(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 February 2017 (09.02.2017)

(10) International Publication Number
WO 2017/023262 A1

(51) International Patent Classification:
G06F 1/32 (2006.01) *G06F 13/14* (2006.01) *G06F 1/26* (2006.01)

(21) International Application Number: PCT/US2015/043217

(22) International Filing Date: 31 July 2015 (31.07.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).

(72) Inventors: **TRUONG, Binh T.**; 11445 Compaq Center Dr. W., Houston, Texas 77070 (US). **SO, Chi**; 11445 Compaq Center Dr. W., Houston, Texas 77070 (US).

(74) Agents: **MAISAMI, Ceyda Azakli** et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

[Continued on next page]

(54) Title: CONTROLLING POWER TO PORTS

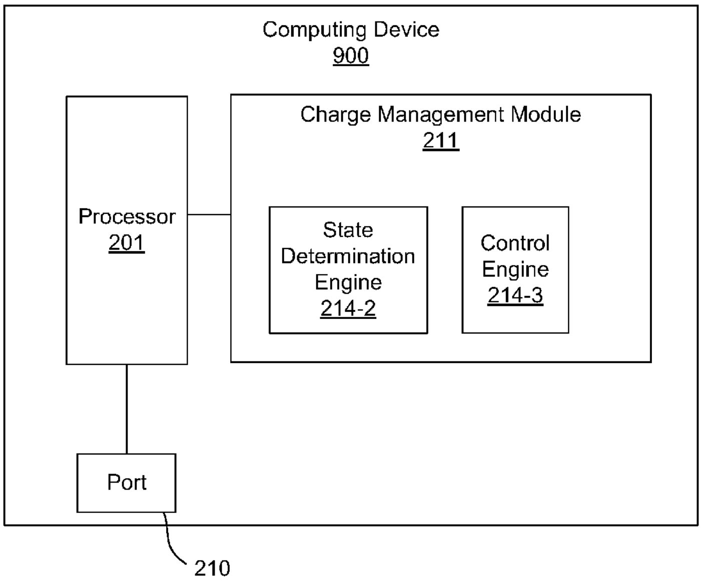

Fig. 9

(57) Abstract: A computing device for controlling power to at least one port includes a processor, and at least one port communicatively connected to the processor. The computing device further includes a charge management module communicatively connected to the processor. The charge management module includes a state determination engine to determine a state of the computing device, and a

[Continued on next page]

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

control engine to control electrical power to the port based at least in part on the state of the computing device and a user-defined time period after disconnection of an alternating current (AC) electrical power supply from the computing device.

CONTROLLING POWER TO PORTS

BACKGROUND

[0001] Many computing devices have ports that connect to and interface with other devices. A portion of such ports implement the Universal Serial Bus (USB) standard. The physical interface specifications of the USB standard allow for a variety of devices to be connected to a computing device using a simple, standardized socket. Hardware devices may use USB or similar ports including, but not limited to, external hard drives, keyboards, mice, wireless network adapters, scanners, printers, and other devices. USB ports may be managed by a host controller. The host controller is a hardware device that serves as an interface between a computing device and the devices connected to the USB ports of the computing device. In addition to managing the flow of data between the host computing device and the other devices, the host controller also provides electrical power from the computing device’s power supply to the devices via the USB ports.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The accompanying drawings illustrate various examples of the principles described herein and are a part of the specification. The examples do not limit the scope of the claims.

[0003] Fig. 1 is a diagram of a computing device in which peripheral devices interface with a computing device through ports, according to one example of the principles described herein.

[0004] Fig. 2 is a block diagram of a computing system including a charge management module within a computing device, according to one example of the principles described herein.

[0005] Fig. 3 is a block diagram of a computing system including a charge management module in the computing device, according to another example of the principles described herein.

[0006] Fig. 4 is a flow diagram of a method of selectively powering ports by a charge management module, according to one example of principles described herein.

[0007] Fig. 5 is a flow diagram of a method of selectively powering ports by a charge management module, according to another example of the principles described herein.

[0008] Fig. 6 is a flow diagram of a method of managing electrical power to ports based on the state of a computing device and a period of time, according to one example of the principles described herein.

[0009] Fig. 7 is a flow diagram of a method of managing electrical power to ports based on the state of a computing device, according to another example of the principles described herein.

[0010] Fig. 8 is a diagram of a computer program product to control power to a number of ports, according to one example of the principles described herein.

[0011] Fig. 9 is a block diagram of a charge management module in a computing device, according to another example of the principles described herein.

[0012] Fig. 10 is a block diagram of a BIOS interface used to configure ports of a computing device, according to one example of the principles described herein.

[0013] Throughout the drawings, identical reference numbers designate similar, but not necessarily identical, elements.

DETAILED DESCRIPTION

[0014] USB standardization provides for some devices to be powered directly from a USB port. With the increasing ubiquity of the USB standard in modern computing devices, it has become increasingly common for devices, particularly small battery-powered peripheral devices such as smart phones, to rely on a port, such as a USB port in a host computing device, as a principal source of electrical power instead of or in addition to a separate power supply.

[0015] As described above, some devices draw power from a host computing device's power supply directly through a USB port, thereby negating the need for a separate, individual power supply for each device. However, in certain scenarios it may be desirable to provide power to such a device through the USB port even when the computing device itself is in a low-power state. In one example, a user may desire for a peripheral device to draw power from the port of the computing device to charge a peripheral device. In another example, a device connected to the computing device may draw power to provide network access for the computing device through the connected device.

[0016] Accordingly, examples described herein provide methods and systems for providing power to a port when a computing device is in a low-power state. A device connected to a port may continue to draw power when a computing device associated with the port is in a low-power state. The device may use the power drawn to charge a battery, continue operation, or similar activities.

[0017] Further, examples described herein provide methods and systems for providing power to a device may use other information related to the computing device. For example, the methods and systems may use the power source of the computing device such as an alternating current (AC) electrical power supply or an electrical energy storage device such as a battery electrically coupled to the computing device to determine when to provide power to the port.

[0018] Still further, examples described herein provide methods and systems for providing power to a device may use other parameters related to the computing device. For example, the methods and systems may provide power to a port for a period of time before turning off power to the port. The period of time to provide power to a port may be adjustable by a user, selectable by a user, definable by a user, or combinations thereof.

[0019] In one example, the firmware of the computing device may control the electrical power to a port. In another example, the firmware may control powering and charging of devices connected to the computing device. In one example, the computing device may include a processor, a port communicatively connected to the processor, and a charge management module communicatively connected to the processor. The charge management module includes a state determination engine and a control engine. The state determination engine determines a state of the computing device. The control engine controls electrical power to the port based on the state of the computing device.

[0020] Still further, examples described herein provide methods for powering ports of a computing device. In one example, the method includes detecting a number of ports associated with a computing device. The method may also include determining whether the computing device associated with the number of ports is in a low-power state. The method includes controlling, for an adjustable period of time, electrical power to each of the number of ports based on the state of the computing device.

[0021] Still further, examples described herein provide a computer program product for charging peripheral devices. In one example, the computer program product includes a non-transitory tangible computer readable storage medium. The non-transitory tangible computer readable storage medium may include computer readable program code embodied therewith. The computer readable program code may include program instructions that, when executed, cause a processor to perform a certain task. The computer readable program code causes a processor to detect a number of ports associated with a computing device. The computer readable program code causes a processor to determine a state of the computing device associated with the number of ports.

The computer readable program code causes a processor to control electrical power to the number of ports based on the state of the computing device.

[0022] As described above, some devices draw power from a host computing device's power supply directly through a port, thereby negating the need for a separate, individual power supply for each peripheral device. However, in certain scenarios it may be desirable to provide power to such a peripheral device through the port even when the computing device itself is in a low-power state.

[0023] To accomplish the above and other goals, the examples described herein include systems and methods for selectively powering devices through ports of a processor when the device is in a low-power state. For example, this may be accomplished by the processor selectively powering at least one port when in a low-power state, in accordance with at least one setting stored by the firmware of the processor. The at least one setting may be obtained through a user-selected option associated with the at least one setting. In one example, the setting may be stored in the Basic Input Output System (BIOS) firmware for the processor. The BIOS may be a part of the firmware. The firmware may be a part of the processor, or may be communicatively connected to the processor. A user may alter the setting stored by the firmware of the processor to enable or disable power to a number of elements of the port control while the processor is in a low-power state. The systems and methods may be implemented by other mechanisms that handle the boot and shutdown of a computing device. By choosing to enable powering to a number of elements in the port control during low-power states of the device, peripheral devices powered through the ports of the processor may continue to operate, receive power, be charged, or combinations thereof, in spite of the low-power state of the computing device. This allows the computing device to reduce its power consumption when in the low-power state while still providing power to the ports. Furthermore, when a user selects disablement of power to the ports entirely during the low-power states of the computing device, energy resources may be conserved.

[0024] As used in the present specification and in the appended claims, the term “firmware” refers to a section of read-only memory programmed with computer program code. A computing device may use firmware to store a portion of the computer program code used to operate the computing device. A computing device may use firmware to store a computer program code to operate the device when a full operating environment is not available. .

[0025] Further, as used in the present specification and in the appended claims, the term “BIOS” refers to a basic input/output system. BIOS may include computer program code that may be stored as a part of firmware. The BIOS may be communicatively connected to the firmware. BIOS controls input and output operations, and may be stored as a part of computer firmware on a computing device.

[0026] Still further, as used in the present specification and in the appended claims, the term “peripheral device” refers to an auxiliary device that connects to the computing device. In one example, a peripheral device may be connected to a computing device to alter or compliment the capabilities of the computing device. In this example, the peripheral device may be a printing device. In another example, a peripheral device may be connected to a computing device in order to charge from the computing device. In this example, the peripheral device may be a smart phone that can charge from a port. In still another example, a peripheral device may be a device connected to the computing device in order to provide additional storage for the computing device. In this example, the peripheral device may be an external storage device.

[0027] Still further, as used in the present specification and in the appended claims, the term “port” refers to an interface by which a peripheral device may be connected to a computing device. A port allows a device to communicatively and physically connect to a computing device. A port may provide electrical power to operate the device, may provide data transfer between the computing device and the peripheral device, or combinations thereof. A port may use an interface protocol, such as universal serial bus (“USB”) interface standard, FIREWIRE International Electrical and Electronics

Engineers ("IEEE") publication 1394 interface standard for a serial bus for high-speed communications and isochronous real-time data transfer developed and distributed by Apple, Inc., or other port interface standards.

[0028] Still further, as used in the present specification and in the appended claims, the term "universal serial bus" ("USB") refers to an industry standard, introduced by the Universal Serial Bus Implemented Forum ("USBIF") that defines the cables, connectors and communications protocols used in a bus for connection, communication, and power supply between computers and electronic devices. A peripheral device may use a USB connector to draw electrical power from and communicate with a computing device.

[0029] Yet further, as used in the present specification and in the appended claims, the term "a number of" or similar language may include any positive number including one to infinity; zero not being a number, but the absence of a number.

[0030] In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present systems and methods. It will be apparent, however, to one skilled in the art, that the present apparatus, systems, and methods may be practiced without these specific details. Reference in the specification to "an example" or similar language means that a particular feature, structure, or characteristic described in connection with that example is included as described, but may not be included in other examples.

[0031] Fig. 1 is a diagram of an example of a computing system (100) having a computing device (101) that interfaces with several external peripheral devices (102, 103, 104, 105) through ports (106, 107, 108, 109). The electrical power to the ports (106, 107, 108, 109) is controlled by a charge management module (111). The ports (106, 107, 108, 109) use a port standard, such as USB. The peripheral devices (102, 103, 104, 105) may include, for example, a wireless network adapter (102), an external hard drive (103), a cell phone (104), and a printing device (105). Each of these peripheral devices (102, 103, 104, 105) may utilize the ports (106, 107) of the computing device (101) to transmit data to and receive data from the computing device (101). Some of the peripheral devices (102, 103, 104, 105) may be powered by their own external power supplies while other peripheral devices (102, 103, 104, 105) may rely on the ports (106, 107, 108, 109) of the computing device (101) to supply electrical power. Smaller peripheral devices (102, 103, 104, 105) such as flash storage devices, mobile phones, PDAs, and media players, among other peripheral devices may rely on the ports (106, 107, 108, 109) to which they are connected for electrical power.

