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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 November 2016 (03.11.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/175756 A1

(51) International Patent Classification:
*G06F 9/24* (2006.01) *G06F 9/22* (2006.01)

(21) International Application Number:
PCT/US2015/027901

(22) International Filing Date:
28 April 2015 (28.04.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventors: **SPOTTSWOOD, Jason**; 11445 Compaq Center Drive W., Houston, Texas 77070 (US). **BROWN, Andy**; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).

(74) Agents: **SHOOKMAN, Jeb Anthony** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

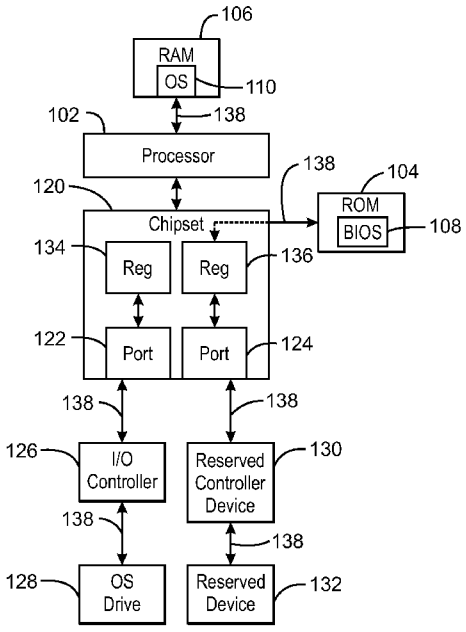
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, 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).

*[Continued on next page]*

(54) Title: HARDWARE FOR SYSTEM FIRMWARE USE

(57) Abstract: A method and a system for reserving a device for a system are provided herein. The method includes accessing a reserved device, where a basic input/output system (BIOS) uses the reserved device. The method includes initializing a register, via the BIOS firmware, to disable a port that connects to the reserved device. The method includes disabling the port that connects to the reserved device. The disabling may occur before the BIOS firmware transfers control of the system to an operating system. The disabling may hide the reserved device from the operating system and reserve the reserved device for the BIOS firmware without interference from the operating system.




100

FIG. 1B

WO 2016/175756 A1

**Declarations under Rule 4.17**:

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

— *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))*

# HARDWARE FOR SYSTEM FIRMWARE USE

## BACKGROUND

[0001] In a computing system, a Basic Input/Output System (BIOS) refers to firmware instructions located in the Read-Only Memory (ROM). The BIOS firmware performs several functions such as initializing and testing the hardware components, loading the operating system from a mass memory device, and interfacing the hardware components of the computing system with the operating system. When the computing system is initially powered ON, the BIOS firmware takes control of the system and runs a series of test referred to as the Power-On Self-Test (POST) to ensure that the components of the system are in proper working order. After the completion of the POST, the BIOS firmware searches for and initializes a boot process to load the operating system. Once a boot program is found, the BIOS firmware loads the operating system and passes control of the computing system to the operating system.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Certain examples are described in the following detailed description and in reference to the drawings, in which:

[0003] Fig. 1A is a block diagram of an example of a system;

[0004] Fig. 1B is a block diagram of a another view of the example of the system;

[0005] Fig. 2A is a process flow diagram of an example of a method for disabling a port in the system;

[0006] Fig. 2B is a process flow diagram of an example of a method for enabling and disabling a port in the system; and

[0007] Fig. 3 is a block diagram of an example of a tangible, non-transitory, computer-readable medium that stores code configured to disable a port using BIOS firmware.

## DETAILED DESCRIPTION

[0008] The Basic Input/Output System (BIOS) is a type of firmware used during initialization of a computing system, for example, the boot-up process. The BIOS

firmware may be in the form of a chip located on a motherboard of the computing system and may contain instructions and setup information related to initialization of the computing system, among other features. The BIOS firmware may use hardware devices of the computing system, including various types of controllers, ports, cards, and the like, to carry out various implementations. For example, the storage of data by the BIOS firmware into larger capacity storage devices has emerged as an alternative to limited storage capacity chips, such as ROM, PROM, EEPROM, and the like. However, after the BIOS firmware releases control of the computing system to an operating system (OS), the OS may detect the larger capacity storage device during a scan of the system. Once detected, the OS may take control of the storage device so as to render it unavailable for BIOS firmware use. Consequently, the BIOS firmware cannot use the larger capacity storage device. In some cases, data loss or corruption may occur if the BIOS firmware accesses the storage device while it is under the control of the OS.

