(12) UK Patent Application (19) GB (11) 2 384 882 (13) A

(43) Date of A Publication 06.08.2003

(21) Application No 0222508.4

(22) Date of Filing 27.09.2002

(30) Priority Data
(31) 0123421 (32) 28.09.2001 (33) GB

(71) Applicant(s)
Lexar Media Inc
(Incorporated in USA - California)
47421 Bayside Parkway, Fremont,
California 94538, United States of America

(72) Inventor(s)
Robert Edwin Payne

(74) Agent and/or Address for Service
Beresford & Co
2-5 Warwick Court, High Holborn,
LONDON, WC1R 5DH, United Kingdom

(51) INT CL[7]
G06F 1/32 , G11C 16/30

(52) UK CL (Edition V)
G4A APWR
G4C C1606S

(56) Documents Cited
GB 2354123 A GB 2347531 A
GB 2335293 A GB 2235797 A
GB 1574058 B US 5606704 B

(58) Field of Search
UK CL (Edition V) G4A
INT CL[7] G06F
Other: Online; EPODOC, JAPIO, WPI.

(54) Abstract Title
Power management system for non-volatile memory acting as a disc storage device

(57) Power management system 20 for a non-volatile memory 14 in a computer is disclosed. The memory system includes a non-volatile memory and a controller 10 which makes the memory appear to the host 12 to have the logical characteristics of a disc drive. The power management system monitors some of the components 21, 22, 24, 26, 28, 30 of the memory controller and their operational activity levels. The power manager then varies the power consumed by selected components of the controller. The power manager could be a discrete system manager. The power manager could comprise an algorithm implemented in firmware. Also the power manager generates the main clock signal memory controller.