[0032] The external peripheral devices (102, 103, 104, 105) shown in Fig. 1 are examples of devices that may interface with a computing device (101) and are in no way to be construed as limiting the scope of the present specification. Any other device that may suit a particular application of the principles described herein may be used in addition to, or as a substitution for, any of a number of the external peripheral devices (102, 103, 104, 105) shown. Similarly, an external peripheral device (102, 103, 104, 105) may use any protocol or standard to connect to the computing device (101) that allows the external peripheral device (102, 103, 104, 105) to draw electrical power from the computing device (101).

[0033] In one example, the peripheral devices (102, 103, 104, 105) are powered even when a processor of the computing device (101) is in a low-power state. The present specification discloses solutions implemented on the firmware level for powering ports (106, 107, 108, 109) of the computing device (101), such as USB ports, during a processor low-power state so that peripheral devices (102, 103, 104, 105) may continue to draw power from the ports (106, 107, 108, 109).

[0034] When a processor in a computing device (101) boots up, the computing device (101) may allow a user to press a key or combination of keys that will allow the user to enter a setup menu for the BIOS (Basic Input Output System) firmware of the computing device (101). The BIOS firmware is a basic operating system that initializes hardware devices such as the hard drive, video display adapter, and other devices when the computer is first powered on. Among other purposes, the BIOS firmware prepares the processor to allow more comprehensive, higher level operating systems such as Windows or Linux to be loaded, executed, and given control of the processor and computing device (101) as a whole. Thus, a menu option system in the BIOS setup menu may be provided to allow the user to either enable or disable power to a number ports (106, 107, 108, 109) when the processor (101) of the computing device enters a low-power state.

[0035] Fig. 2 is a block diagram of a computing system (200) including a charge management module (211) within a computing device (101), according to one example of principles described herein. The charge management module (211) of the computing device (101) may, when executed by the processor (201), cause the processor (201) to provide electrical power to a number of ports (210-1, 210-2). Even though two ports (210-1, 210-2) are depicted in Fig. 2, any number of ports may be include within the computing device (101) in this example and in in the examples described herein. Powering the USB ports (210-1, 210-2) allows a number of peripheral devices (225-1, 225-2) to receive power from the computing device (101) via the USB ports (210-1, 210-2).

[0036] The computing system (200) includes the computing device (101). Examples of computing devices (101) include servers, desktop computers, laptop computers, personal digital assistants (PDAs), mobile devices, smartphones, gaming systems, tablets, or other electronic devices. The computing device (101) of Fig. 2 may be part of a general purpose computer. However, in alternative examples, the computing device (101) may be part of an application specific integrated circuit. The number of devices (225-1, 225-2) may be connected to the computing device (101) via the number of ports (210-1, 210-2), and may draw electrical power through the number of ports (210-1, 210-2).

[0037] The computing device may include a processor (201) and a charge management module (211). In some examples, the processor (201) and the charge management system (211) are located within the same physical component, such as a server, or a network component. The charge management module (211) may be implemented as hardware, software, firmware, or a combination thereof.

[0038] In one example, the charge management module (211) may be implemented as computer program code, and the examples described herein will be described in the context of the charge management module (211) being implemented in the form of computer program code within software, firmware, or combinations thereof. The computer program code may interact with components of the computing device to control electrical power to the ports. The computer program code may be located in the physical component's main memory, caches, registers, non-volatile memory, or elsewhere in the physical component's memory hierarchy.

[0039] In another example, the charge management module (211) may be located on a separate device in communication with the computing device (210) over a network. Further, the data structures may be accessed from a remote location over a network connection while the programmed instructions are located locally. Thus, the computing device (101) may be implemented on a user device, on a server, on a collection of servers, or combinations thereof.

[0040] In another example, the charge management module (211) may be implemented as part of a read-only memory system providing computer program code to execute on a processor. In this example, the read-only memory system may include firmware or BIOS (205) as depicted in Fig. 2 and as will be described in more detail below.

[0041] In still another example, the charge management module (211) may be implemented as a hardware device as mentioned above. The hardware device may be a component of the computing device (101). The hardware device may read data from memory to control electrical power to the ports (210-1, 210-2).

[0042] The computing device (101) may be utilized in any data-processing scenario, including stand-alone hardware, mobile applications, a computing network, or combinations thereof. Further, the computing device (101) may be used in a computing network, a public cloud network, a private cloud network, a hybrid cloud network, other forms of networks, or combinations thereof. In one example, the methods provided by the computing device (101) are provided as a service over a network by, for example, a third party. In this example, the service may include, for example, the following: a Software as a Service (SaaS) hosting a number of applications; a Platform as a Service (PaaS) hosting a computing platform including, for example, operating systems, hardware, and storage, among others; an Infrastructure as a Service (IaaS) hosting equipment such as, for example, servers, storage components, network, and components, among others; application program interface (API) as a service (APIaaS), other forms of network services, or combinations thereof. The present systems may be implemented on one or multiple hardware platforms, in which the modules in the system can be executed on one, or across multiple, platforms. Such modules can run on various forms of cloud technologies and hybrid cloud technologies or offered as a SaaS (Software as a service) that can be implemented on or off the cloud. In another example, the methods provided by the computing device (101) are executed by a local administrator.

[0043] The processor (201) may include the hardware architecture to retrieve executable code embodied in the charge management module (211) and execute the executable code. The executable code may, when executed by the processor (201), cause the processor (201) to implement at least the functionality of controlling the electrical power delivered to a number of ports (210-1, 210-2). The functionality of the computing device (101) is in accordance to the methods of the present specification described herein. In the course of executing code, the processor (201) may receive input from and provide output to a number of hardware units.

[0044] In one example, the charge management module (211) may be stored as computer program code in any memory capable of storing data such as programmed instructions or data structures used by the computing device (101). The charge management module (211) may include executable program code that is executed by the processor (201) or other processing device. As will be discussed in more detail below, the charge management module (211) may specifically include computer code representing a number of applications that the processor (202) executes to implement at least the functionality described herein.

[0045] The charge management module (211) may interface with various types of memory (221), including volatile and non-volatile memory. For example, the charge management module (211) of the present example may access volatile random access memory (volatile RAM) (216), read-only memory (ROM) (218), and non-volatile random access memory (non-volatile RAM) (220). Many other types of memory may also be utilized, and the present specification contemplates the use of as many varying type(s) of memory as may suit a particular application of the principles described herein. In certain examples, different types of memory may be used for different data storage. For example, in certain examples the processor (202) may boot from read-only memory (ROM) (218), maintain nonvolatile storage in the non-volatile RAM (non-volatile RAM) (220), and execute program code stored in volatile random access memory (volatile RAM) (216).

[0046] Generally, the charge management module (211) may be stored in a computer readable medium, a computer readable storage medium, or a non-transitory computer readable medium, among others. In one example, the charge management module (211) may be implemented as part of a BIOS associated with a processor (201). In another example, the charge management module (211) may include an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples of the computer readable storage medium may include, for example, the following: an electrical connection having a number of wires, a portable computer diskette, a hard disk, a volatile random access memory (RAM) (216), a read-only memory (ROM) (218), non-volatile RAM (NVM) (220), an erasable programmable read-only memory (EPROM or Flash memory), a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain or store computer usable program code for use by, or in connection with, an instruction execution system, apparatus, or device. In another example, a computer readable storage medium may be any non-transitory medium that can contain or store a program for use by, or in connection with, an instruction execution system, apparatus, or device.

[0047] In the computing device (101), a processor (201) communicates with a firmware BIOS (205) stored in the memory (221). The BIOS (205) includes a number of user-alterable settings for selectively enabling power to a plurality of ports (210), consistent with the principles described herein. In one example, the ports (210) are USB ports; however, any other suitably powered port may be used in accordance with the principles described herein.

[0048] The processor (201) includes a plurality of general purpose input/output (GPIO) ports (212) that may be communicatively coupled to the control nodes of a number of hardware switches (215). As such, each of the hardware switches (215-1, 215-2) may open or close based on a digital signal received from a GPIO port (212) in the processor (201). Each of the hardware switches (215) may be disposed between a corresponding port (210) and its power supply. The power supply may be from a USB host controller (Fig. 3, 308) that interfaces with the processor (201) and has a default setting of always providing power to the ports (210) as will be described in more detail in connection with Fig. 3. Thus, depending on the signal from the associated GPIO ports (212) of the processor (201), power may be selectively enabled or disabled to the ports (210). The signals provided by the GPIO ports (212) to the hardware switches (215) may be determined by the aforementioned user-alterable setting(s) stored in one or more of the types of memory (221) with which the BIOS (205) is associated. Any switch including, for example, digitally controlled switches, or other types of switches that best suit a particular application may be used. The functionality provided by the switches (215) and GPIO ports (212) may be obtained using any number of different or additional electronic devices, circuitry, and executable code.

[0049] In one example, the GPIO ports (212) may continue to provide a digital signal to the switches (215) when the processor (201) has entered a low-power state. Additionally or alternatively, the switches (215) may retain a state (i.e., open or closed) set by the GPIO (212) ports and only receive another signal from the GPIO ports (212) when being toggled to another state.

[0050] The computing device (101) of the computing system (200) further includes a number of engines used in the implementation of the charge management module (211). The various engines within the computing device (101) include executable program code that may be executed separately. In this example, the various engines may be stored as separate computer program products. In another example, the various engines within the computing device (101) may be combined within a number of computer program products; each computer program product including a number of engines.

[0051] In the example of Fig. 2, the charge management module (211) includes a number of engines (214). The term "engines" (214) refers to computer program code which, when executed by the processor (201), performs a designated function. As illustrated, the number of engines (214) includes a detection engine (214-1), a state determination engine (214-2), and a control engine (214-3).

[0052] The detection engine (214-1) detects a number of ports (210). The detection engine (214-1) may poll a number of ports (210) by sending a signal through a GPIO (212) port requesting that any ports that receive the signal respond. The response from the port (210) identifies the port that is communicatively connected to the processor (201). The detection engine (214-1) may read system configuration data that identifies a number of ports.

[0053] The state determination engine (214-2) determines a state of the computing device (101). A number of states that the computing device (101) and the processor (201) may exist in will now be described. The computing device (101) may be in an off state, an on state, or a low-power state. These power states may be defined by the Advanced Configuration and Power Interface (ACPI) standard.

[0054] A computing device in an off state does not provide electrical power to any portion of the computing device or peripheral devices associated with the computing device. In one example, the off state may be state S5 state as defined by the ACPI standard. However, in examples described herein, the computing device (101) provides power to a number of ports (210) even when the computing device (101) is in an off state. Thus, in some example, the off state may be included as a low-power state as described herein and recited in the claims. The powering of ports (210) in an off state of the computing device (101) will be described in more detail below.

[0055] A computing device in an on-state may provide electrical power to any portion or peripheral device that may use electrical power. In one example, the on state may be an S0 state as defined by the ACPI standard, in which the processor (201) is in its standard working mode wherein no low-power state is initiated. An example of an on state includes when the computing device (101) is in a running state. A running state indicates that the computing device (101) is executing computer program code on a processor.