[0009] To store the larger capacity data, the BIOS firmware may use hardware devices that include non-volatile storage technology such as Not AND (NAND) flash memory or a serial peripheral interface (SPI) flash memory. However, the use of NAND flash memory and SPI flash memory to store larger capacity data may involve performance and storage capacity issues. For example, with a data capacity of 4 gigabytes (GB), the storage capacity of the NAND flash memory may limit the amount of applications, data, and log information stored by the BIOS firmware. Moreover, due to its transfer bus speed, the SPI flash memory may slow data transfers and thus, limit performance capabilities of the BIOS firmware to store data.

[0010] As previously stated, the example of data storage to a larger capacity storage device by the BIOS firmware is merely one instance where the BIOS firmware may use the hardware of the computing system. In other examples, the BIOS firmware may use a network controller to carry out various features such as error reporting to a central server or remote management, and so forth. Further, the BIOS firmware may use other types of hardware devices including various types of controllers, ports, cards, and the like, to carry out various implementations.

[0011] Examples describe herein provide techniques for BIOS firmware to use hardware devices connected to a computing system. In particular, the BIOS

firmware may reserve hardware devices connected to the computing system for use without interference from an operating system. In some examples, the BIOS firmware may use the hardware device for its own use during it boot-up process. After boot-up, but before the BIOS transfers control of the computing system to the operating system, the BIOS firmware may initialize a register in a chipset to disable a port that connects to the hardware device. The disabling of the port may hide the hardware device attached to the port from the OS and as a result, reserve the device for the BIOS firmware without interference from the operating system.

[0012] Fig. 1A is a block diagram of an example of a system, for example, a computing system 100. The computing system 100 may include a laptop computer, desktop computer, tablet computer, server, or the like. The computing system 100 may include internal and external computing components where data can be transmitted among the computing components via a system of wires, e.g., a bus, as discussed herein.

[0013] In the present examples, the computing system 100 may include a processor 102 that is adapted to execute stored instructions, as well as memory devices that store data executable by the processor 102. The processor 102 can be a single core processor, a multi-core processor, a computing cluster, or any number of other configurations. A memory device, such as a random access memory (RAM) device 106, may be connected to the processor 102. The RAM device 106 may be used to store volatile memory contents that would otherwise be lost when the computing system 100 is powered off. In some examples, the RAM device 106 may include dynamic random access memory (DRAM) device, a static random access memory (SRAM) device, or any other suitable memory systems.

[0014] The computing system 100 may also include a ROM device 104, for example, an electronically erasable programmable read only memory (EEPROM) device or any other suitable memory systems. The ROM device 104 may store basic input/output system (BIOS) instructions that are often referred to as the BIOS firmware 108, as shown in Fig. 1A. The BIOS firmware 108 is software used to startup (boot) a computing system, and which may be used to interface the various computing components with an operating system (OS) 110. Specifically, the BIOS firmware 108 performs several functions including enabling a booting process to load

the OS 110 into the RAM device 106 from disk storage. The OS 110, of the present examples, may include a software program used to manage hardware and software resources of the computing system 100.

[0015] The processor 102 may connect to a display interface 112 configured to connect the computing system 100 to one or more display devices 114. The display devices 114 may include a computer monitor, television, or projector, among other peripheral devices, that are externally connected or built into the computing system 100.

[0016] The processor 102 may also connect to an input/output (I/O) device interface 116 configured to connect the computing system 100 to one or more I/O peripheral devices 118. The I/O peripheral devices 118 may include, for example, a keyboard, a mouse, a printer, or a pointing device, among other peripheral devices, that are externally connected or built into the computing system 100.