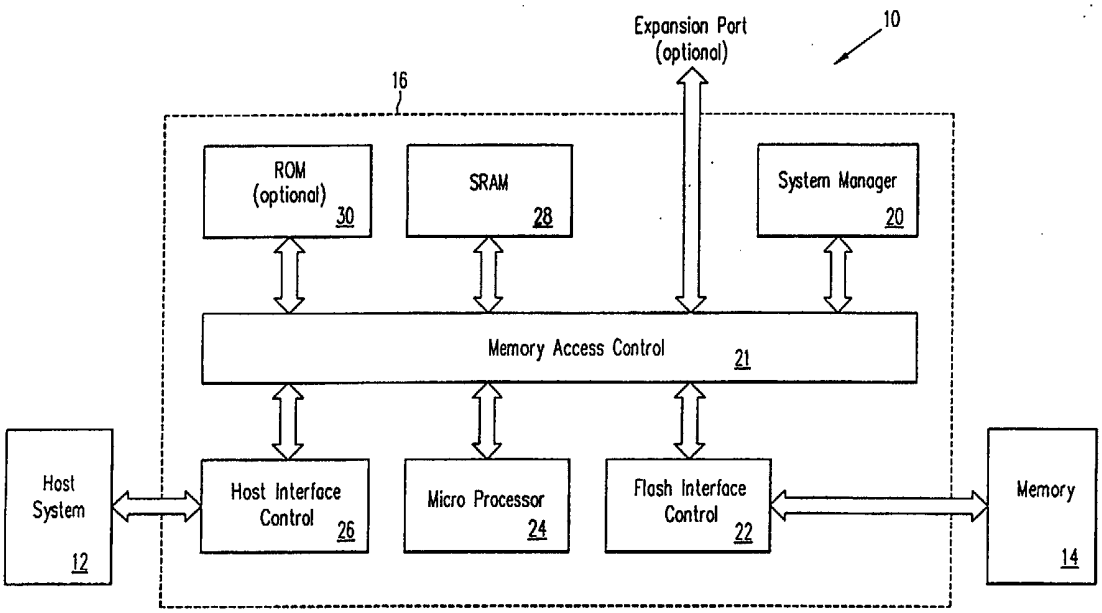


FIG. 1

GB 2 384 882 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

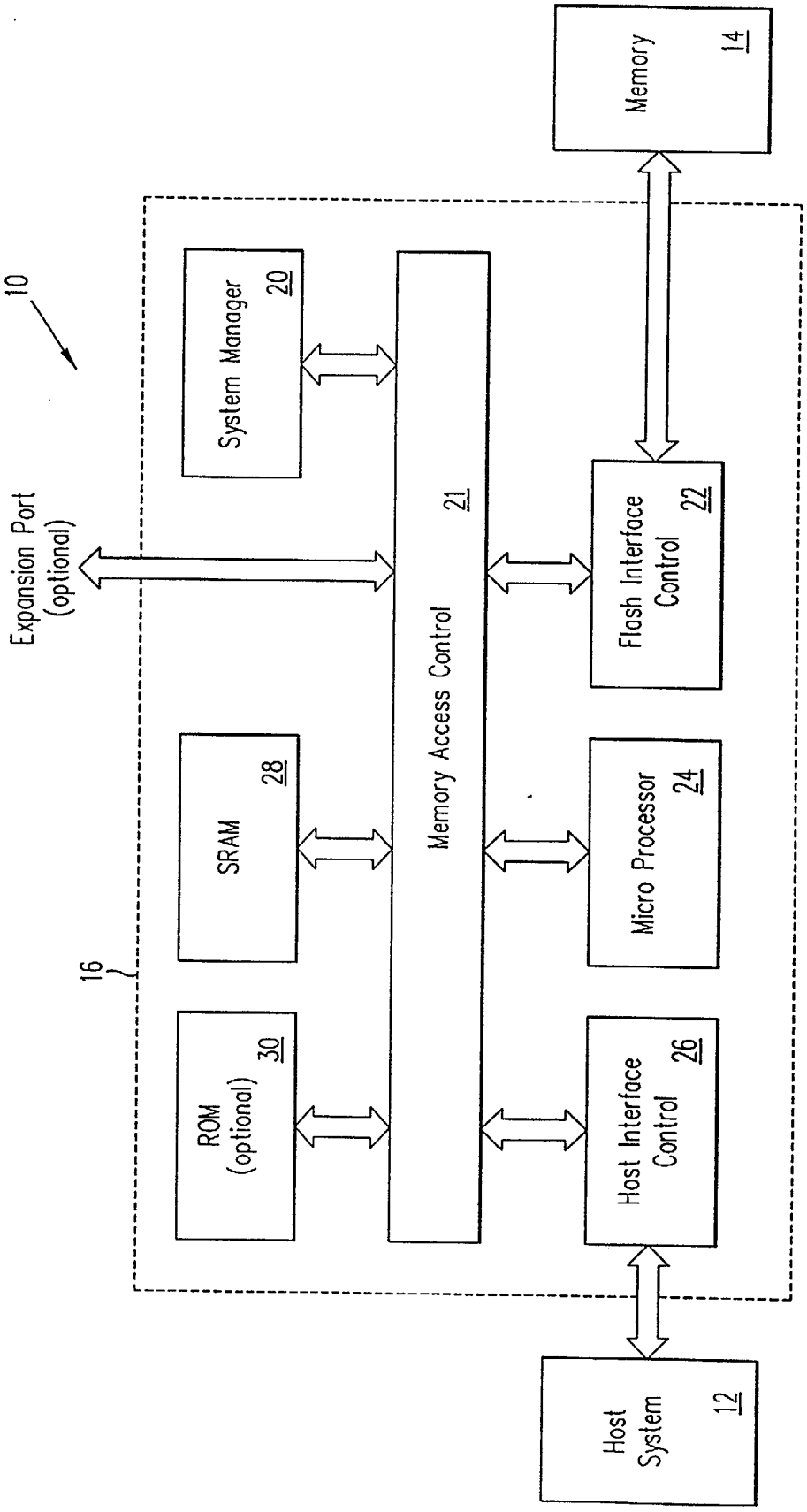

10
Memory
14
System Manager
20
Memory Access Control
21
Flash Interface
Control
22
Expansion Port
(optional)
16
SRAM
28
Micro Processor
24
ROM
(optional)
30
Host Interface
Control
26
Host
System
12