[0056] Another example of an on state is when the computing device is configured to be always-on, always-connected (AOAC). An AOAC state indicates that the computing device (101) will be on whenever a power source is available or connected to the computing device (101). In one example, the AOAC state is a state when the computing device does not enter a low-power state as long as electrical power is available. A computing device (101) operating under an AOAC state may use an alternating current (AC) electrical power supply or an electrical energy storage device such as a battery. In an AOAC state wherein the AC electrical power supply is disconnected from the computing device (101), the computing device (101) may exhaust the electrical energy storage device without entering a low power state. In one example, an AOAC state indicates that the computing device (101) will maintain power to communicate with other devices, such as peripheral devices (225) or network devices, but may reduce electrical power to other portions of the computing device (101).

[0057] A computing device (101) in a low-power state may remove electrical power from a number of elements in the computing device. Examples of low-power states include a sleep state, a hibernate state, a hybrid sleep state, a connected-standby (CS) state, a modern-standby (MS) state, sleep states S1, S2, S3, and S4 as defined by the ACPI standard, or other types of low-power states. In one example, a computing device in a low-power state may remove electrical power from a hard disk drive, cause an element of the computing device to operate at a lower performance level, or combinations thereof.

[0058] In an example where the computing device causes an element of the computing device to operate at a lower performance level, the lower performance level causes the element of the computing device (101) to consume less electrical power. The lower performance level may be achieved by reducing the amount of electrical power provided to the element of the computing device (101). This may allow the computing device (101) to operate for a longer period of time while, for example, using an electrical energy storage device. This allows for the user to experience a longer period of use of the computing device (101) as may otherwise be obtained in a non-low-power state.

[0059] Returning again to the description of types of low-power states, a sleep state indicates a low-power state that reduces the consumption of electrical power and allows the computing device (101) to resume operation without rebooting or reissuing instructions. A hibernate state indicates that the computing device (101) causes computer program code to save data prior to entering an off state, and then restores the data upon ending the hibernate state. A hybrid sleep state causes data to be saved prior to entering a low-power state. The hybrid sleep state provides the stable storage of data that hibernate provides, but allows the computing device (101) to quickly recover in a similar manner as a sleep state provides.

[0060] A connected-standby (CS) state indicates an extremely low power consumption while maintaining network connectivity. In one example, a connected standby is a screen-off sleep state. In this example, any time the computing device (101) has an associated display device in an off state, it may be in a connected standby state. While the computing device (101) is in a connected standby state, it can pass through a number of hardware and software operating modes. For most of the time spent in a connected standby, the hardware may be in a low-power state and the software may be paused or stopped. However, the computing device (101) intermittently powers up to process incoming execution calls or perform other application-related background activities.

[0061] A modern-standby (MS) state indicates that the computing device (101) may reduce or eliminate electrical power to some portions of the computing device (101), while allowing other portions to continue to operate. An example of an MS state is when electrical power on the computing device may be reduced as a whole by removing power from some elements of the computing device, while maintaining enough power to receive and process packets on a network adapter. In an MS state, when the computing device (101) receives a certain packet via the network adaptor, the computing device (101) resumes full operation.

[0062] The computing device includes a state determination engine (214-2). In one example, the state determination engine (214-2) may determine the state of the computing device (101). In one example, determining the state of the computing device (101) may be accomplished by determining the ACPI standard global states, device states, processor states, performance states, or combinations these states that the computing device (101) currently experiences. In another example, the state determination engine (214-2) may determine information associated with an electrical power supply for the computing device (101). In still another example, the state determination engine (214-2) may determine when the computing device (101) is using an alternating current electrical power source, such as electrical power provided by a wall socket. In yet another example, the state determination engine (214-2) may determine when the computing device (101) is using an electrical energy storage device, such as a battery, to provide electrical power to the computing device (101). When the computing device (101) is using an electrical energy storage device, the state determination engine (214-2) may determine an amount of electrical energy stored in the electrical energy storage device for purposes of providing power to the ports (210-1, 210-2) or not based on a number of user-definable parameters.

[0063] The computing device includes a control engine (214-3). The control engine (214-3) controls electrical power to the port (210) based on a number of parameters. In the examples described herein, a port (210-1, 210-2) may receive electrical power from the computing device (101) without maintaining data connectivity through the port. In this example, data transfer may not be provided via a number of the ports (210-1, 210-2) between the computing device (101) and the peripheral devices (225-1, 225-2), but electrical power may be maintained via the a number of the ports (210-1, 210-2) so that the peripheral devices (225-1, 225-2) may continue to be provided power from the computing device (101).

[0064] In one example, the control engine (214-3) controls electrical power to the port (210) based on the state of the computing device (101). In one example, the control engine (214-3) may control electrical power to a number of ports (210-1, 210-2, collectively referred to herein as 210) so that the ports (210) maintain electrical power while the computing device (101) is in a low-power state or even when the computing device (101) is in an off state.

[0065] The control engine (214-3) may control a number of additional ports (210). Each port (210) may be controlled independently of the other ports (210). For example, port (210-1) may be controlled to receive electrical power while port (210-2) may be controlled to not receive power. Further, each port may be individually powered or not based on a number user-adjustable, user-selectable, and user definable parameters, or a combination of user-adjustable, user-selectable, and user definable parameters.

[0066] With reference to Fig. 10, a block diagram of a BIOS interface (1000) used to configure ports (210) of a computing device (101), according to one example of the principles described herein, is depicted. The BIOS interface is an example of how a user may be prompted to enter a number of parameters associated with the powering of the ports (210) of the computing device (101). However, any interface within the BIOS or outside of a BIOS access may be presented or made available to a user to define the parameters described herein.

[0067] The interface (1000) is entered at block 1001. The user is presented in the interface (1000) with the option to select from a number of ports (1002, 1003, 1004) to configure. Any number of ports may be displayed in the interface (1000) as designed by the ellipsis between port 2 (1003) and port N (1004). Selection of any one of the of ports (1002, 1003, 1004) will cause the interface (1000) to display a number of state switches (1005, 1006, 1007, 1008, 1009, 1010) in order to configure the selected port (1002, 1003, 1004) based on the state the user wishes to address. For example, if the user wishes to configure port 1 (1002), the user will select a graphic identifying port 1 (1002) and will be presented with graphics identifying the state switches (1005, 1006, 1007, 1008, 1009, 1010) from which the user wishes to configure. In this example, the user may wish to configure how the computing device (101) powers the ports 1 (210-1) when the computing device (101) is in a connected-standby (CS) state. The user will select the state CS switch (1006).

[0068] A user may be presented with a state setting (1011) graphic that allows the user to define or set the parameters as to the selected state switches (1005, 1006, 1007, 1008, 1009, 1010). In the above example, the user, having selected the state CS switch (1006), will be prompted to select a number of parameters associated with port 1 (210-1) and how port 1 (210-1) is powered by the computing device (101) during a connected-standby (CS) state of the computing device (101). These parameters include, for example:

- a period or amount of time before power to the port (210) is disabled;
- a period or amount of time before power to the port (210) is disabled after an alternating current (AC) electrical power supply is disconnected from the computing device (101);
- a power source (e.g. alternating current (AC) electrical power supply or an electrical energy storage device) currently provided to the computing device;
- when the computing device (101) is using an electrical energy storage device, whether the electrical energy storage device drops below a predetermined charge capacity threshold;
- whether the computing device (101) is in an always-on, always-connected (AOAC) state;
- whether the computing device (101) is in an off state;

other parameters, or combinations thereof.

[0069] The user may select any number of these parameters. Once the user selects and approves the parameters at block 1011, the computing device (101) then enables (block 1012) or disables (block 1013) power to the ports (210) based on the parameters selected. Enabling (block 1012) power to the ports (210) includes powering (block 1014) the selected port based on the state setting (1011). Disabling (block 1013) power to the ports (210) includes disconnecting or turning off (block 1015) power to the selected port (210) based on the state setting (1011).

[0070] In one example, the control engine (214-3) may control a port (210) to receive electrical power for a period of time, and disable the electrical power to the port (210) once the control engine (214-3) has determined that the period of time has lapsed. As indicated above in connection with Fig. 10, this period of time is user-adjustable. In this example, the user may be prompted to input or select a number of time increments such as seconds, minutes, quarter hours, half hours, hours, days, other time increments, or combinations thereof. The indicator of the period of time may be a value accessed by the control engine (214-3) and adjusted, selected, and/or defined by a user as described above in connection with Fig. 10. For example, the indicator of the period of time the port (210) is to receive electrical power may be user-specified to be thirty minutes. The port (210) will receive electrical power for thirty minutes after entering a low-power state, and electrical power to the port (210) is turned off after the 30-minute period of time. This allows a user to ensure that the peripheral device (225-1, 225-2) is provided power. For example, the user may desire his or her cell phone (Fig. 1, 104) to completely charge. The user may know that the cell phone (fig. 1, 104) charges within a 30-minute time period if coupled to a port (210) that is powered on. In this example, the user may adjust the period of time the port (210) is to receive electrical power to 30 minutes to ensure that the cell phone (Fig. 1, 104) completely charges despite the computing device (101) being in a low-power state.

[0071] In another example, the period of time may be adjusted by the user to be two hours. The port (210) will receive electrical power for two hours after entering a low-power state. The period of time may be specified by a user to control an amount of time the port is to receive power. In one example, computer program code may specify the amount of time the port (210) is to receive power. For example, computer program code, when executing on a processor, may cause the processor to set a value in memory (221) to control the amount of time a port (210) is to receive power.

[0072] In one example, the user may specify the period of time by setting a value in BIOS. In this example, a user interface (1000) may be presented to a user on a BIOS interface (1000) presented during a booting process of the computing device (101) as described above. The user may interact with the BIOS interface (1000) to adjust, select, or define the period of time the port is to receive power, under what on, off, or low-power states the computing device (101) is to provide power to the ports (210), whether the computing device (101) is to provide power to the ports (210) if an AC electrical power supply us disconnected from the computing device (101), or other parameters described herein.

[0073] In another example, a user interface may be provided outside a BIOS environment. In this example, the user may adjust the parameters described herein without also accessing the BIOS. In this example, a reboot or restart of the computing device (101) may be initiated in order to make the user-selected parameters take effect. A user may be prompted in this example to reboot or restart the computing device (101) in order for the user-selected parameters to take effect.

[0074] In this manner, the user may define the period of time a port (210) is to receive power to be different for different low-power states. Each port may have a different user-defined period of time to provide electrical power for each low-power state. For example, a port (210) may receive power for one hour when the computing device (101) is in a sleep state, and may receive power for thirty minutes when the computing device (101) is in a hibernate state. Further, a port (210) may receive power for a period of time when the computing device (101) is in an off state. Thus, the control engine (214-3) may cause electrical power to be delivered to a port (210) for a period of time before turning the electrical power to the port off. For example, a port (210) may receive electrical power for sixty minutes, and then the port (210) may have electrical power turned off to save electrical power.

[0075] Turing again to Fig. 2, the charge management module (211) detects, with the detection engine (214-1), a number of ports (210-1, 210-2). Port 210-1 and port 210-2 may be detected. The charge management module (211) identifies a state of a computing device (101) associated with the ports (210). In one example, the computing device (101) may be determined to be in a MS state. The charge management module (211) controls the electrical power to the ports (210) based on the determination that the computing device (101) is in an MS state. In this example, the electrical power to port 210-2 may be maintained while the electrical power to port 210-1 may be turned off based on a user-selected parameter associated with the individual ports (210-1, 210-2).

[0076] Referring now to Fig. 3, a block diagram of a computing system including a charge management module (211) in the computing device (101), according to the principles described herein is depicted. The computing device (101) includes a processor (201) executing computer code from the BIOS (205) portion of firmware. The BIOS (205) stores a number of user-adjustable settings for selectively enabling power to a plurality of ports (310), consistent with principles described herein.

