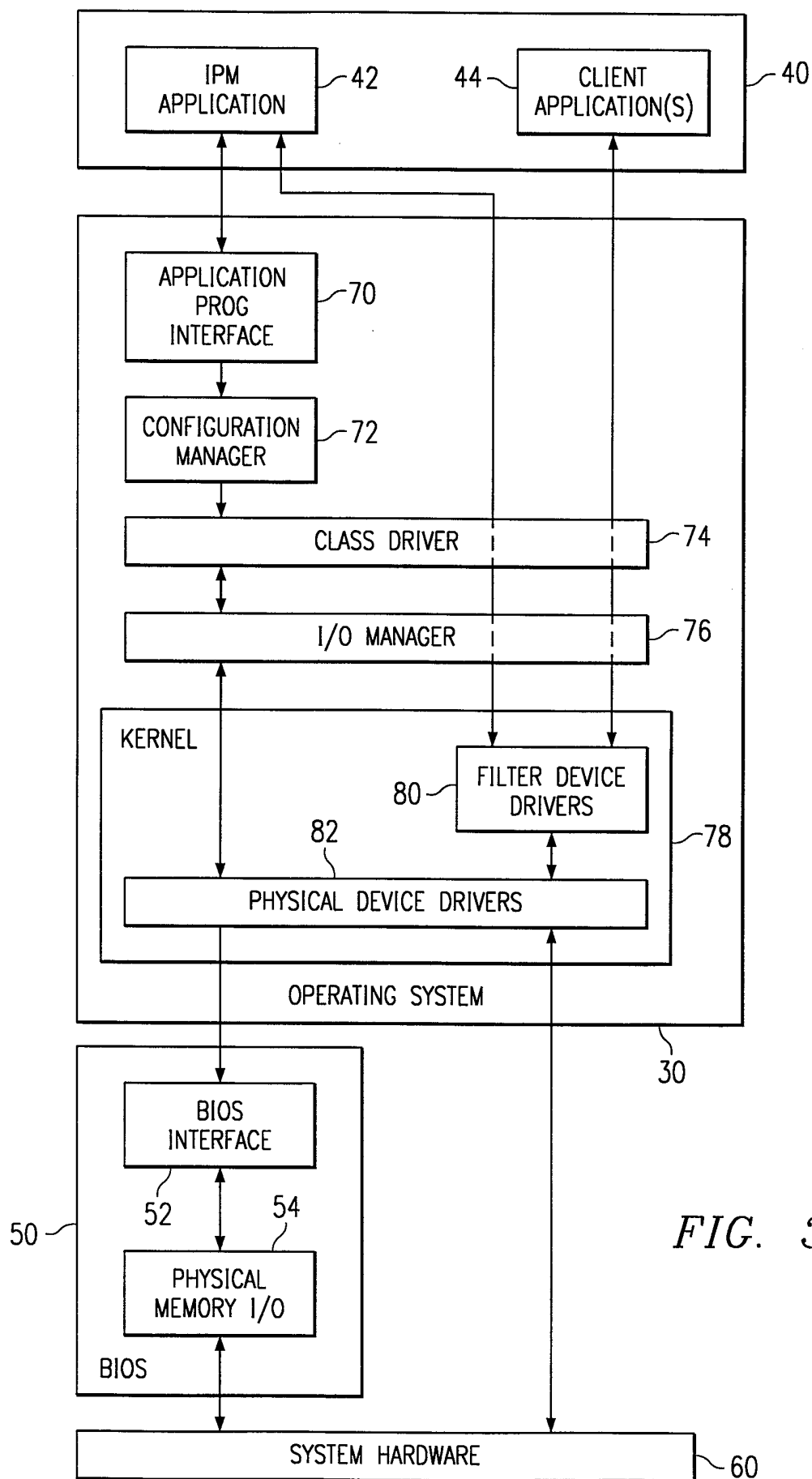


FIG. 3

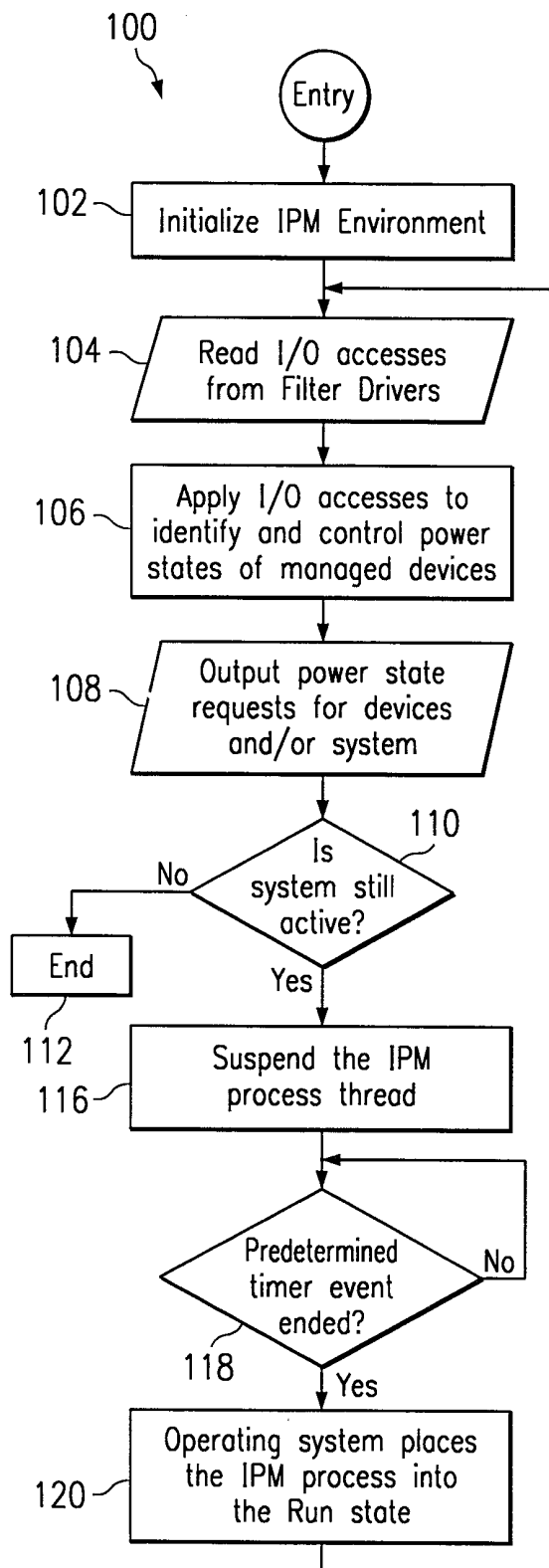


FIG. 4

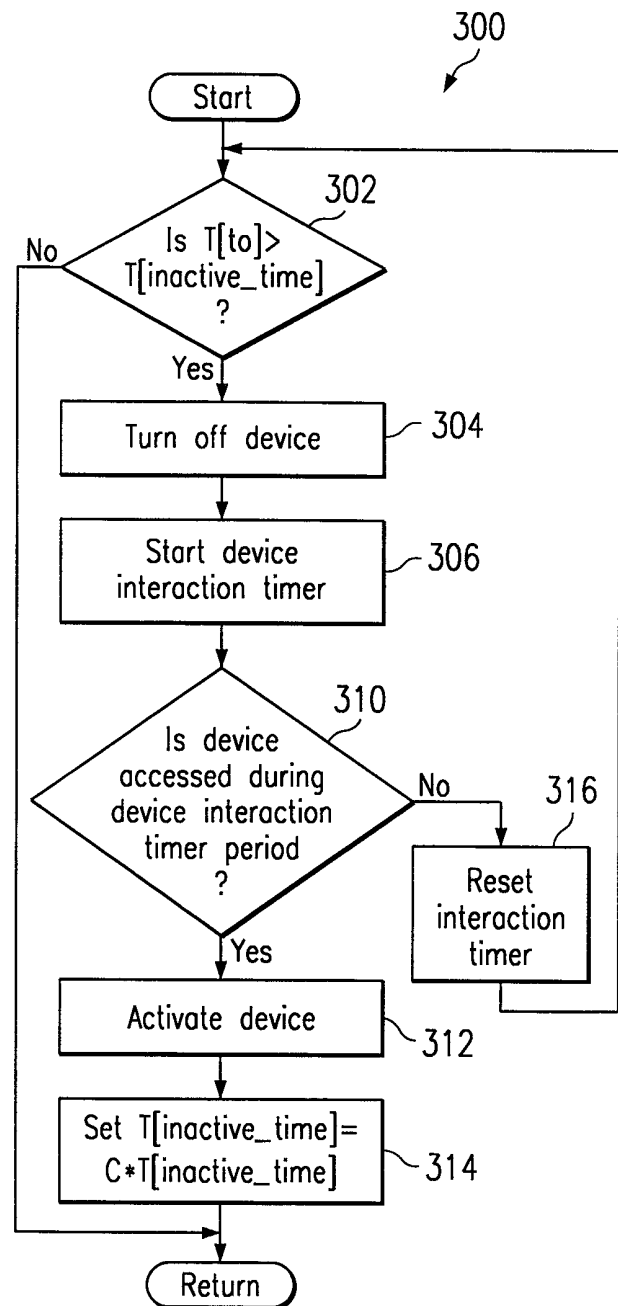


FIG. 6

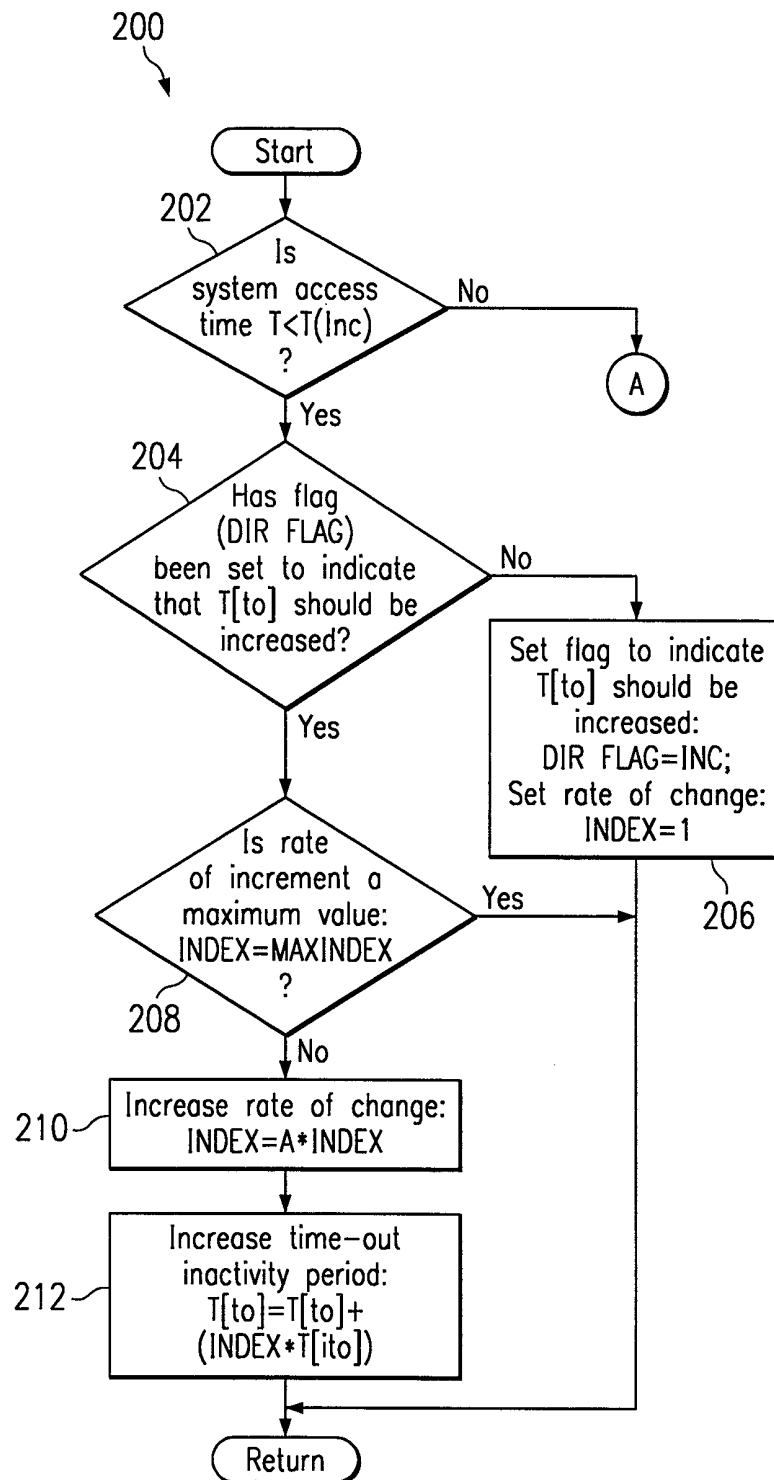
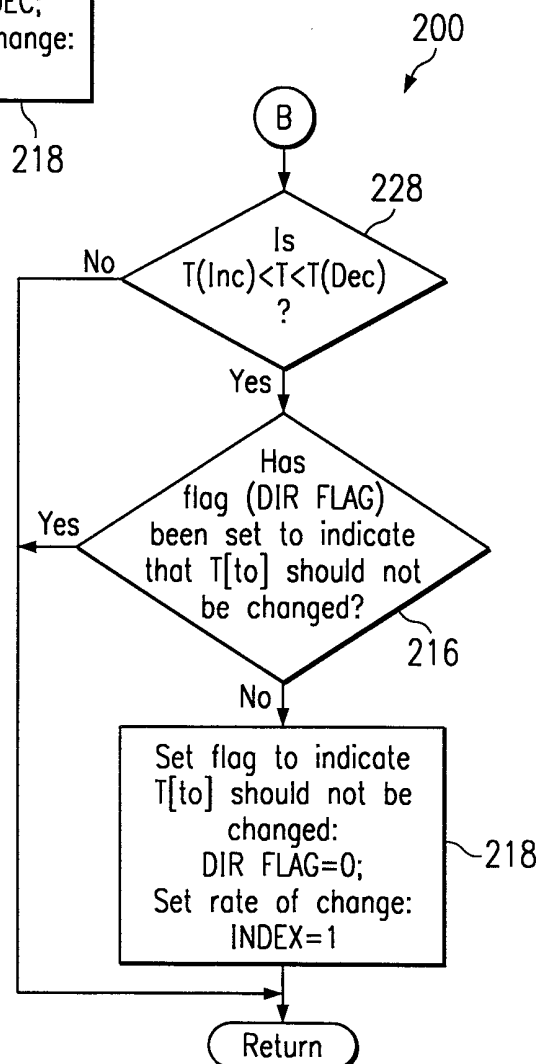
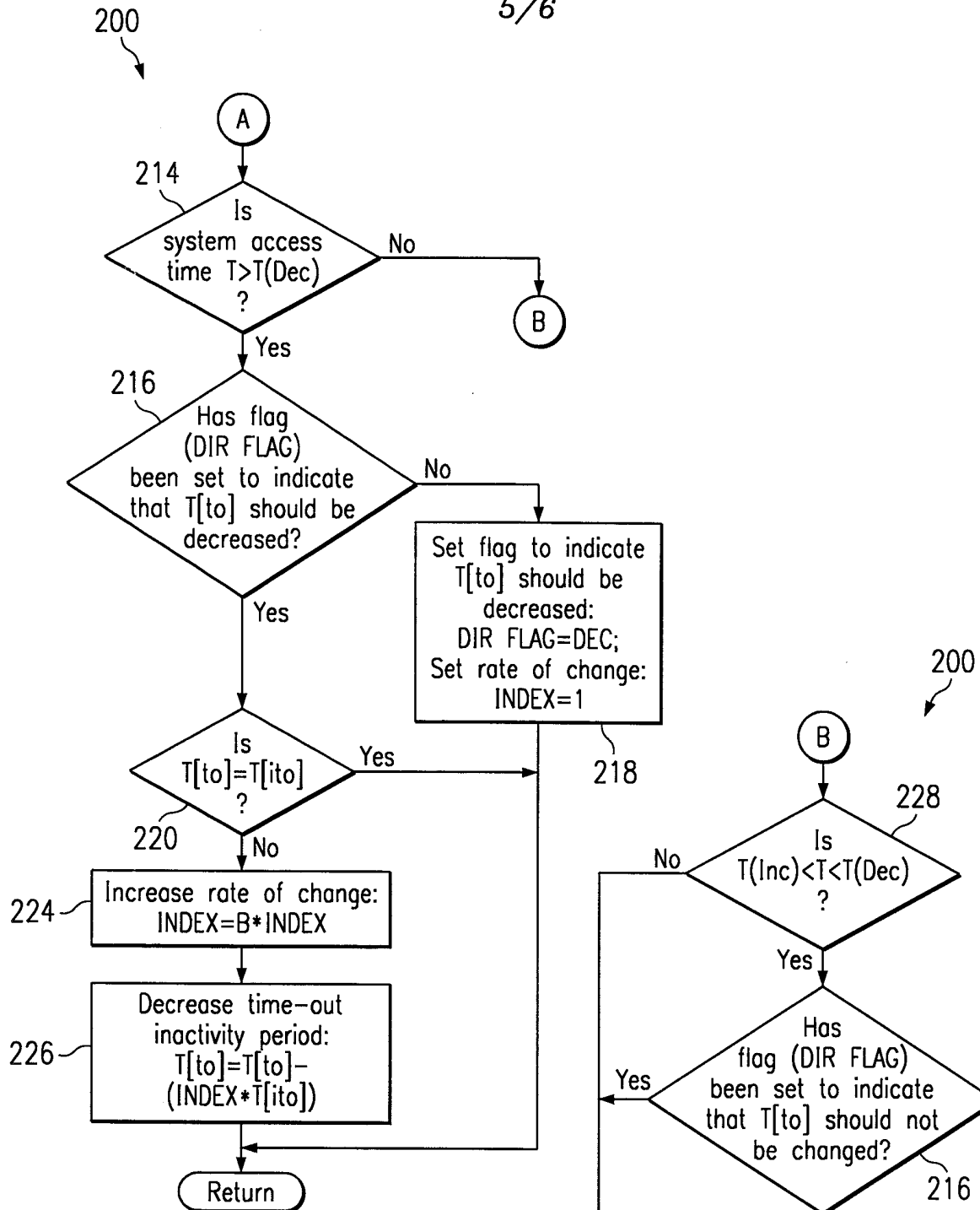
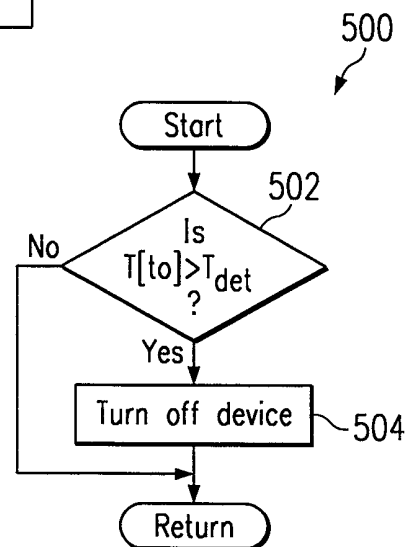