[0017] A chipset 120 is electrically connected to the processor 102. The chipset 120 is a group of integrated circuits that manage the data flow between the processor 102 and other internal or external hardware devices. The chipset 120 may include ports to provide an interface between the processor 102 and other devices. As shown in Fig.1, the chipset 102 includes two ports, 122 and 124, however, additional ports may be present. Depending on the specifications of the manufacturer, an internal hardware device, such as an I/O controller 126 may connect to the ports. For example, the I/O controller 126 may connect to the port 122. The I/O controller 126 may further connect to an OS drive 128 used by the OS 110, for example, as a storage device.

[0018] A hardware device, such as a reserved controller device 130, may connect to the port 124 and may further connect to a reserved device 132. In this manner, the reserved controller device 130 may act as an interface so as to provide an intermediary link between the processor 102 and the reserved device 132.

[0019] In certain examples, the reserved controller device 130 may be removed and the reserved device 132 may be directly attached to the port 124. For example, the reserved device 132 may include a network interface controller (NIC) directly attached to the port 124 to connect the computing system 100 to an external computing network.

[0020] In the present examples, the BIOS firmware 108 may perform a sequence of events after the computing system 100 is powered on. For example, the BIOS firmware 108 may perform diagnostic tests on computing components or load instructions related to the OS 110 from a disk to the RAM device 106, among other events. After the OS 110 is loaded into the RAM 106, the BIOS firmware 108 turns control of the computing system 100 over to the OS 110, which is ready for user interaction.

[0021] However, before the OS 110 is loaded into the RAM device 106, the BIOS firmware 108 may use the reserved device 132 to carry out various functions, such as for data storage or as a controller to connect the computing system 100 to a computer network. Thereafter, the BIOS firmware 108 may transfer control of the computing system 100 to the OS 110. In order to prevent the OS 110 from using the reserved device 132 after the OS 110 takes control, the BIOS firmware 108 may program the chipset 120 with configuration data to disable the port 124 that is connected to the reserved controller device 130, as will be further discussed with respect to Fig. 1B. The disabling of the port 124 may act to hide the reserved controller device 130 once the OS 110 takes control and initiates a configuration scan of the system 100. Further, the disabling of the port 124 may act to hide any devices located behind the device 130, such as, the reserved device 132. Once hidden, the reserved controller device 130 and the reserved device 132 may not be subjected to interference or use from the OS 110 or any other hardware or software of the computing system 100. In some examples, as will be further explained, the BIOS firmware 108 may use the reserved device 132 after the OS 110 takes control. In this case, the BIOS firmware 108 may initialize the register 136 with configuration data to enable the port 124 connected to the reserved device 132. The BIOS firmware 108 may then access the reserved device 132 for its own use and subsequently, disable the port 124 to hide the device 132 from the OS 110.

[0022] Fig. 1B is a block diagram of a detailed view of the example of the system. Like numbered items are as described with respect to Fig. 1A. As previously discussed, the disabling of the port 124 may hide the reserved controller device 130 including the reserved device 132 connected thereto. To hide the reserved device 132 from the OS 110, the BIOS 108 may initialize a register located in the the

chipset 120 to disable the port 124 connected to the reserved controller device 130. As shown in Fig. 1B, the chipset 120 includes several registers, including registers 134 and 136. The registers 134 and 136 are data storage areas that may be used to control the behavior of the chipset 120 and its component parts, such as the ports 122 and 124. In the present examples, the register 136 may be a function disable register. Thus, the register 136 may be initialized to disable the port 124 to hide the reserved device 132 from the operating system 110.