FIG. 1

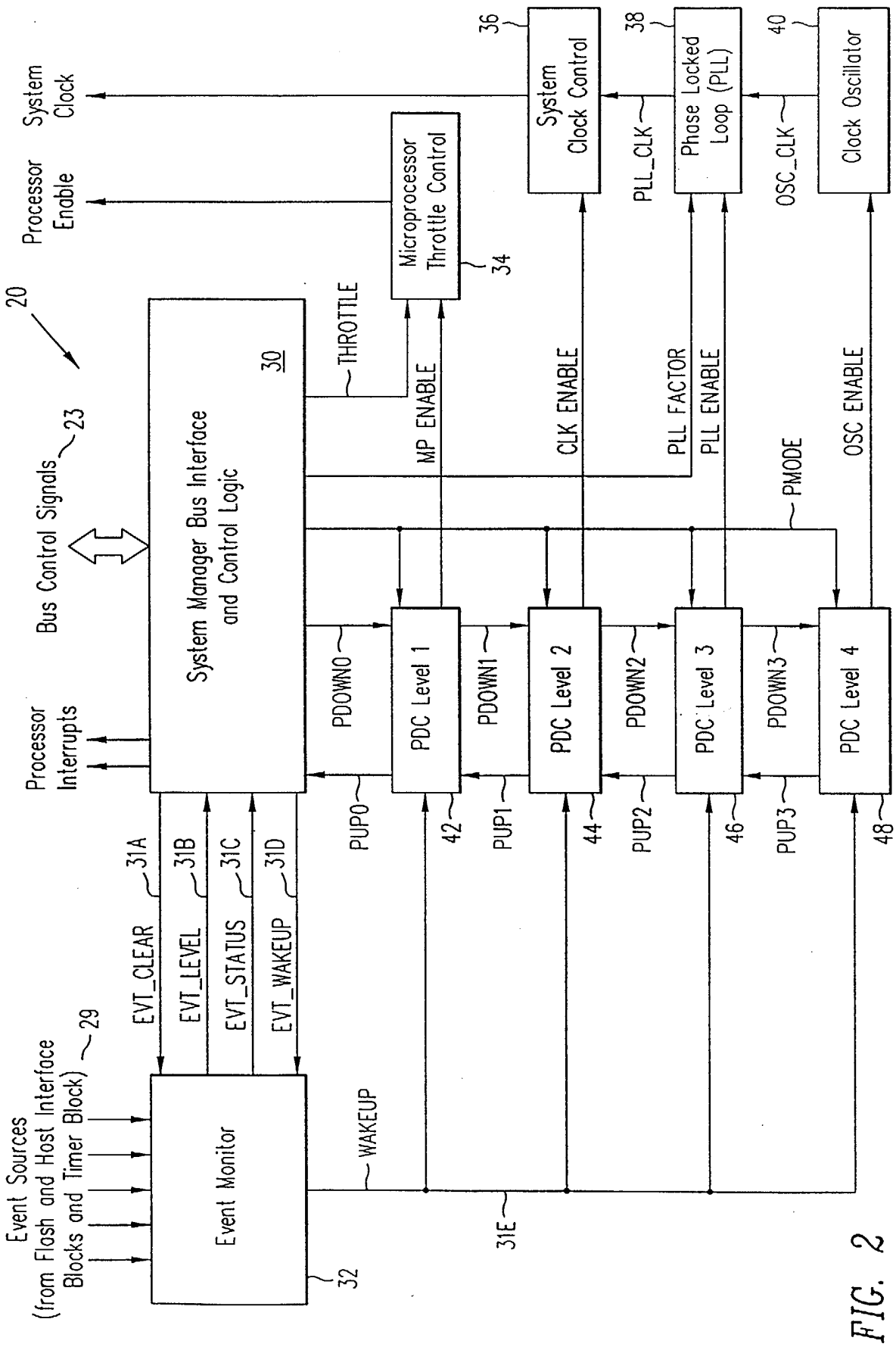

20
System Clock
Processor Enable
36
System Clock Control
38
Phase Locked Loop (PLL)
40
Clock Oscillator
Microprocessor Throttle Control
34
THROTTLE
MP ENABLE
CLK ENABLE
PLL_CLK
PLL FACTOR
PLL ENABLE
OSC_CLK
OSC ENABLE
PMODE
Bus Control Signals
23
System Manager Bus Interface and Control Logic
30
Processor Interrupts
PDOWN0
PDOWN1
PDOWN2
PDOWN3
PDC Level 1
PDC Level 2
PDC Level 3
PDC Level 4
PUP0
PUP1
PUP2
PUP3
42
44
46
48
31A
31B
31C
31D
EVT_CLEAR
EVT_LEVEL
EVT_STATUS
EVT_WAKEUP
Event Sources (from Flash and Host Interface Blocks and Timer Block)
29
Event Monitor
32
WAKEUP
31E

FIG. 2

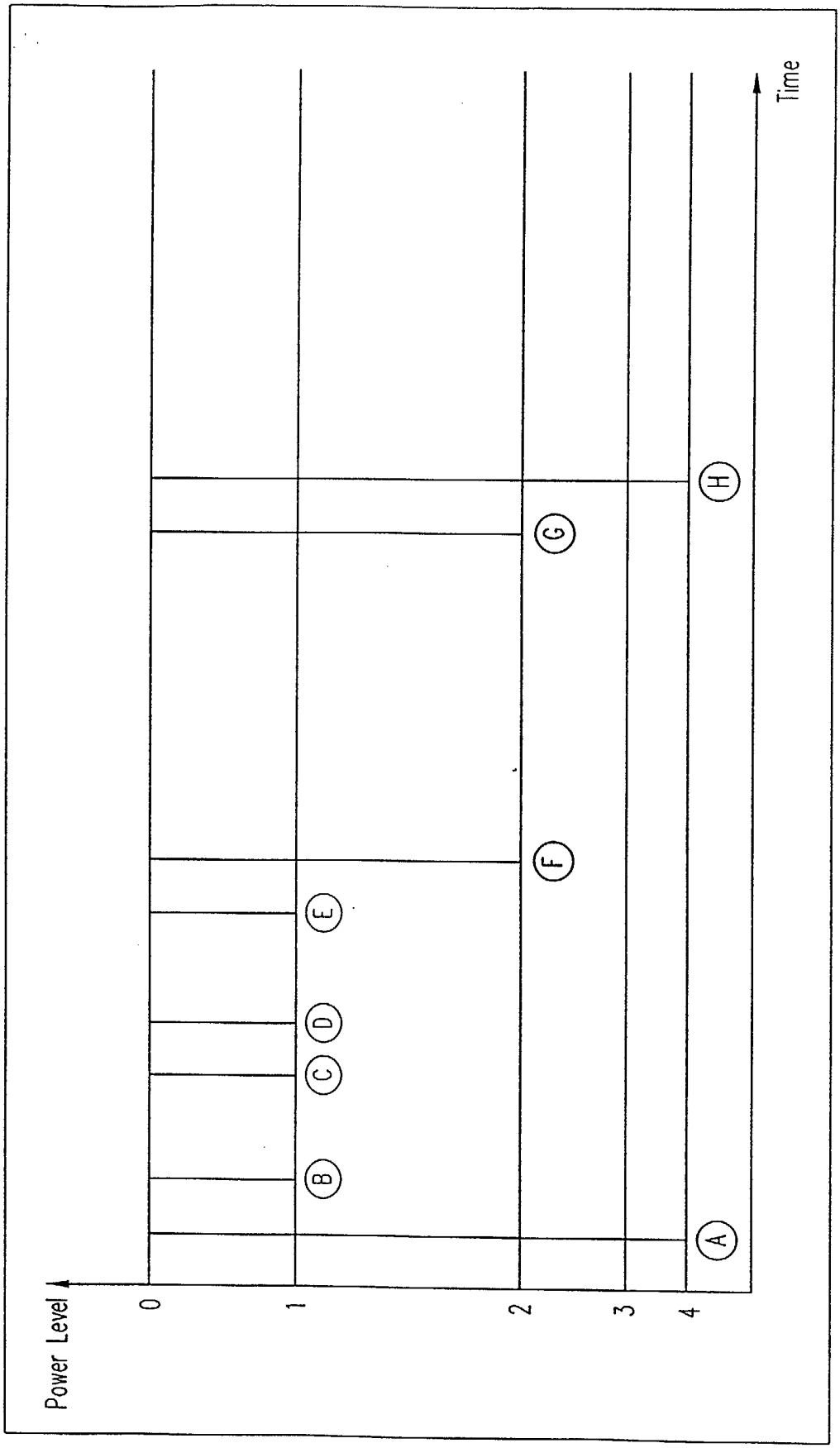


FIG. 3

2384882

Power Management System

This invention relates to a power management system for managing power used and energy consumed in a computer system, particularly a portable computer system where there is often a limit to the peak power that can be supplied and where the energy is usually provided by batteries which have a shorter life if required to deliver increased energy.

This invention is also particularly directed to a flash disc device which is a memory system having a controller which presents the logical characteristics of a disc storage device to a host but which uses a non-volatile semi-conductor memory device as its physical storage medium.