[0077] The processor (201) is communicatively coupled to a charge management module (211) which controls the ports (310-1, 310-2, 310-3, 310-4, 310-5, collectively referred to herein as 310). As depicted, the ports (310) are Universal Serial Bus (USB) ports. The USB Ports (310) serve as a physical and software or logical bridge between the processor (201) and a number of peripheral devices (325-1, 325-2, collectively referred to herein as 325). In the present example, when the processor (201) is entering a low-power state, the charge management module (211) may wake the USB host controller (308) after computer program code providing an operating environment has been unloaded from the processor (201). The charge management module (211) may then set a port power bit in the USB host controller (308) for each USB port (310-1, 310-2, 310-3, 310-4, 310-5) to be powered during the low-power state according to the user-adjustable settings stored by the BIOS (205). In this manner, the USB host controller (308) assists in the assignment of powering of the ports (310) such that individual ports (310) function differently and individually based on the user-adjustable settings.

[0078] Fig. 4 is a flow diagram of a method (400) of selectively powering ports (210, 310) by a charge management module (211), according to one example of principles described herein. The present method (400) may be performed by, for example, the computing devices (101) depicted in Figs. 2 and 3. In one example, the method (400) may be used in computing devices (101) where the firmware includes a user-alterable BIOS setting which allows for powering of ports (Fig. 3, 310) while the processor (Fig. 3, 201) is in a low-power state.

[0079] In the method (400), the processor (201) determines (block 401) a device state. Determining (block 401) the device state may result in an indication that the device is in a low-power state such as a sleeps state, a hibernate state, a hybrid sleep state, a CS state, or an MS state. The method (400) includes identifying (block 402) a number of ports to control during a number of different states of the computing device (101). The method (400) includes controlling (block 403) the ports (210, 310) such that the ports (210, 310) maintain an electrical charge during the different states of the computing device (101). In one example, the ports (210, 310) are controlled (block 403) based on the user selected parameters described above in connection with Fig. 10. The method (400) further includes determining (block 404) whether the ports (210, 310) are to be powered on (determination YES, 405) or powered off (determination NO, 407). A port (210, 310) that is powered on (block 406) maintains electrical power during the state of the computing device (101). A port (210, 310) that is powered off (block 408) does not have electrical power during the state of the computing device (101).

[0080] Fig. 5 is a flow diagram of a method (500) of selectively powering ports (210, 310) by a charge management module (211), according to another example of the principles described herein. The method (500) may be performed by, for example, the computing device (101) depicted in Figs. 2 and

3. In one example, the method (500) may be used in computing devices (101) where the firmware includes a user-alterable BIOS setting that allows for powering of ports (210,310) while the processor (201) is in a low-power state.

[0081] In the method (500), the processor (Fig. 3, 201) detects (block 501) a number of ports associated with the computing device (101), and determines (block 502) when the computing device (101) associated with the number of ports is in a low-power state. In response to a determination that the computing device is in a low-power state, the processor (201) controls (block 503) the electrical power to each of a number of ports based on a user-adjustable period of time. In one example, the controlling (block 503) controls a number of peripheral devices that may be used to receive electrical power during a low-power state.

[0082] In relation to the methods described in Figs. 4 and 5, the charge management module (211) detects a number of ports associated with a computing device (101). The charge management module (Fig. 2, 211) may receive identification from a number of ports (210, 310), initiate communication with a number of the ports (210, 310), read data identifying the number of ports (210, 310), or may otherwise gain information about the ports (210, 310) that are present in the computing device (101).

[0083] The charge management module (Fig. 2, 211) determines (block 502) when a port is to receive electrical power during a low-power state by progressively or concurrently selecting each of a number of ports (Fig, 3, 310) and determining whether the selected port (Fig. 3, 310) is currently drawing power. When the selected port (Fig. 3, 310) is currently drawing power, it may be assumed that a device (225, 325) that uses electrical power is connected to the selected port (210, 310). A power bit corresponding to the selected port (210, 310) may be stored within, for example, memory 221, or otherwise set such that the charge management module (Fig. 2, 211) ensures that the selected port (210,310) remains powered when the processor (201) enters the low-power state. Once no additional ports remain for selection, the processor (201) may enter a low-power state.

[0084] Turning again to Fig. 5, the charge management module (211) controls (block 503) electrical power, for a period of time, to each of the number of ports (210, 310) based on the state of the computing device (101). During the period of time, the ports (210, 310) receive electrical power. After the period of time has lapsed, electrical power to the ports (210, 310) turned off based on each port's (210, 310) respective periods of powering time defined by the user.

[0085] Fig. 6 is a flow diagram of a method (600) of managing electrical power to ports based on the state of a computing device and a period of time, according to one example of principles described herein. The method (600) includes detecting (block 601) a number of ports (210, 310) associated with the computing device (101).

[0086] The method (600) includes receiving (block 602) a period of time to provide electrical power to each of the number of ports (210, 310). The period of time received may be configured by a user to be different for each of the number of ports (210, 310) as described above in connection with Fig. 10.

[0087] As described above, the method (600) includes determining (block 603) a state of the computing device associated with the number of ports (Fig. 2, 210) including determining when the computing device (101) associated with the number of ports (210, 310) is in a low-power state. Examples of low-power states include a sleep state, a hibernate state, a hybrid sleep state, a connected-standby (CS) state, a modern-standby (MS) state, sleep states S1, S2, S3, and S4 as defined by the ACPI standard, or other types of low-power states.

[0088] As described above, the method includes controlling (block 604) electrical power to each of the number of ports based on the state of the computing device. As described above, a first port may be powered on when the computing device is in a low-power state. A second port may be powered off when the computing device is in a low-power state.

[0089] Fig. 7 is a flow diagram of a method (700) of managing electrical power to ports based on the state of a computing device, according to one example of principles described herein. The method (700) includes detecting (block 701) a number of ports associated with a computing device.

The detecting (701) may use a system resource, such as a system registry, to determine a number of ports. The detecting (701) may detect a number of ports in the computing device.

[0090] The computing device, when operating in a full-power state, includes providing power (705) to all ports. Each peripheral device may be drawing power from a port. The charge management module (211) determines which ports need power (710) when the computing device (101) is in a low-power state.

[0091] The charge management module (Fig. 2, 211) identifies when a port is drawing power (715). When a port is drawing power, a power bit is set in the charge management module (Fig. 2, 211) for the port (720) to cause the port to draw power during a low-power sate. Additional ports (725) may be identified when the additional bits are set to cause the additional ports to draw power during a low-power state. When each port has had a power bit set in the charge management module (Fig. 2, 211), the computing device (101) enters a low power state (730), with the power bit set for each port that is to maintain power.

[0092] In the examples of Figs. 4 through 7, a low-power state may include an off state. As mentioned above, a computing device (101) in an off state does not provide electrical power to any portion of the computing device or peripheral devices associated with the computing device. In one example, the off state may be state S5 state as defined by the ACPI standard. However, in some examples described herein, the computing device (101) provides power to a number of ports (210) even when the computing device (101) is in an off state. Thus, in some example, the off state may be included as a low-power state as described herein and recited in the claims. Thus, in the methods described in connection with Figs. 4 through 7, when a power state of the computing device (101) is determined, or when the methods determine whether the computing device (101) is at a low-power state, this includes an off state.

[0093] In this example, as the computing device (101) begins to enter an off state, the control engine (214-3) of the change management module (211) sets one or more ports (210) to enable power after the computing device

(101) enters the off state. The user may enter this parameter using the state settings (1011) presented in the BIOS interface (1000). In one example, the user may also select a period of time before turning off power to the port (210) after the computing device (101) has entered the off state. In one example, this off state port powering parameter functions only as long as the AC electrical power supply is connected to the computing device (101). Thus, if the AC electrical power supply were removed during the off state port powering, the power to the port (210) may be disabled. However, in another example, the off state port powering parameter may function even if the AC electrical power supply were disconnected from the computing device (101). In this example, the computing device (101) would continue to power the port (210) in the off state using the power capacity of the electrical energy storage device (i.e., battery) of the computing device (101).

[0094] Fig. 8 is a diagram of a computer program product (804) to control power to a number of ports (210, 310), according to one example of the principles described herein. In this example, the computer program product (804) is connected to processing resources (802) that includes at least one processor (201) and other resources used to process programmed instructions stored on the computer program product (804). The computer program product (804) represents any memory capable of storing data, such as programmed instructions or data structures used by, for example the charge management module (211). The programmed instructions shown stored in the computer program product (804) include a port detector (814-1), a device state determination engine (814-2), and a port power controller (814-3).

[0095] The computer program product (804) includes a computer readable storage medium that contains computer readable program code to cause tasks to be executed by the processing resources (801). The computer readable storage medium may be a tangible and/or physical storage medium. The computer readable storage medium may be any appropriate storage medium that is not a transmission storage medium. A non-exhaustive list of computer readable storage medium types includes non-volatile memory, volatile memory, random access memory, flash memory, electrically erasable program read-only memory, other types of memory, or combinations thereof. The computer program product (804) executes the systems and methods described herein.

[0096] The port detector (814-1) represents programmed instructions that, when executed, cause the processing resource (802) to detect a number of ports (210, 310) associated with a computing device (101). The port detector (814-1) identifies the ports (210, 310) that will be controlled by the port power controller (814-3).

[0097] The device state determination engine (814-2) represents programmed instructions that, when executed, cause the processing resource (802) to determine a state of the computing device associated with the number of ports (210, 310). The device state determination engine (814-2) may determine the state of the computing device (101) hardware, an electrical power supply of the computing device (101), software associated with the computing device (101), or similar factors influencing the functioning of the computing device (101).

[0098] The port power controller (814-3) represents programmed instructions that, when executed, cause the processing resource (801) to control electrical power to the number of ports (210, 310) based on the state of the computing device (101) and other user-selectable, user-definable, and/or user-adjustable parameters. In one example, the port power controller (814-3) may cause electrical power to be provided to the ports (210, 310) during a low-power state. In another example, the port power controller (814-3) may cause electrical power to be provided to the ports (210, 310) for a period of time and then turn off the power to the port. The period of time may be specified by a user via the state setting (1011) presented in the BIOS interface (1000). In one example, the period of time may be specified by the user may be stored as a value in memory. The value specifying the period of time may be set in non-volatile memory. For example, a value of two hours may be stored in non-volatile memory. In this example, when the computing device enters a low-power state, the port continues to receive electrical power for two hours after entering the low-power state.

[0099] In one example, the computer program product (804) for charging peripheral devices includes a non-transitory computer readable storage medium. The computer readable storage medium includes computer readable program code embodied therewith. The computer readable program code includes program instructions that, when executed, cause a processor (201) to detect a number of ports (210, 310) associated with a computing device (101). The program instructions further, when executed, cause the processor (201) to determine a state of the computing device (101) associated with the number of ports (210, 310).

[00100] In one example, the program instructions further, when executed, cause the processor (201) to control electrical power to the number of ports (210, 310) based on whether the computing device is in an always-on, always-connected (AOAC) state or in a connected-standby (CS) state. In this example, in response to a determination that the computing device is in an AOAC state, the user may be prompted to choose whether to maintain power to the ports (210, 310) or not maintain power to the ports (210, 310). The prompting of the user is initiated in this instance to ensure that the user knows that the device is in an AOAC state, and, if the AC electrical power supply should become disconnected from the computing device (101) that the computing device (101) will deplete the electrical power within its electrical energy storage device (i.e., battery). This provides the user with the knowledge as to what may occur to unsaved data on the computing device (101) if the computing device's (101) electrical energy storage device is completely depleted.

[00101] In one example, in response to a determination that the computing device is in a CS state, the computing device (101) may maintain power to the ports (210, 310). In the CS state, the computing device (101) may not deplete the electrical power within its electrical energy storage device if the AC electrical power supply should be disconnected. However, when compared to an AOAC state that requires power to always be consumed from either the AC electrical power supply or the electrical energy storage device, the SC state provides for a much lower power consumption such that depletion of the electrical energy storage device in instances where the AC electrical power supply is disconnected may not occur as rapidly as compared to an AOAC state. However, in this example, a user may still be prompted of the switch to the CS state and/or the disconnection of the AC electrical power supply from the computing device (101) in a CS state.