[0023] A system bus 138, as used in the present examples, is a set of wires that electrically connect and move data between the internal and external devices of the computing system 100, as shown in Fig. 1B. The BIOS firmware 108 may electrically connect to the chipset 120 through the bus system 138 in order to write values into the register 136. Further, the system bus 138 may provide a common electrically pathway between the reserved controller device 130, the reserved device 132, and the port 124, among other devices of the computing system 100. A bus protocol is a set of rules that governs the operations of the system bus 138. In the present examples, the system bus 138 may use a peripheral component interconnect (PCI) bus protocol or a PCI Express (PCIe) bus protocol. Hence, the reserved controller device 130 may be a PCIe reserved controller device and the reserved device 132 may be a PCIe device. In the case of a storage implementation, the reserved controller device 130 may be a PCIe storage controller and the reserved device 132 may be a PCIe non-volatile solid-state storage device. In the present examples, the PCIe bus may be the primary bus standard as it may improve the efficiency of the reserved controller device 130 to transmit and receive data to and from the other hardware devices, for example, the ROM device 104.

[0024] As previously described, the disabling of the port 124 may occur before the BIOS firmware 108 transfers control of the computing system 100 to the OS 110. Once the OS 110 takes control, it may initiate a scan to determine the configuration of the computing system 100, for example, detecting coupled devices. However, the OS 110 may not detect the reserved controller device 130 and thus, the reserved device 132 connected thereto. Consequently, if the OS 110 desires to use a device, for example, for storage, it may select only the devices that are visible to it during the configuration scan, for example, the OS drive 128.

[0025] Other examples illustrate use of hardware devices by the BIOS firmware 108. In some examples, the BIOS firmware 108 may use hardware devices after control has passed to the OS 110. For example, power-loss to the computing system 100 may occur after the OS 110 has gained control. A power source may be switched to a battery-based power system and a back-up of the RAM device 106 may be initiated. The back-up may require the use of the reserved device 132, for example, a large capacity storage device. In the case where the port 124 was disabled before the OS 110 takes control, the BIOS firmware 108 may enable the port 124 connected to the reserved device 132. In particular, the BIOS firmware 108 may initialize the register 136 with values to enable the port 124 connected to the reserved device 132. Once enabled, the BIOS firmware 108 may initiate a full backup of the data to the reserved device 132. The port 124 may be disabled again after the BIOS firmware 108 completes the backup but before the OS 110 regains control. Consequently, the reserved device 132 may be hidden from the OS 110 after it re-initializes and retakes control.

[0026] Fig. 2A is a process flow diagram of an example of a method for disabling a port in the system. The BIOS firmware controls the system from the initial start-up until the OS takes control of the system. In particular, one of the primary functions of the BIOS firmware includes loading the OS into the random access memory (RAM) of the system from disk storage. However, in some cases, the BIOS firmware may use hardware of the system to carry out various functions. In order to reserve the hardware for the BIOS firmware and to prevent the OS from using the hardware, a port connected to the hardware may be disabled to prevent the OS from detecting the hardware.

[0027] At block 202, the BIOS firmware may access the reserved device for use. In some examples, the BIOS firmware may use the reserved device for the storage of data or as a network controller to provide error reporting, among other implementations. At block 204, the BIOS firmware may initialize a register to disable a port connected to the reserved device. To initialize the register, the BIOS firmware may write values to the register. In the present examples, the values may instruct the register to disable the port connected to the reserved device. At block 206, the method includes disabling the port connected to the reserved device, where the

disabling occurs before the BIOS firmware transfers control of the system to the OS. In some examples, the initializing may take place after the BIOS firmware passes control of the system to the OS. The disabling of the port hides the reserved device from the OS during a configuration scan of the system and thus, prevents the OS from using the reserved device. In this manner, the disabling of the port reserves the reserved device for the BIOS firmware use without interference from the OS.

[0028] Fig. 2B is a process flow diagram of an example of a method for enabling and disabling a port in the system. In some examples, the need may arise for the BIOS firmware to be enabled after the OS has taken control of the system. For example, a disruption to the system may include a loss of power, among other disruptions that interrupt the operations of the system. Since the port connected to the reserved device is disabled prior to initiating the OS, as previously discussed, the port may need to be re-enabled for BIOS firmware use. At block 208, the method may include enabling the disabled port connected to the reserved device after the OS has taken control. The BIOS firmware may enable the port by writing values to the register to enable the port connected to the reserved device. At block 210, the method may include the BIOS firmware accessing the reserved device for use after enabling. At block 212, the method may include disabling the port connected to the reserved device after the BIOS firmware use but before the OS re-takes control of the system. The disabling of the port may hide the reserved device from the OS after it takes control.