Minimising peak power (where power is energy per unit time) and minimising energy consumption are sometimes conflicting aims. To minimise the peak power drawn by the Flash Storage System may require that the Flash Storage System takes longer to perform its operations, which can lead to higher energy dissipation since the system is active over a longer period, though at a lower power over this period.

The standard Flash Controller consists of a number of hardware blocks. These blocks include a Host Interface

Block, a Flash Interface Block and a Microprocessor block which are connected to memories via a System Bus. Each of these hardware blocks consumes energy within the Flash Controller. The Host Interface and Flash Interface blocks also consume energy on external interfaces. To minimise the energy consumption of the whole computer system requires the minimisation of energy consumption within the Flash Controller itself, within the Flash memory, and on the Flash and Host Interfaces.

According to a first aspect of the invention there is provided a power management system for use in a computer system having a memory system incorporating a non-volatile memory and a controller which presents the logical characteristics of a disc storage device to a host, the power management system comprising means for monitoring the operational activity levels within at least some of the components of the controller and arranged, in response to the monitored levels, to vary the power consumed by selected components of the controller.

Preferably the power management system further comprises at least one power management algorithm which is implemented within firmware of the power management system.

In another of its aspects the present invention

comprises a non-volatile memory system having a controller incorporating a plurality of components and which presents the logical characteristics of a disc storage device to a host, wherein the controller incorporates a power management system having means for monitoring the operational activity levels within at least some of the components of the controller, said means being arranged, in response to the monitored levels, to vary the power consumed by selected components of the controller.

The power management system may be embodied in a discrete system manager or in a distributed manner through components of the controller.

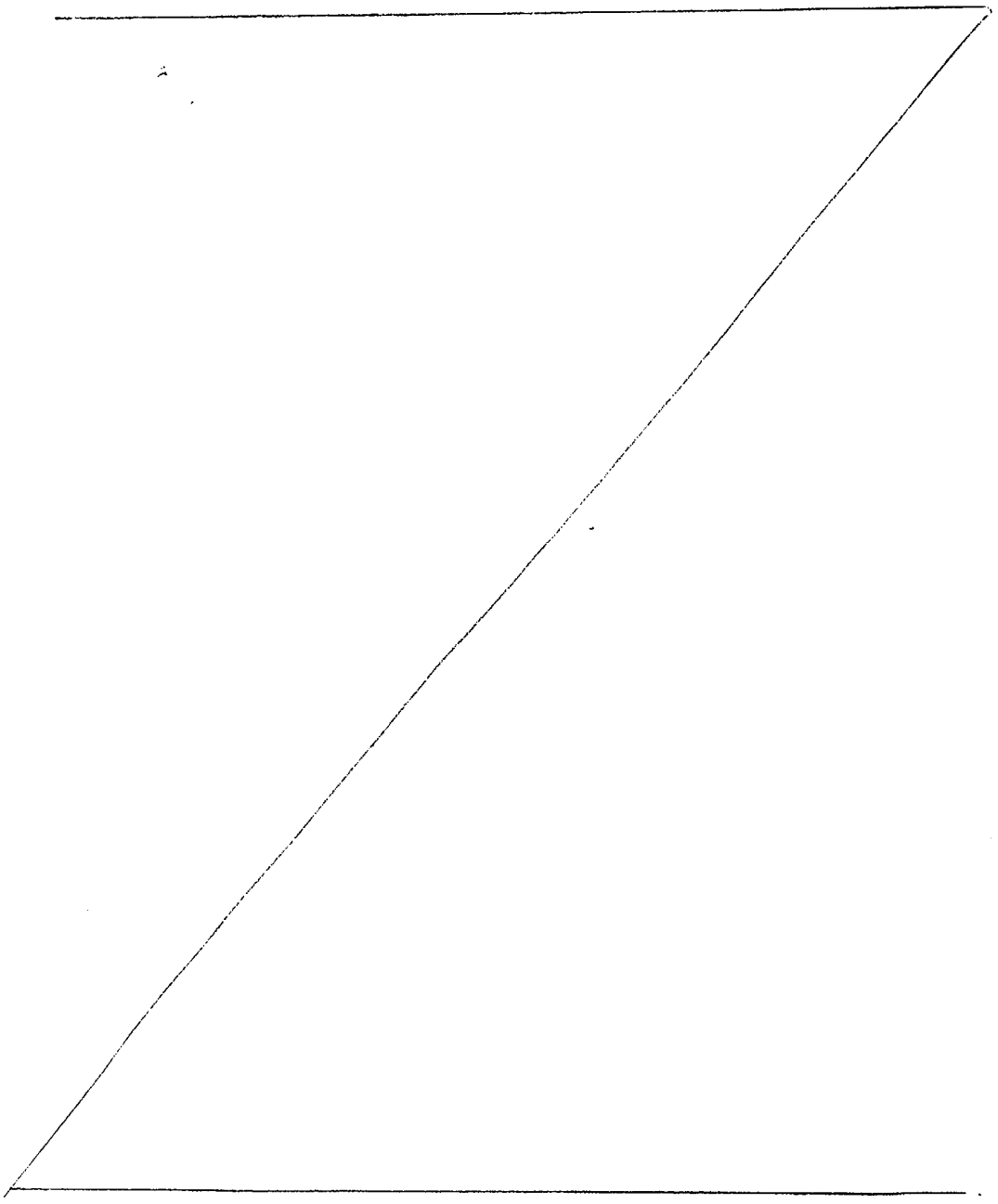
Preferably the power management system generates the main clock signals for the controller and determines which are active and the frequency of such active clock signals.

These and other aspects of the invention will become apparent from the following description taken in conjunction with the accompanying drawings in which:

Fig 1 illustrates a computer system incorporating a power management system in accordance with the present invention;

Fig 2 illustrates the power management system of Fig 1 in greater detail ; and

Fig. 3 illustrates an example of the way in which the controller 16 of Figure 1 switches between different power levels during the execution of a Write Sector command from a host.