[00102] Fig. 9 is a block diagram of a charge management module (211) in a computing device (900), according to another example of the principles described herein. The computing device (900) includes a processor (201), and at least one port (210) communicatively connected to the processor. The charge management module (211) is communicatively connected to the processor (201). The charge management module (211) includes a state determination engine (214-2) to determine a state of the computing device (900). The charge management module (211) also includes a control engine (214-3) to control electrical power to the port (210) based at least in part on the state of the computing device (900) and a user-defined time period after disconnection of an alternating current (AC) electrical power supply from the computing device.

[00103] In one example, the control engine (214-3) controls each of the number of additional ports (210) independently based on a number of user-defined parameters associated with each of the additional ports (210). In this manner, a plurality of ports included in the computing device (101) may have different user-selected parameters as described above in connection with Fig. 10. One example of the user-selected parameters is the user-defined time period. In one example, the user-defined time period is user-adjustable via a user interface such as, for example, the BIOS interface (1000) described above in connection with the Fig. 10.

[00104] In one example, the control engine (214-3) further controls the electrical power to the at least one port (210) based on whether the AC electrical power supply is present relative to the computing device as defined by a user-selection. In this example, a user may wish to have the ports (210) of the computing device (900) powering peripheral devices only when the AC electrical power supply is connected to the computing device (900) in order to save the computing device (900) from depleting electrical power within its electrical energy storage device (i.e., battery). In this example, the user may, via the BIOS interface (1000) described above in connection with the Fig. 10, select a state setting (1011) that provides for power to a number of the ports (210) to be disabled if the AC electrical power supply is disconnected from the computing device (900). The user may, however, not care about depletion of the electrical power within its electrical energy storage device, and, instead, prefer the port to remain powered in order to prioritize for the powering or charging of a peripheral device coupled to the computing device (900). In this situation, the user may indicate such a desire via the state setting (1011) of the BIOS interface (1000).

[00105] In another example, the control engine (214-3) further controls the electrical power to the at least one port (210) in a different manner based on whether the computing device is using an electrical energy storage device or the AC electrical power supply is present relative to the computing device. In this example, the state setting (1011) of the BIOS interface (1000) provide for a user to indicate how the user would like the computing device (900) to power the ports (210) in the two different powering states where the AC electrical power supply is coupled to the computing device (900), or where the AC electrical power supply is not coupled to the computing device (900) and the computing device (900) is utilizing the electrical energy storage device (i.e., the computing device's (900) battery). This flexibility in user-definable settings allows a user to account for a myriad of different eventualities in connection with the powering of ports (210).

[00106] Aspects of the present systems and methods are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to examples of the principles described herein. Each block of the flowchart illustrations and block diagrams, and combinations of blocks in the flowchart illustrations and block diagrams, may be implemented by computer usable program code. The computer usable program code may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the computer usable program code, when executed via, for example, the processor (Fig. 2, 202) of the computing device (101) or other programmable data processing apparatus, implements the functions or acts specified in the flowchart and/or block diagram block or blocks. In one example, the computer usable program code may be embodied within a computer readable storage medium; the computer readable storage medium being part of the computer program product. In one example, the computer readable storage medium is a non-transitory computer readable medium.

[00107] A device that may provide electrical power to ports allows peripheral devices to be charged or operated while a computing device is in a low-power state. Peripheral devices may be charged with a reduced impact to the power supply of the computing device.

[00108] The preceding description has been presented to illustrate and describe examples of the principles described. This description is not intended to be exhaustive or to limit these principles to any precise form disclosed. Many modifications and variations are possible in light of the above teaching.

CLAIMS

WHAT IS CLAIMED IS:

1. A computing device for controlling power to at least one port, the computing device comprising:

a processor;

at least one port communicatively connected to the processor;

a charge management module communicatively connected to the processor, the charge management module comprising:

a state determination engine to determine a state of the computing device; and

a control engine to control electrical power to the port based at least in part on the state of the computing device and a user-defined time period after disconnection of an alternating current (AC) electrical power supply from the computing device.

2. The computing device of claim 1, wherein the control engine controls the electrical power to the at least one port based at least in part on whether the computing device is in a low-power state or off state,

wherein, in response to the determination engine determining that the computing device is in a low-power state, providing power to the at least one port, and

wherein, in response to the determination engine determining that the computing device is entering an off state, removing power from the at least one port.

3. The computing device of claim 2, wherein the control engine controls each of the number of additional ports independently based on a number of user-defined parameters associated with each of the additional ports.

4. The computing device of claim 1, wherein the user-defined time period is user-adjustable via a user interface.

5. The computing device of claim 1, wherein the control engine further controls the electrical power to the at least one port based on whether the AC electrical power supply is present relative to the computing device as defined by a user-selected parameter.

6. The computing device of claim 1, wherein the control engine further controls the electrical power to the at least one port in a different manner based on whether the computing device is using an electrical energy storage device or whether the AC electrical power supply is present relative to the computing device.

7. The computing device of claim 4, wherein the user interface is a BIOS user interface.

8. A method for controlling power to a number of ports, the method comprising:

detecting a number of ports associated with a computing device;

determining when a computing device associated with the number of ports is in a low-power state; and

in response to a determination that the computing device is in a low-power state, controlling electrical power to each of the number of ports for a user-adjustable period of time.

9. The method of claim 8, further comprising prompting a user to select the user-adjustable period of time via a BIOS interface.

10. The method of claim 9, wherein a different period of time is user-definable for each of the number of ports, individually.

11. The method of claim 9, wherein determining when a computing device associated with the number of ports is in a low-power state comprises:

determining whether the low-power state is an off state or another low-power state; and

in response to a determination that the low-power state is the off state, prompting a user to choose whether to maintain power to the ports or not maintain power to the ports before entering the off state.

12. A computer program product for controlling power to a number of ports, the computer program product comprising:

a non-transitory computer readable storage medium, said computer readable storage medium comprising computer readable program code embodied therewith, said computer readable program code comprising program instructions that, when executed, cause a processor to:

detect a number of ports associated with a computing device;

determine a state of the computing device associated with the number of ports; and

control electrical power to the number of ports based on whether the computing device is in an always-on, always-connected (AOAC) state or in a connected-standby (CS) state;

in response to a determination that the computing device is in an AOAC state, prompting a user to choose whether to maintain power to the ports or not maintain power to the ports; and

in response to a determination that the computing device is in a CS state, maintain power to the ports.

13. The computer program product of claim 12, wherein controlling the electrical power to the number of ports based on the state of the computing device controls the electrical power to each of the ports independently.

14. The computer program product of claim 12, further comprising computer readable program code that, when executed, cause the processor to prompt a

user via a user interface to select a number of state settings for each of the number of ports, individually.