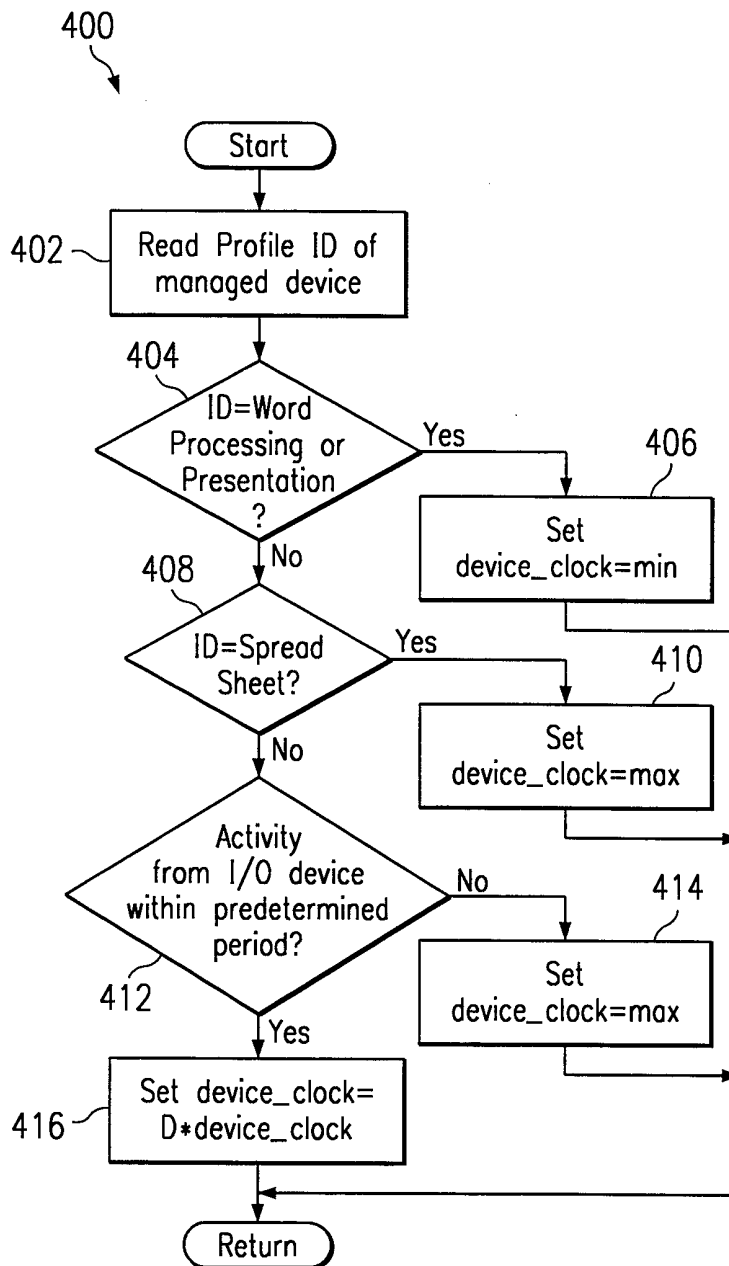


FIG. 5A

5/6



6/6



INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 99/26187

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F1/32

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)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 504 907 A (STEWART GREGORY N ET AL) 2 April 1996 (1996-04-02) the whole document	1-42
X	EP 0 735 457 A (MITSUBISHI ELECTRIC CORP) 2 October 1996 (1996-10-02) column 7, line 12 - column 10, line 18 column 12, line 13 - line 54 figures 2,3,16	1-42
X	EP 0 750 248 A (IBM) 27 December 1996 (1996-12-27) column 12, last paragraph - column 15, paragraph 4; figures 5,6	1-42

☐

Further documents are listed in the continuation of box C.

☒

Patent family members are listed in annex.

* 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 document 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 another 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

10 March 2000

Date of mailing of the international search report

20/03/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Ciarelli, N

INTERNATIONAL SEARCH REPORT

information on patent family members

International Application No

PCT/US 99/26187

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5504907 A	02-04-1996	NONE	
EP 0735457 A	02-10-1996	JP 8272495 A	18-10-1996
		CA 2172925 A	01-10-1996
		CN 1140273 A	15-01-1997
		US 5809316 A	15-09-1998
EP 0750248 A	27-12-1996	JP 9006465 A	10-01-1997
		US 5822597 A	13-10-1998