As is shown in figure 1 a computer system 10 comprises a flash memory 14, a flash controller 16 and a host system 12. The controller 16 comprises a host interface block 16, a microprocessor 24, a flash interface block 22, an SRAM 28, a ROM 30, all of which are connected to a memory access control structure or system bus 21 in a manner which is well known in the art and which enables the memory system 16, 14, to present to the host system 12 the logical characteristics of a disc storage device.

Controller 16 of system 10 additionally incorporates a power management system in the form of a system manager 20.

Having a discrete System Manager Block 20 simplifies the design and explanation of the power management features, however, the features to be described could equally be distributed and incorporated into other blocks within the hardware of controller 16. The term 'system manager' is intended to embrace both the distributed and the discrete arrangements.

The System Manager 20 is concerned with the control of reset, timing and interrupt signals within the controller 16, and the control logic for these signals may also be incorporated within the System Manager, however, this is not

necessary for achieving power management.

Figure 2 shows the structure of the system manager 20. The system manager 20 comprises a system manager bus interface and control logic block 30 which connects the system manager 20 to the system bus 21 via bus interface 23. The system manager 20 also includes an event monitor block 32, a microprocessor throttle control block 34, system clock control block 36, a phase locked loop (PLL) block 38, clock oscillator block 40 power-down controller (PDC) level 1 block 42, PDC level 2 block 44, PDC level 3 block 46 and PDC level 4 block 48.

The System Manager 20 generates clock signals for the rest of the controller 16. Though one clock is shown, multiple clocks for different parts of the controller 16 may be generated. Whether a clock is enabled, and its frequency, are determined by power management features. The System Manager 20 also generates other control signals to control activity within the Flash Controller 16. One signal enables the microprocessor 24. Firmware reads and writes to memory-mapped registers within the system manager 20 across the interface 30 via the system bus 21.

The Event Monitor 32 takes signals from a number of

event sources 29 within the Flash Controller 16. These event sources 29 indicate when significant system events have occurred within other blocks of the controller 16 such as the Flash and Host Interface Blocks 22, 26 or the Microprocessor 24. For example, the flash and host interface blocks 22, 26 typically indicate when certain operations have completed via these signals. Events that are used to generate processor interrupts or wake-up from a power-down state are listed in Table 3. Synchronous events require the system clock to be running and so will not be generated when the controller is in power-down level 2 or higher. The processor 24 typically indicates when a special event has occurred, such as a request to enter a Debug or Test Mode. The event monitor 32 and the system manager bus interface and control logic block 30 communicate via four channels which carry the signals EVT_CLEAR 31A, EVT_LEVEL 31B, EVT_STATUS 31C, and EVT_WAKEUP 31D.

The Event Monitor 32 feeds the existing level of these events to the Bus Interface and control logic block 30 on the EVT_LEVEL signal, which is composed of one bit per event. The event source signals can be de-asserted by the source at any time. In some cases, when the controller 16 is in a low-power state, it may not be able to respond to

the EVT_LEVEL signal immediately, and could miss an event. Thus, the Event Monitor 32 provides a second copy of the events called EVT_STATUS, that cannot be de-asserted by the source of the event, but can be set (even when the rest of the system is in a low-power state). An event in EVT_STATUS can only be de-asserted by the Bus Control Logic Block 30 asserting the appropriate bit on the EVT_CLEAR signal or a System Reset.

During certain low-power modes, the Event Monitor 32 may be the only active part of the system manager 20. If required, it outputs a signal 31E to the rest of the System Manager 20 called WAKEUP, which causes the rest of the system manager 20 to exit from a low-power mode. The WAKEUP signal is asserted when an event is asserted on EVT_STATUS and the corresponding event bit is asserted on EVT_WAKEUP. Thus, the Bus Interface block 30 can control which events cause the manager 20 to wake-up. Firmware via the Bus Interface 30 block reads the values of both EVT_LEVEL and EVT_STATUS and asserts EVT_CLEAR and EVT_WAKEUP.

Included within the System Manager 20 are blocks 36, 38, 40 for generating the main clock signals for the controller 16 and for enabling other blocks within the controller 16 such as the Microprocessor 24.

The clock generation chain consists of clock oscillator module 40, that generates the fundamental clock for the controller 16 (named OSC_CLK). The frequency of this clock is normally determined by external components such as a Quartz Crystal or a Resistor-Capacitor charging/discharging circuit. The next stage in the clock generation is the PLL (Phase Locked Loop) block 38 which takes the fundamental clock frequency OSC_CLK and multiplies it by a factor to generate the signal PLL_CLK. Finally, this goes into the system clock control block 36, which controls the distribution of the clock to the rest of the controller 16.

The System Manager 20 has four power-down modes that are used to control which clock signals are active within the controller 16. Successive Levels of Power-Down mode turn off more functionality within the controller 16 to save power. The term Power-Down Mode 0 is used to describe normal system operation when all parts of the controller 16 are active. Table 1 illustrates how functionality of the controller 16 is progressively turned off to save power with successive Power-Down Modes.

In Power-Down Mode 1 which is determined by block 42, the Microprocessor 24 is disabled in a controlled fashion, so that other blocks within the controller 16 such as the

Flash and Host Interface Blocks 22, 26 can continue to access memory 14 and perform their functions. The enable signal to the processor 24 is turned off using the MP_ENABLE signal from block 42 that feeds into the Processor Throttling Block 34.

In Power-Down Mode 2 which is determined by block 44, the System Clock 36 is disabled using the CLK_ENABLE signal from block 44. Functions within the controller 16 that rely on the system clock being enabled are disabled and their power dissipation reduced or eliminated. A result of this is that the main system bus 21 will be disabled so that communication between blocks within the controller 16 across the bus 21will be disabled.