15. The computer program product of claim 14, wherein the user interface is a BIOS user interface.

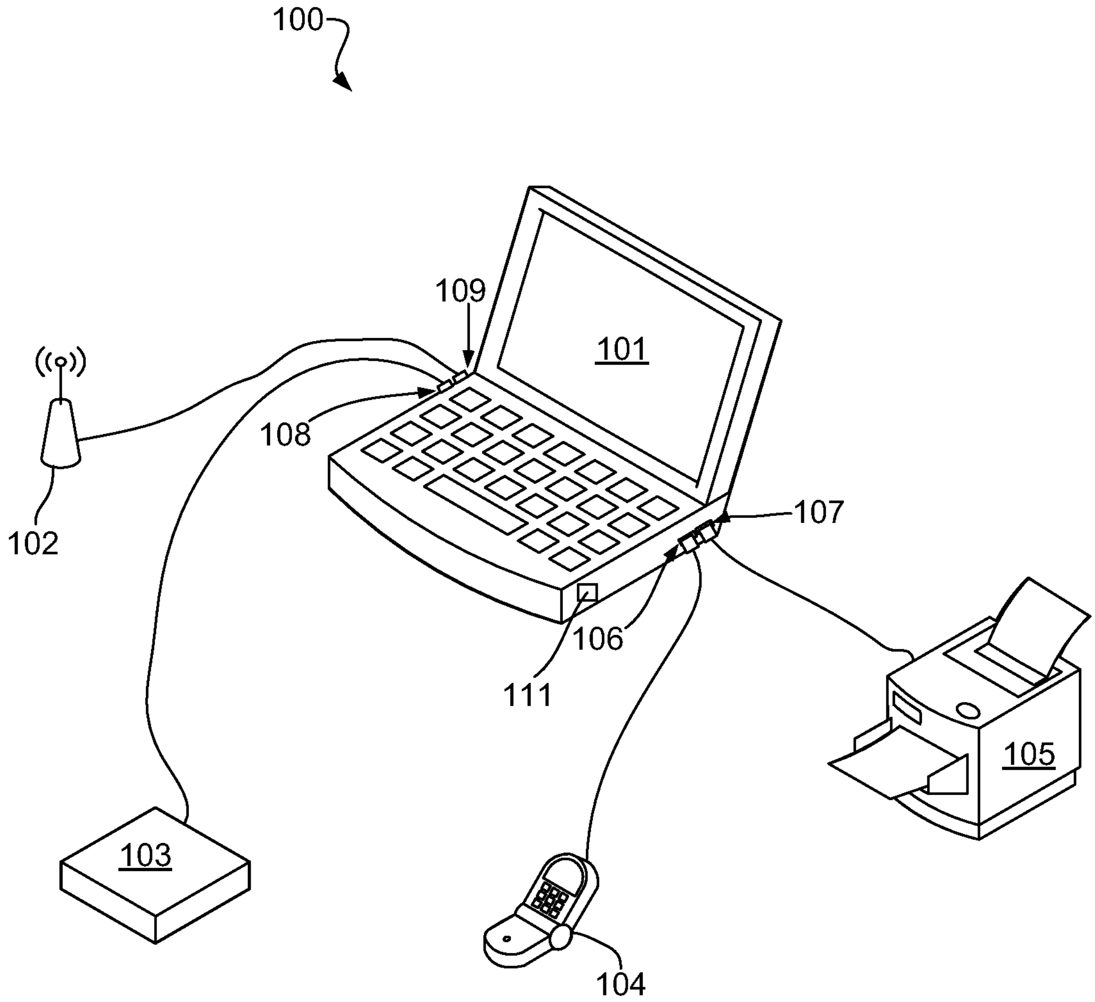

Fig. 1

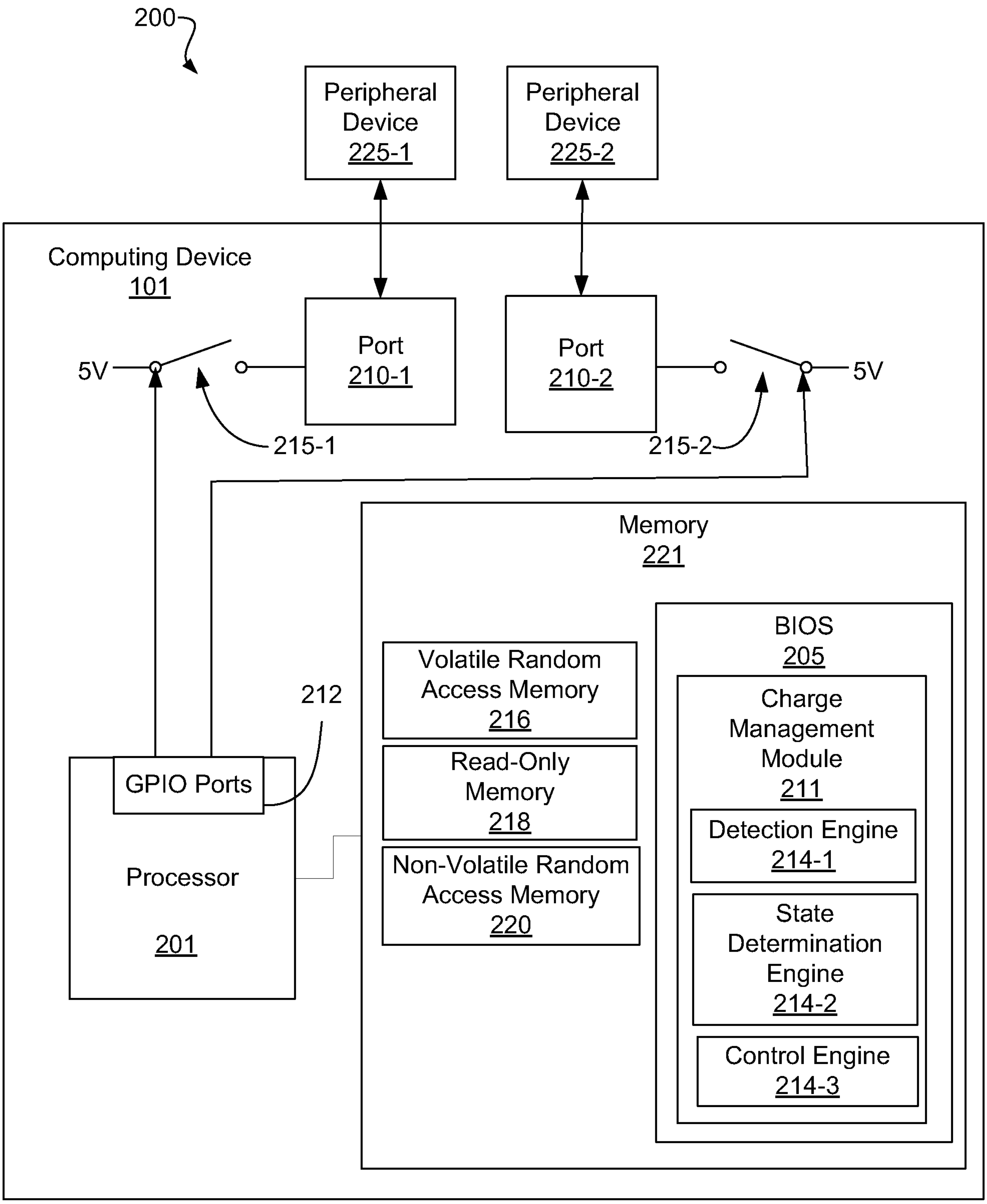
200
Peripheral Device 225-1
Peripheral Device 225-2
Computing Device 101
5V
Port 210-1
Port 210-2
5V
215-1
215-2
Memory 221
212
Volatile Random Access Memory 216
Read-Only Memory 218
Non-Volatile Random Access Memory 220
BIOS 205
Charge Management Module 211
Detection Engine 214-1
State Determination Engine 214-2
Control Engine 214-3
GPIO Ports
Processor 201

Fig. 2

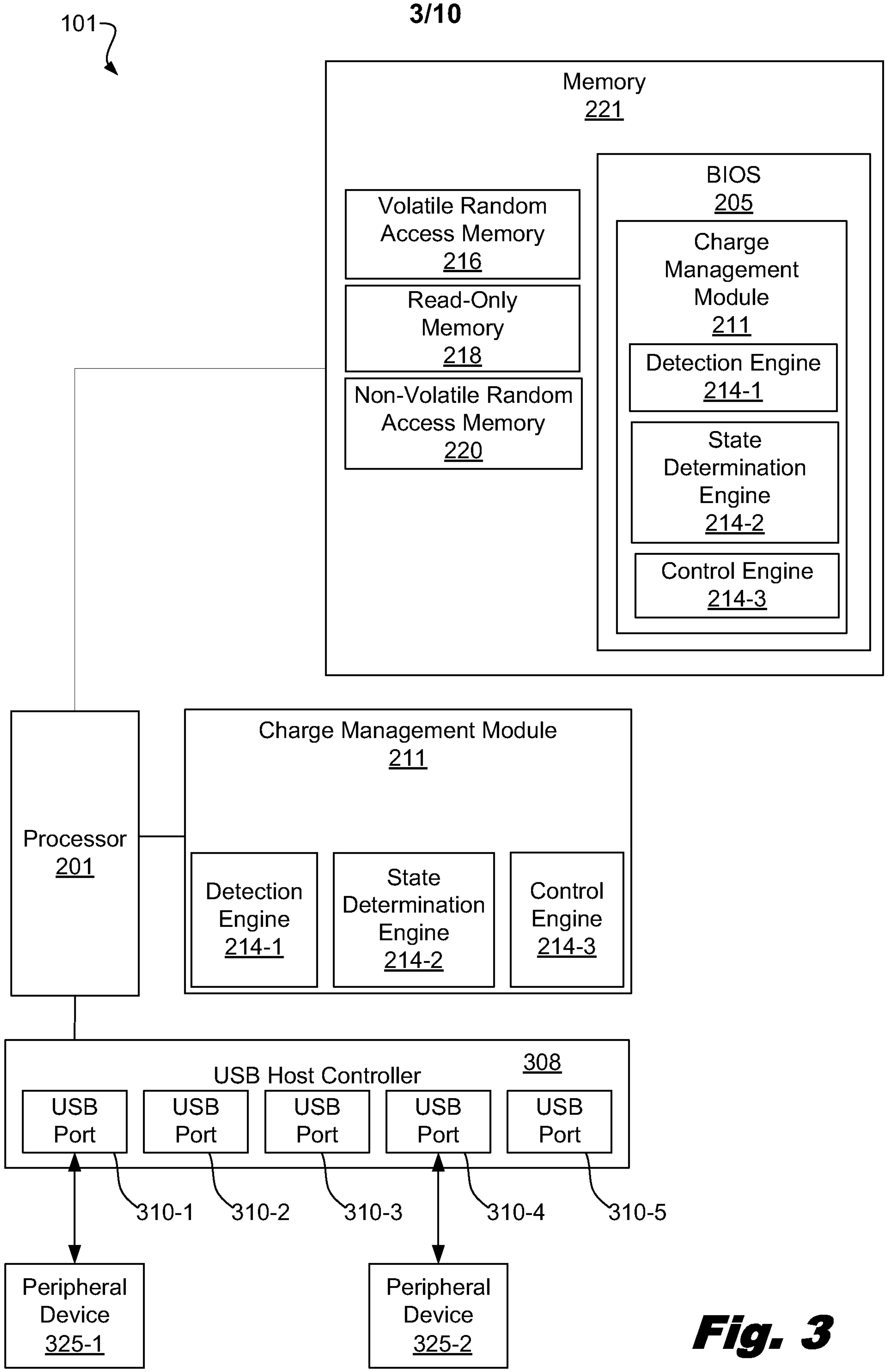
101
Memory
221
Volatile Random Access Memory
216
Read-Only Memory
218
Non-Volatile Random Access Memory
220
BIOS
205
Charge Management Module
211
Detection Engine
214-1
State Determination Engine
214-2
Control Engine
214-3
Processor
201
Charge Management Module
211
Detection Engine
214-1
State Determination Engine
214-2
Control Engine
214-3
USB Host Controller
308
USB Port
USB Port
USB Port
USB Port
USB Port
310-1
310-2
310-3
310-4
310-5
Peripheral Device
325-1
Peripheral Device
325-2

Fig. 3

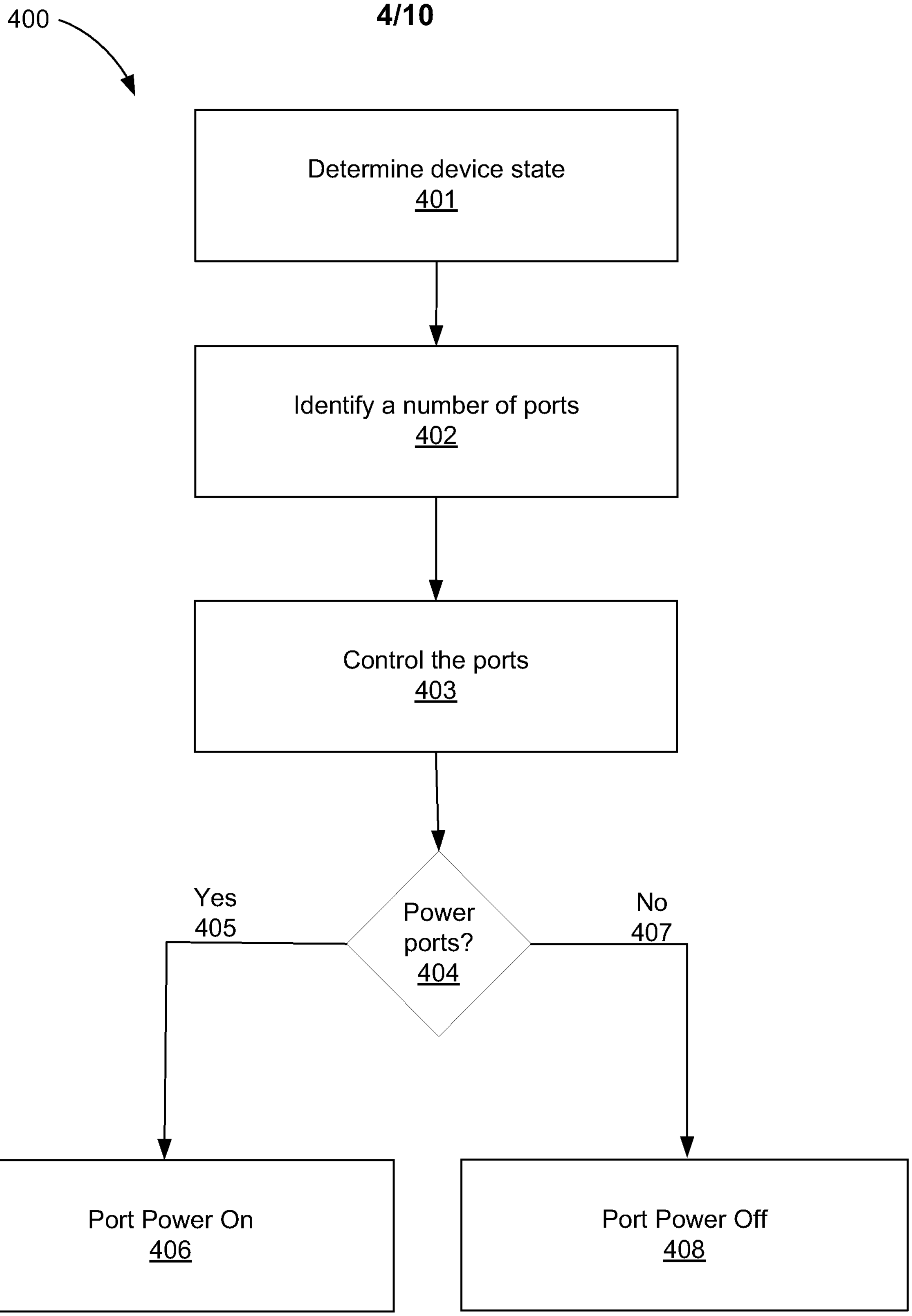
400
Determine device state
401
Identify a number of ports
402
Control the ports
403
Power ports?
404
Yes
405
No
407
Port Power On
406
Port Power Off
408