[0029] Fig. 3 is a block diagram of an example of a tangible, non-transitory, computer-readable medium 300 that stores code configured to disable a port using BIOS firmware. The computer-readable medium 300 may store code that can be accessed by a processor 302 over a computer bus 304 to perform the methods previously described. The computer-readable medium 300 can include a hard disk drive, a solid-state non-volatile storage device, a network controller, a graphics processor unit, among others.

[0030] The computer-readable medium 300 may include modules 306, 308, and 310. For example, an access module 306 may be configured to allow BIOS (basic input/output system) firmware to access a reserved device for use. An initialization module 308 may be configured to initialize a register in a chipset, via the BIOS

firmware, to disable a port connected to the reserved device. The initialization module 306 may perform its operations before the BIOS firmware passes control of a system to an operating system (OS). A disable module 310 may be configured to disable the port connected to the reserved device, wherein the disabled port acts to hide the reserved device from the OS. The disabling of the port may occur before control of the system is passed to the OS so that the OS may not use the reserved device.

[0031] The BIOS firmware may use the reserved device for the storage of larger capacity data, to carry our various features for a network controller, among other implementations. In particular, the BIOS firmware may initialize a register with data values to disable a port connected to the reserved device. The disabling of the port may hide the reserved device from the OS. In this manner, the reserved device can be reserved for use by the BIOS firmware without interference from the OS.

[0032] While the present techniques may be susceptible to various modifications and alternative forms, the embodiments discussed above have been shown only by way of example. However, it should again be understood that the techniques is not intended to be limited to the particular embodiments disclosed herein. Indeed, the present techniques include all alternatives, modifications, and equivalents falling within the true spirit and scope of the appended claims.

CLAIMS

1. A method of using firmware in a system, comprising:
accessing a reserved device, wherein a basic input/output system (BIOS) firmware uses the reserved device;
initializing a register, via the BIOS firmware, to disable a port connected to the reserved device; and
disabling the port connected to the reserved device, wherein the disabling occurs before the BIOS firmware transfers control of the system to an operating system, wherein the disabling hides the reserved device from the operating system, and wherein the hiding reserves the reserved device for the BIOS firmware without interference from the operating system.

2. The method of claim 1, wherein the initializing comprises writing values to the register that instructs the register to disable the port connected to the reserved device.

3. The method of claim 1, wherein the disabling hides a reserved controller device, wherein the reserved device is located behind the reserved controller device, and wherein the hiding of the reserved controller device hides the reserved device.

4. The method of claim 1, wherein the initializing takes place before or after the BIOS firmware uses the reserved device.

5. The method of claim 1, comprising:
enabling the disabled port connected to the reserved device after the operating system takes control; and
accessing the reserved device for the BIOS firmware after the enabling.

6. The method of claim 5, comprising:

disabling the port connected to the reserved device after the BIOS firmware uses the reserved device, wherein the disabling hides the reserved device from the operating system after the operating system retakes control.

7. A system comprising:

a processor;

a chipset connected to the processor, wherein the chipset comprises a plurality of ports and a plurality of registers;

a non-volatile memory connected to the chipset,

a basic input/output system (BIOS) firmware, wherein the BIOS firmware is stored in the non-volatile memory;

a reserved device connected to at least one port of the chipset, wherein the BIOS firmware initializes at least one register in the chipset to disable the at least one port connected to the reserved device, wherein the reserved device is reserved for BIOS firmware use; and

an operating system connected to the processor, wherein the reserved device is hidden from the operating system.

8. The system of claim 7, wherein the BIOS firmware writes values to the at least one register, and wherein the values instruct the at least one register to disable the at least one port connected to the reserved device.