In Power-Down Mode 3 which is determined by block 46, the PLL 38 is disabled using the PLL_ENABLE signal from block 46. The PLL 38 may take a certain time to synchronise with the OSC_CLK signal from oscillator 40, so a synchronisation delay , is usually required when the controller is powering up from Power-Down Mode 3 to Power-Down Mode 2.

In Power-Down Mode 4 which is determined by block 48, the clock oscillator 40 is disabled using the OSC_ENABLE signal from block 48. The clock oscillator 40 may take a

certain time to start oscillating again depending on the nature of the external components used to determine the clock frequency, so a delay is required when the controller is powering up from Power-Down Mode 4 to Power-Down Mode 3.

As regards sequencing of power-down and power-up each PDC 42, 44, 46, 48 receives a request for entry to a power-down mode on a PDOWN signal or entry to a power-up mode on a PUP signal. For example, the first PDC 42 will power-down the part of the controller 16 that it controls and then if this is not the target Power-Down Mode (as indicated on the PMODE signal issued by block 30), it will assert its PDOWN signal to the next PDC 44 so that it should power-down and so on.

When the Event Monitor 32 asserts the WAKEUP signal 31E, the PDCs sequentially from the PDC of the target Power-Down Mode will wake-up the part of the controller 16 that it is responsible for. If a delay is required before this part of the controller 16 is ready then the PDCs ensure that this delay is met. The length of the delay may be configured by Firmware writing to registers within the Bus Interface and Control Logic 30. The value of these registers is passed onto the appropriate PDC, which alters the Power-Up delay to reflect the register value. These signals

indicating the length of the delay are not illustrated. Finally the PDC asserts its Power-Up output PUP which causes the next PDC in the chain to wake-up in a similar fashion.

Initially entry to a power-down mode is made by firmware writing to a register within the System Manager Bus Interface and Control Logic block 30, which indicates the desired Power-Down Mode. This causes the PMODE signal to indicate the target power-down level, and the PDOWN0 signal to be asserted which initiates the entry into the Power-Down Mode.

The modular structure of one PDC 42, 44, 46, 48 for each section of the clock generation and processor control 34, 36, 38, 40 allows the structure to be easily adapted for different clock generation structures. In addition, this structure is robust in that it guarantees that the controller 16 is powered-down and powered-up in an orderly manner, so that, for example, the processor 24 is not powered-up before the system clock signal is enabled.

A second power-management feature of the manager 20 is the ability to change the clock frequency by changing the multiplication factor that relates the PLL 38 input frequency to its output frequency. Lowering the clock frequency lowers the power dissipation within the controller

16, but also can reduce the data transfer performance of the controller 16.

To vary the PLL multiplication factor firmware writes to a register within the Bus Interface and Control Logic block 30 which sets the value of PLL_FACTOR that indicates the PLL Multiplication Factor. In some cases, Firmware may want the value of PLL_FACTOR to be reset to a certain value when a system event occurs. For example, if the Firmware sets a low clock frequency to reduce power, but it then wants to process an interrupt quickly, the firmware may not want to continue to run at the low clock frequency. However, it takes a certain amount of time to write to the register that determines PLL_FACTOR. To overcome this problem, the firmware can set a flag within the Bus Interface and Control Logic block 30 which will cause the block 30 to reset the PLL_FACTOR when certain events occur as indicated by the Event Monitor 32.

The Microprocessor 24 is often the main source of power dissipation in the Controller 16 as it consumes power itself and also is the main source of memory access requests within the Controller 16. To allow the power consumption of the Microprocessor 24 to be controlled the System Manager 20 includes a Microprocessor Throttle Block 34.

The Throttle Block 34 controls how often the Microprocessor 24 is enabled. The fewer clock cycles that the Microprocessor 24 is enabled for, the lower the power it consumes. The mechanism used to achieve disablement of the Microprocessor 24 can vary. For example, the output of throttle block 34 can be used directly to disable the Microprocessor 24 or to switch off the clock signal within controller 16 to the Microprocessor 24. Alternatively, the output of the throttle block 34 can be used to deny access of the Microprocessor 24 to the Main-System Bus 21, thus preventing it from fetching instructions and causing it to halt.

The Throttle Block 34 takes two inputs: one is MP_ENABLE from the PDC 42 for Power-Down Mode 1. The MP_ENABLE signal is used to completely disable the Microprocessor 24 when Power-Down Mode 1 is entered. The other input to the Throttle Block 34 is THROTTLE which consists of three values M, S and B which determine the proportion of time that the Microprocessor 24 is enabled. The value of THROTTLE can be changed by the firmware writing to registers within the Bus Interface and Control Logic Block 30.

The values M (Mark) and S (Space) determine the ratio

of clock cycles for which the Microprocessor 24 is enabled and disabled. The B value determines the minimum number of clock cycles in a row that the Microprocessor 24 will be enabled or disabled for. This allows the Microprocessor 24 to gain access to the Memory 14 for a minimum number of clock cycles, since there would be overhead and inefficiencies when enabling and disabling the Microprocessor 24 for too few clock cycles.

Thus, the values M, S and B indicate that the Microprocessor 24 is enabled for M*B clock cycles from every (M+S)*B clock cycles. The hardware interleaves M blocks of cycles with S blocks of cycles in an optimum way to minimise long sequences of cycles with the Microprocessor 24 disabled, which could reduce the Microprocessor 24 responsiveness to events such as interrupts.

The pattern set by M, S, and B is as follows. The pattern starts with a block of B clock cycles with the Microprocessor 24 being enabled and then B clock cycles with the Microprocessor 24 being disabled. The alternation of blocks of the Microprocessor 24 being enabled and disabled repeats up to the minimum value from M and S. If M=S then the pattern now repeats, otherwise, if M>S, then the Microprocessor 24 is enabled for M-S blocks of B clock

cycles and then the pattern repeats but if M<S then the Microprocessor 24 is disabled for S-M blocks of B clock cycles and then the pattern repeats.