Fig. 4

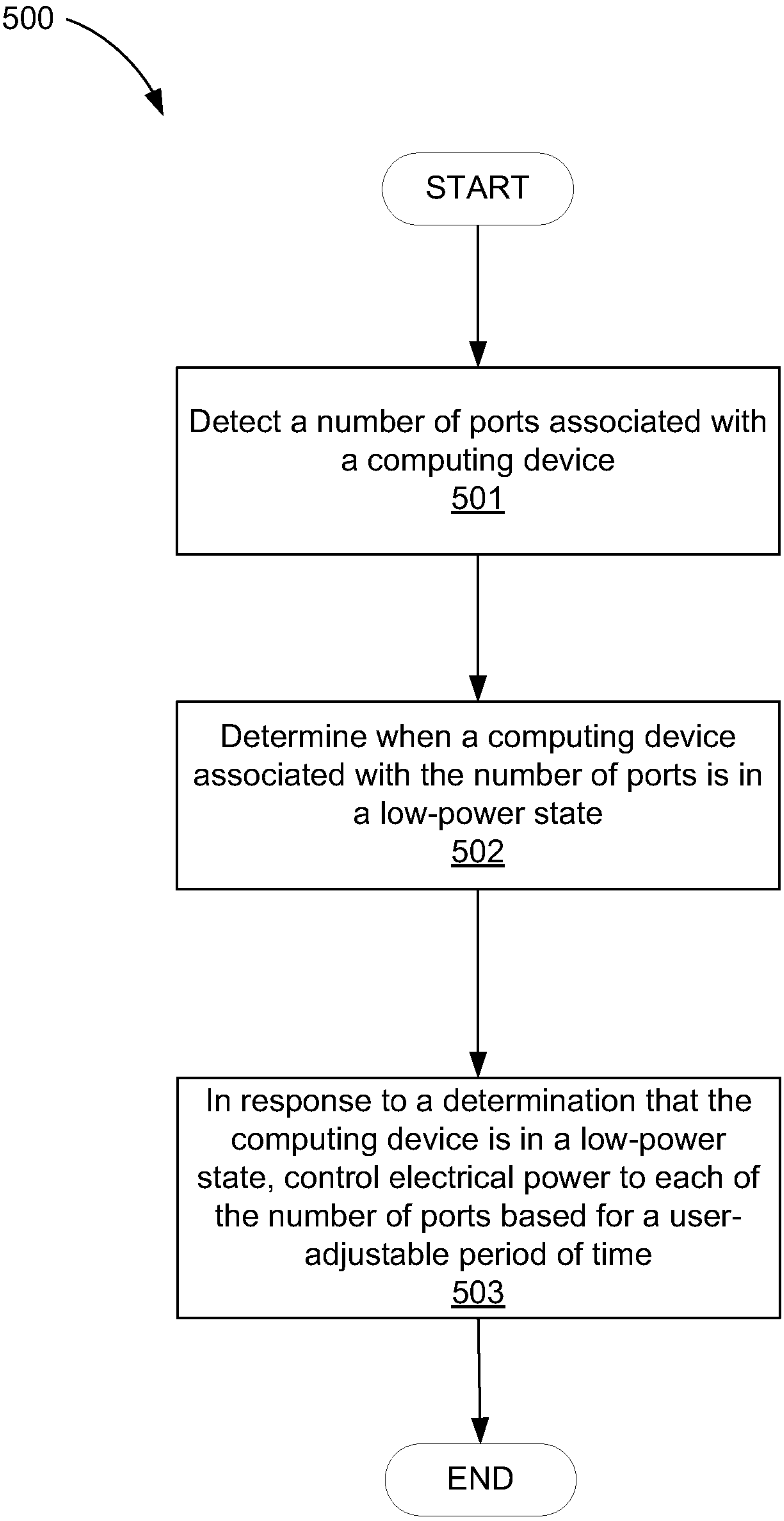
500
START
Detect a number of ports associated with a computing device
501
Determine when a computing device associated with the number of ports is in a low-power state
502
In response to a determination that the computing device is in a low-power state, control electrical power to each of the number of ports based for a user-adjustable period of time
503
END

Fig. 5

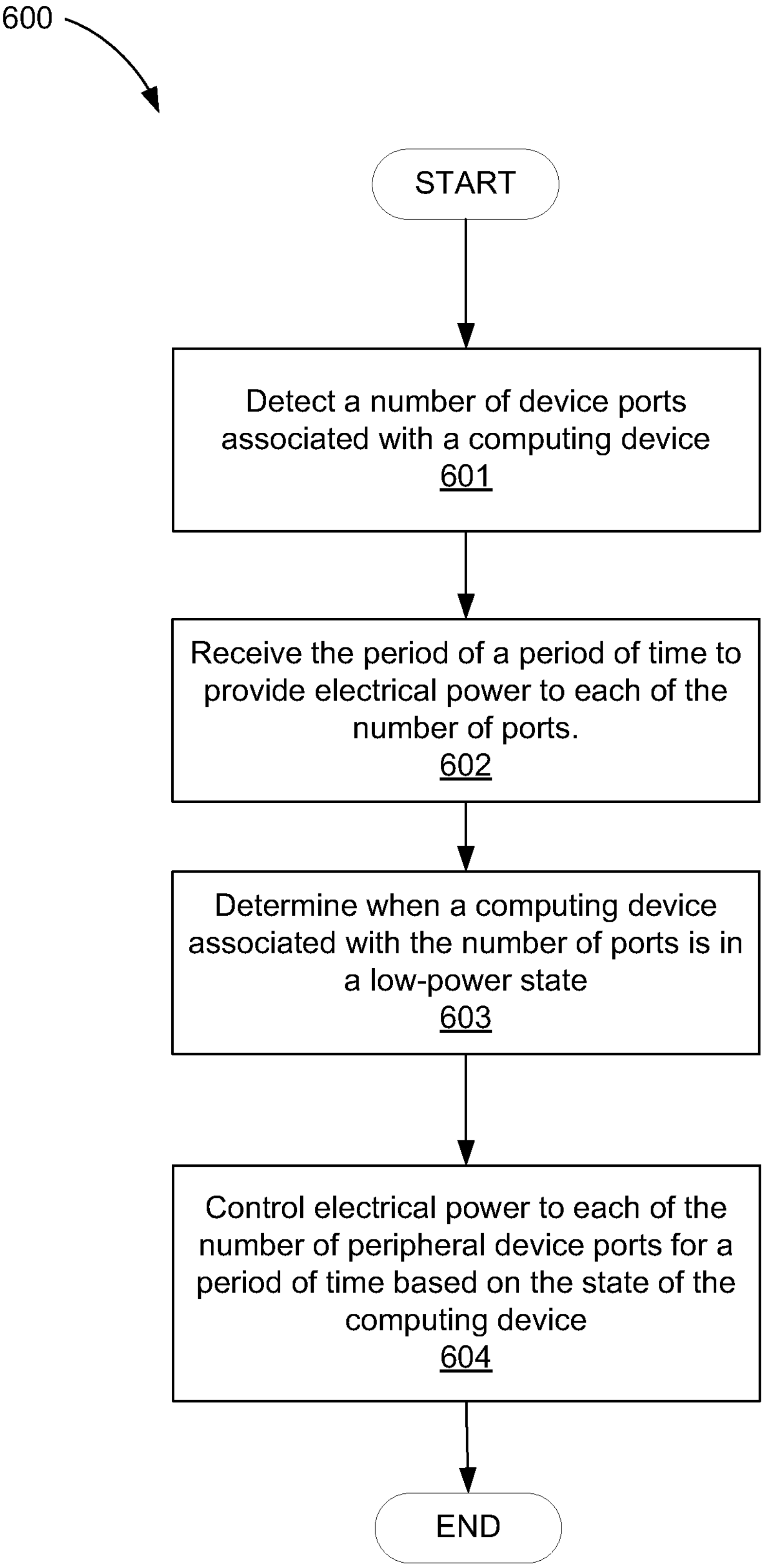
600
START
Detect a number of device ports associated with a computing device
601
Receive the period of a period of time to provide electrical power to each of the number of ports.
602
Determine when a computing device associated with the number of ports is in a low-power state
603
Control electrical power to each of the number of peripheral device ports for a period of time based on the state of the computing device
604
END

Fig. 6

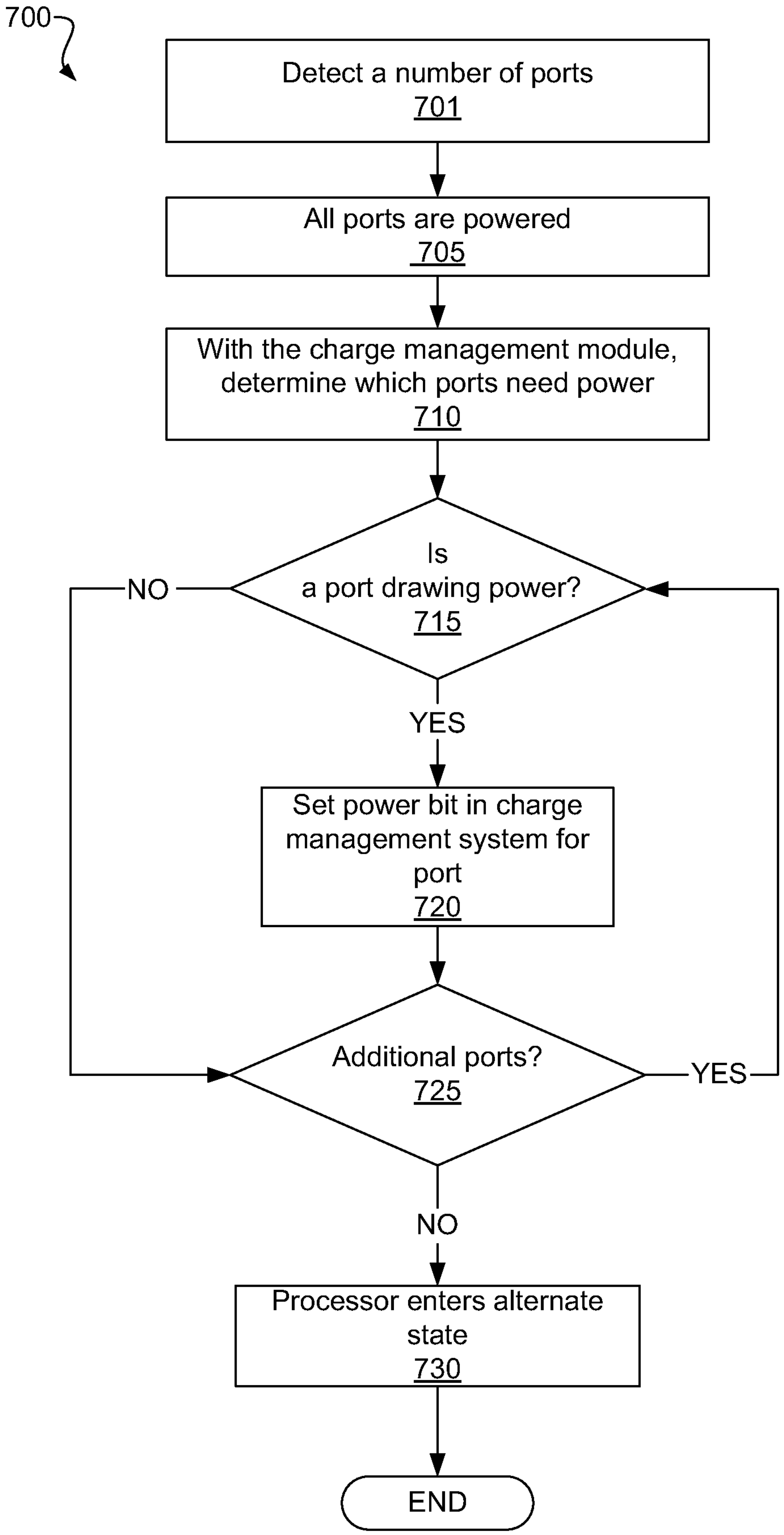
700
Detect a number of ports
701
All ports are powered
705
With the charge management module, determine which ports need power
710
Is a port drawing power?
715
NO
YES
Set power bit in charge management system for port
720
Additional ports?
725
YES
NO
Processor enters alternate state
730
END