The hardware always ensures that the first block within a pattern of enabling and disabling the Microprocessor 24 has the Microprocessor 24 enabled regardless of the value of M, thus setting M=0 is equivalent to M=1 in order to prevent the Microprocessor 24 being never enabled. The hardware also interprets the value of B=0 as the maximum block size allowed by the hardware.

Some examples of the patterns possible of the hardware enabling and disabling the Microprocessor 24 for different values of M, S and B are shown in Table 2.

Other schemes for defining the ratio of cycles for which the Microprocessor 24 is enabled and disabled are possible.

As with the PLL Multiplication Factor feature discussed previously, it is useful to allow Firmware to allow the Microprocessor Throttle 34 to reset the time for which the Microprocessor 24 is enabled to its maximum value when certain system events occurs, to allow for fast reaction to controller events. Firmware can write to a register within the Bus Interface and Control Logic block 30 to enable this

feature.

It will be appreciated that the Flash Interface Block 22 is non standard in that it incorporates features to operate with a range of different main system clock frequencies, since the main clock frequency of the controller 16 may be changed to reduce power consumption by operation of blocks 36, 38 of the system manager 20.

If the main clock frequency of the flash controller is changed then this will affect the timing of signals generated by the Flash Interface Block 22. If the clock frequency is increased then the timing of signals on the flash interface may become too quick for the Flash memory 14. If the clock frequency is decreased then the transfer rate of data to and from the Flash memory 14 will be reduced.

If the Flash Interface 22 is a major source of power dissipation, then it may be advantageous to reduce the transfer rate on the interface to reduce peak power consumption, but this reduces the data transfer rate to and from the Flash memory 14.

Accordingly, to support these power management modes, the Flash Interface Block 22 is designed to allow the timing of signals to and from the Flash memory 14 to be changed

relative to the main controller system clock. Two features in the Flash Interface Block 22 are incorporated to support this. The first feature is a frequency divider circuit that is placed on the main clock that supplies the basic timing reference for the signals on the Flash Interface 22. This allows the speed of the Flash Interface to be reduced to reduce peak power consumption, without affecting the frequency of the main clock. The second feature is that the timing of signals in the Flash interface 22 can be controlled on a clock cycle by clock cycle basis. When the main clock frequency is decreased; this allows the timing of signals to be made quicker by reducing the number of clock cycles for which a signal on the Flash Interface is asserted or de-asserted.

Finally, the Flash Interface Block 22 is designed so that it can use the power management features within the Flash memory 14 which requires the Flash memory select signal to be taken to a voltage close to that of the power supply rail to engage a low power mode.

It will further be understood that the host interface block 26 is different from the flash interface block 22, in that most actions on the host interface block 26 are initiated and timed by the host 12 and not by the controller

16. Many host interface protocols allow the Flash Storage System 14, 16 to indicate at system power-up what host interface timing will be used, but do not allow this timing to be changed later.

Though in most Host Interface protocols, the data transfer rate to and from the Flash Storage System 14, 16 is determined by the Host 12, most host interface transfer protocols allow the Flash Storage System 14, 16 to indicate when it is ready to accept the transfer of data or of a command. The Flash Controller 16 uses this feature of the host interface 26 to control the rate of data and command transfer and thus minimise peak power though this reduces system performance. To support this, the Host Interface block 26 needs to be flexible in when it asserts signals that say if it is ready to accept a command or do a data transfer. If features are incorporated to let hardware automatically set these signals, then the automatic setting of these signals should be configurable, so that flags can be set under direct Firmware control if necessary for power management.

If the protocol allows for the basic timing of a data or command transfer to be slowed down by asserting signals on the interface 26 during the transfer then these should

also be settable by Firmware to allow the transfer rate to be reduced.

Reference has been made to firmware that has to utilise the power management features within the flash controller hardware to minimise power consumption with minimal impact on performance. This will now be explained.

If at a point within the firmware, the firmware has to stop and wait for an event monitored by the Event Monitor 32, then the firmware enables the system manager 20 to wake-up on this event and then enter a power-down mode. The Power-Down Mode powered-down to is determined by activity in other parts of the controller 16. For example, if the Host or Flash interface 26, 22 need the clock to be running to transfer data then Power-Down Level 1 is the maximum level that can be entered. Higher power down levels can be entered if no such constraint exists, but may be limited by the time taken for the Oscillator 40 and PLL 38 to power-up.

Other events that are required to interrupt the processor 24 also trigger the system manager 20 to wake-up. If the system manager 20 is woken-up by an interrupt event, it then responds to the interrupt and then the Firmware returns to the power-down mode selected, if the system event being waited for has not occurred.

Examples of doing this are events such as waiting for the host 12 to issue a command or transfer data, or waiting for Flash memory 14 to complete an operation.

Events within the controller 16 that are not monitored by the Event Monitor 32 cannot cause the system manager 20 to wake-up. In these cases the Microprocessor 24 has to wait, polling a register until the event occurs. In this mode, the Microprocessor 24 must be active but need not run at full speed.

Accordingly using the Microprocessor Throttling mechanism of block 34 can reduce power consumption in this case by reducing the frequency of polling the register. Also when reducing the clock frequency will not affect the performance of other parts of the controller 16 then the PLL multiplication factor of block 38 can be reduced. When the event being polled for has occurred, the firmware can return the controller 16 to its normal operating frequency.