Fig. 7

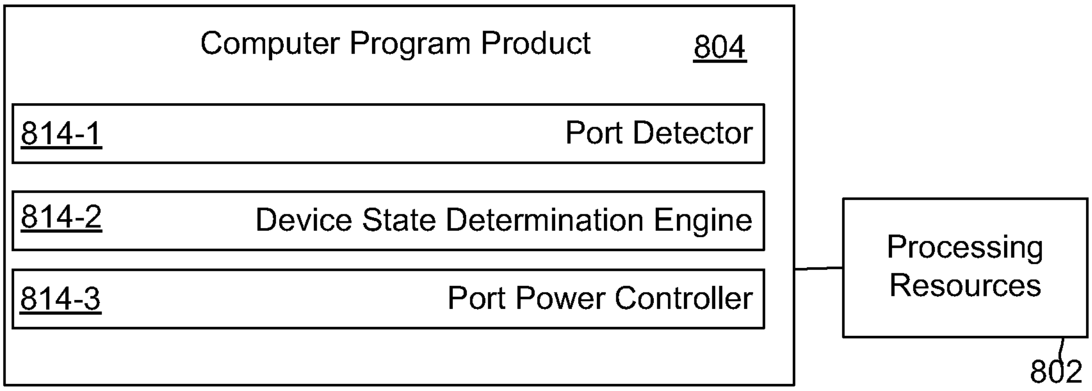
Computer Program Product
804
814-1
Port Detector
814-2
Device State Determination Engine
814-3
Port Power Controller
Processing Resources
802

Fig. 8

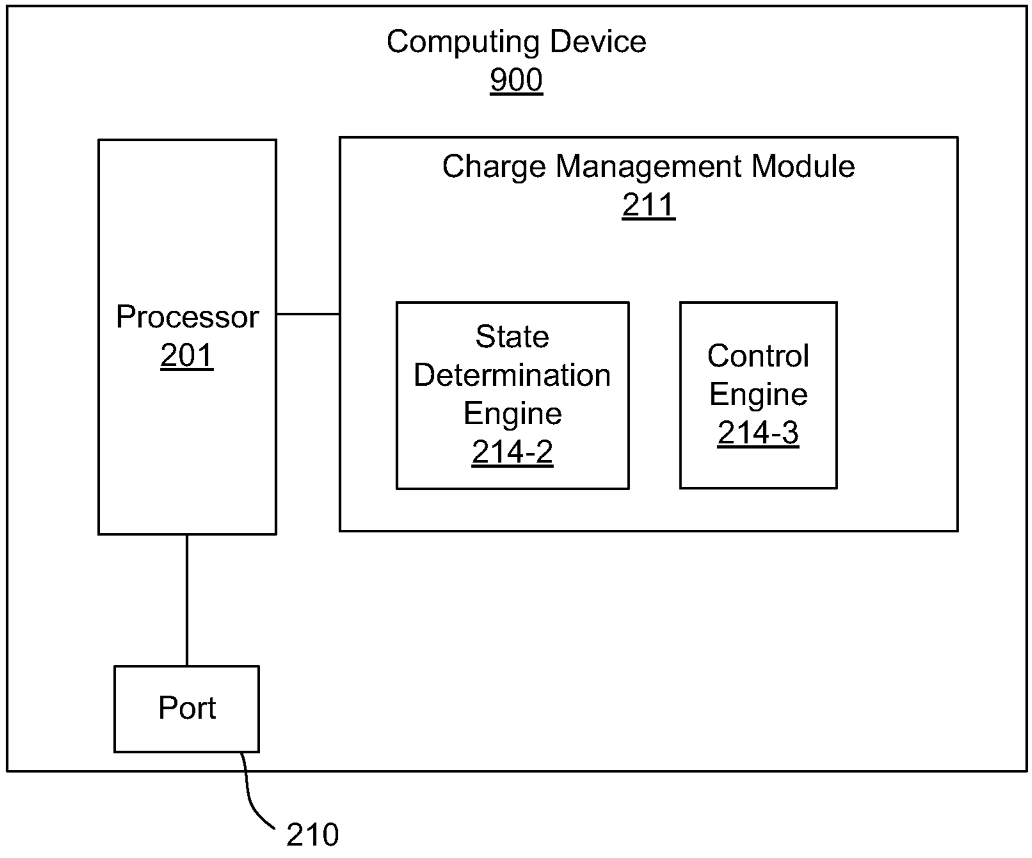
Computing Device
900
Charge Management Module
211
Processor
201
State Determination Engine
214-2
Control Engine
214-3
Port
210

Fig. 9

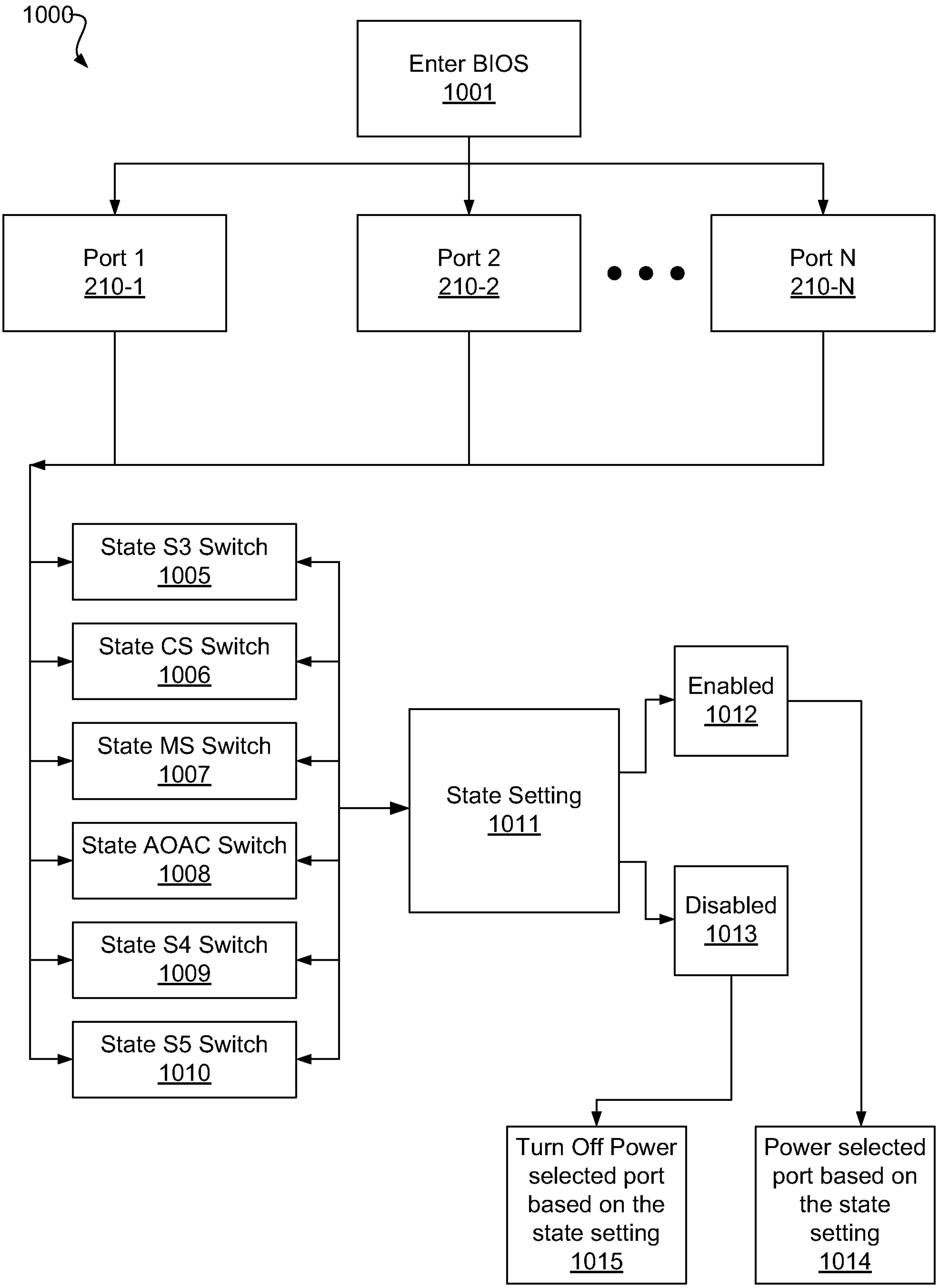
1000
Enter BIOS 1001
Port 1 210-1
Port 2 210-2
Port N 210-N
State S3 Switch 1005
State CS Switch 1006
State MS Switch 1007
State AOAC Switch 1008
State S4 Switch 1009
State S5 Switch 1010
State Setting 1011
Enabled 1012
Disabled 1013
Turn Off Power selected port based on the state setting 1015
Power selected port based on the state setting 1014

Fig. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/043217

A. CLASSIFICATION OF SUBJECT MATTER

G06F 1/32(2006.01)i, G06F 1/26(2006.01)i, G06F 13/14(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 1/32; G06F 3/00; H02J 7/00; G06F 13/14; G06F 1/26

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: computing, power, state, charge, port, time

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2013-0154551 A1 (JEFFREY KEVIN JEANSONNE et al.) 20 June 2013 See paragraphs [0017], [0022], [0028], [0030], [0042]; claims 1, 10-11, 15; and figures 3-4, 7.	1-15
Y	US 2015-0067373 A1 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 05 March 2015 See paragraphs [0019], [0027], [0037]; claim 12; and figures 2, 7.	1-15
A	US 2010-0219790 A1 (PETER CHADBOURNE et al.) 02 September 2010 See paragraphs [0017]-[0018]; and figure 1.	1-15
A	US 2012-0005502 A1 (GREG R. FIEBRICH et al.) 05 January 2012 See paragraphs [0016]-[0017]; and figure 2.	1-15
A	US 2009-0132730 A1 (KYUNG-BEOM KIM et al.) 21 May 2009 See paragraph [0044]; and figure 6.	1-15

□ Further documents are listed in the continuation of Box C. ☒ See patent family 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 application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of 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 25 April 2016 (25.04.2016)	Date of mailing of the international search report **26 April 2016 (26.04.2016)**
Name and mailing address of the ISA/KR International Application Division Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 35208, Republic of Korea Facsimile No. +82-42-481-8578	Authorized officer LEE, Dong Yun Telephone No. +82-42-481-8734

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/043217

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013-0154551 A1	20/06/2013	CN 103080868 A	01/05/2013
		DE 112010005782 B4	12/02/2015
		DE 112010005782 T5	02/05/2013
		GB 201302679 D0	03/04/2013
		GB 2496552 A	15/05/2013
		JP 2013-536965 A	26/09/2013
		JP 5634608 B2	03/12/2014
		KR 10-2013-0097198 A	02/09/2013
		WO 2012-030348 A1	08/03/2012
US 2015-0067373 A1	05/03/2015	US 2012-0017101 A1	19/01/2012
		US 8914649 B2	16/12/2014
		WO 2010-090646 A1	12/08/2010
US 2010-0219790 A1	02/09/2010	CN 102388520 A	21/03/2012
		CN 102388520 B	21/01/2015
		KR 10-2011-0133574 A	13/12/2011
		TW 201107984 A	01/03/2011
		US 8736227 B2	27/05/2014
		WO 2010-099483 A2	02/09/2010
		WO 2010-099483 A3	06/01/2011
US 2012-0005502 A1	05/01/2012	US 2009-0271644 A1	29/10/2009
		US 8037331 B2	11/10/2011
US 2009-0132730 A1	21/05/2009	JP 2009-537890 A	29/10/2009
		KR 10-0796302 B1	21/01/2008
		KR 10-2007-0111028 A	21/11/2007
		US 7945711 B2	17/05/2011
		WO 2007-133038 A1	22/11/2007