When firmware determines that it needs to limit power consumption on the Host Interface block 26, then it can reduce the power being consumed by indicating to the Host 12 that it is busy, even when it has actually finished an operation or is ready to accept data. During this time, the controller 16 can perform other operations, or the

controller 16 can enter a Power-Down Mode for a period of time to lower power consumption and trigger the system manager 20 to wake-up after a specified time by using an event triggered by a timer within the Flash Controller 16. At this point, the controller 16 can release busy and continue operation. As an alternative to asserting busy, the controller 16, if the host interface transfer protocol permits it, can slow down the host transfer timing. This allows the host 12 to continue data transfer but at a reduced rate.

When the controller cycle time is changed to reduce power, Firmware may choose to adjust the timing of the Flash Interface 22 to use fewer clock cycles to ensure that the transfer rate to memory 14 is maintained.

If Firmware wants to reduce power consumption specifically on the Flash Interface 22 then it can lengthen the timing of commands on the Flash Interface 22, though this will reduce the transfer rate to memory 14. One example, of lengthening Flash Commands is when polling the status of the memory 14. Firmware can lengthen the timing of the polling command, and then set the Flash Interface 22 to trigger an event when the polling command has finished. Firmware can then go to sleep for the duration of the

polling command.

When Firmware enters sections of code that requires the Microprocessor 24 to be active for a long period of time, then peak power consumption can be reduced by using the Microprocessor Throttle Mechanism of block 34 and the PLL Multiplication Factor of block 38.

An example of how the controller switches between different power levels during the execution of a Write Sector command from a host is given with reference to Figure 3. The relative levels of the different power levels are for illustration only. In this example, the controller does not need to respond rapidly to host commands, and the startup times of clock oscillator 40 and phase locked loop 38 during wake-up from power-down levels 4 and 3 are not important. If fast response to a host command is required, it might not be possible to switch to power-down level 4 when the host interface is in the idle state.

At time A the host writes a command to the controller, which generates event 4 shown in Table 3 and causes the controller to wake-up through levels 3, 2 and 1 before the processor starts executing in level 0.

The processor clears the host command event and sets up the DMA hardware to allow the host to transfer data to

the controller. Once the DMA is set up, the controller is put into power-down level 1 at time B. It is not possible to enter a higher power-down level as the DMA transfer requires that the system clock is running.

When the host transfers the required data at time C, event 5 is generated and wakes up the controller to power-down level 0. The processor now sets up the Flash Interface Control to transfer the data to Flash memory and then reverts to power-down level 1 at time D. Again a higher power-down mode cannot be used because the transfer to Flash memory requires that the system clock is running.

When the data transfer to Flash memory completes at time E, event 7 is generated. The controller again wakes up to power down level 0. The processor checks that the transfer was successful, starts the Flash programming operation and then enters power-down level 2 at time F, which halts the system clock.

At time G, the Flash programming operation completes and the Flash busy line makes a low to high transition, which generates event 6. The controller wakes up through power-down level 1 to power-down level 0. The processor checks that the programming operation was successful, sets up the response to the host and powers-down to level 4 at

time H.

Power Down Mode:	0	1	2	3	4
Microprocessor Enabled	Y	N	N	N	N
System Clock Enabled	Y	Y	N	N	N
PLL Enabled	Y	Y	Y	N	N
Oscillator Enabled	Y	Y	Y	Y	N

Table 1 - Power Down Modes

M	S	B	Pattern of Enabling and Disabling Microprocessor *M= microprocessor enabled* *S = microprocessor disabled*
1	1	1	MS
1	1	3	MMMSSS
1	3	3	MMMSSSSSSSSS
4	2	1	MSMSMM
0	2	2	MMSSSS
2	0	2	MMMM

Table 2 - Pattern of Microprocessor Access with varying M,S and B

Event	Synchronous/ Asynchronous	Description
0	Async	*Host Reset 0* Triggered when the host reset goes from logic 1 to logic 0.
1	Async	*Host Reset 1* Triggered when the host reset goes from logic 0 to logic 1
2	Async	*Host Software Reset 0* Host Software reset bit change from 1 to 0 triggers this event.
3	Async	*Host Software Reset 1* Host Software reset bit change from 0 to 1 triggers this event.
4	Async	*Host Command* The host interface block generates this interrupt when a new host command is received.
5	Async	*Host DMA Completion* The host interface block generates this interrupt when a Host DMA Transfer is complete.
6	Async	*Flash Not Busy* Generated when a BUSY signal from Flash memory goes high.
7	Sync	*Flash Interface Control Operation Complete* Generated when completes the current sequence of operations.
8	Sync	*Timer Interrupt* Generated when the internal timer reaches zero.
9	Async	*Host Activity* Generated when activity on Host Interface.

Table 3

CLAIMS

1. A power management system for use in a computer system comprising:

A memory system including,

A non-volatile memory; and

A controller coupled to the non-volatile memory representing the logical characteristics of a disc storage device to a host and including power management means for monitoring the operational activity levels within at least some of the components of the controller and arranged, in response to the monitored levels, to vary the power consumed by selected components of the controller.

2. A power management system as recited in claim 1 wherein the system is embodied in a discrete system manager.

3. A power management system substantially as described with reference to figure 2 of the drawings.

Application No: GB 0222508.4
Claims searched: 1 - 3
Examiner: David P Maskery
Date of search: 28 May 2003

Patents Act 1977 : Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance	
X	1 and 2	GB 2354123 A	(PHOENIX TECH) See pages 3 and 9.
X	"	GB 2347531 A	(IBM) See page 4.
X	"	GB 2335293 A	(COGENCY TECH) See pages 5 and 9.
X	"	GB 2235797 A	(APPLE COMPUTER) See page 5
X	"	GB 1574058	(TOKYO SHIBAURA ELECTRIC) See whole document.
X	"	US 5606704	(INTEL CORP) See whole document.

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC[V]:

G4A, G4C

Worldwide search of patent documents classified in the following areas of the IPC[7]:

G06F, G11C

The following online and other databases have been used in the preparation of this search report:

EPODOC, JAPIO, WPI.