9. The system of claim 7, wherein the reserved device is located behind a reserved controller device and wherein the reserved controller device is to hide the reserved device from the operating system after the at least one port connected to the reserved device is disabled.

10. The system of claim 7, wherein the at least one port connected to the reserved device is disabled prior to transferring control of the system to the operating system.

11. The system of claim 7, wherein during a disruption to the system, the at least one port connected to the reserved device is enabled for the BIOS firmware and disabled prior to transferring control back to the operating system.

12. The system of claim 7, wherein the reserved device is a non-volatile storage device, a controller, or port.

13. A tangible, non-transitory, computer-readable medium comprising a processor to carry out instructions of using firmware in a system, the instructions comprising:

access a reserved device, wherein a basic input/output system (BIOS) firmware uses the reserved device;

initialize a register, via the BIOS firmware, to disable a port connected to the reserved device; and

disable the port connected to the reserved device, wherein the disabling occurs before the BIOS firmware transfers control of the system to an operating system, wherein the disabling hides the reserved device from the operating system, and wherein the hiding reserves the reserved device for the BIOS firmware without interference from the operating system.

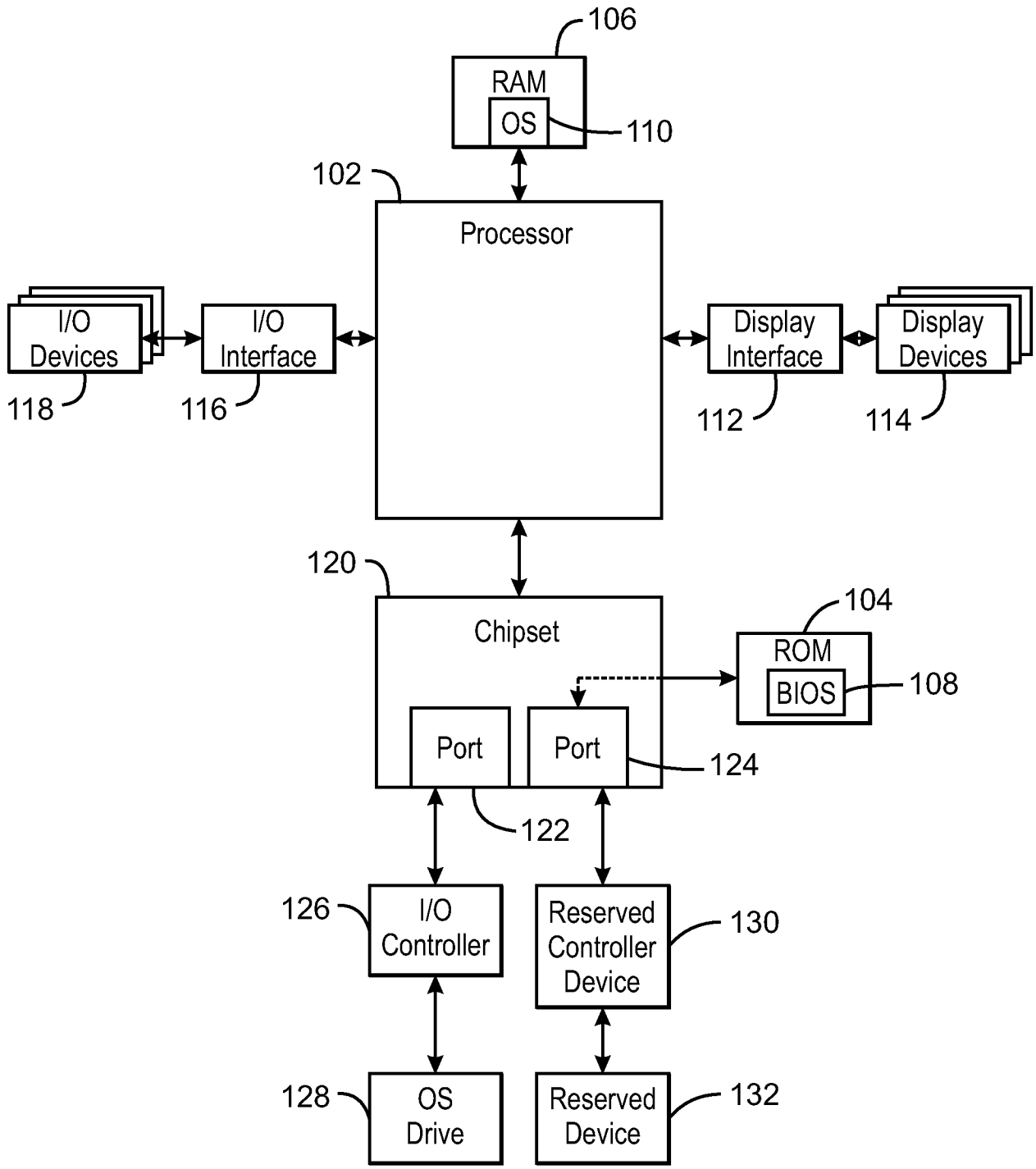
14. The tangible, non-transitory, computer-readable medium of claim 13, comprising:

enable the disabled port connected to the reserved device after the operating system takes control; and

access the reserved device for the BIOS firmware after the enabling.

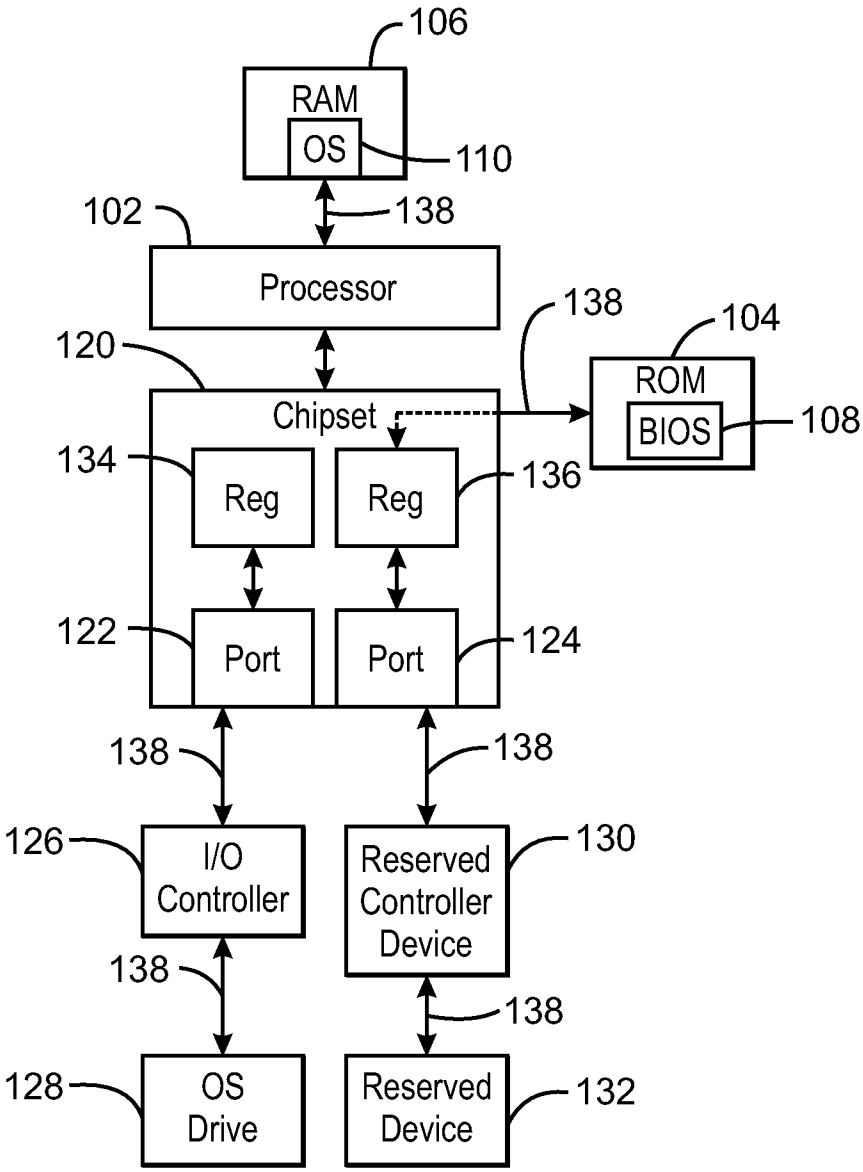
15. The tangible, non-transitory, computer-readable medium of claim 13, comprising:

disable the port that connects to the reserved device after the BIOS firmware uses the reserved device, wherein the disabling hides the reserved device from the operating system after the operating system retakes control.


106
RAM
OS
110
102
Processor
I/O Devices
118
I/O Interface
116
Display Interface
112
Display Devices
114
120
Chipset
104
ROM
BIOS
108
Port
Port
124
122
126
I/O Controller
Reserved Controller Device
130
128
OS Drive
Reserved Device
132

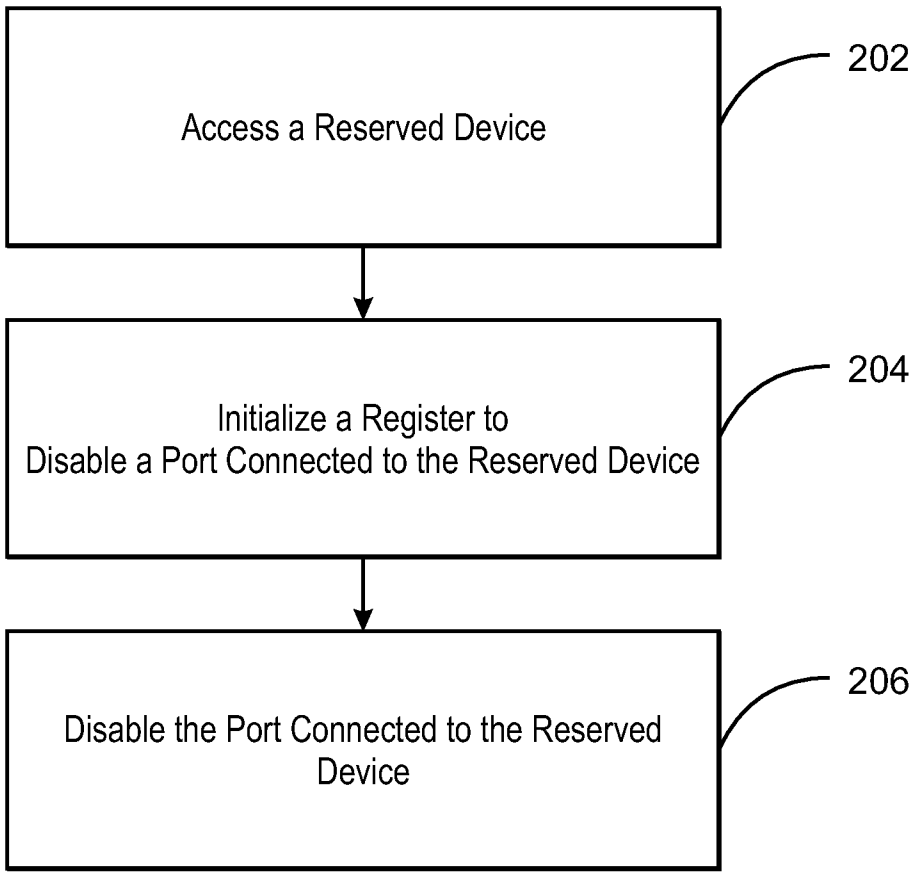
100

FIG. 1A


106
RAM
OS
110
102
138
Processor
138
104
120
ROM
BIOS
108
Chipset
134
Reg
Reg
136
122
Port
Port
124
138
138
126
I/O Controller
Reserved Controller Device
130
138
138
128
OS Drive
Reserved Device
132

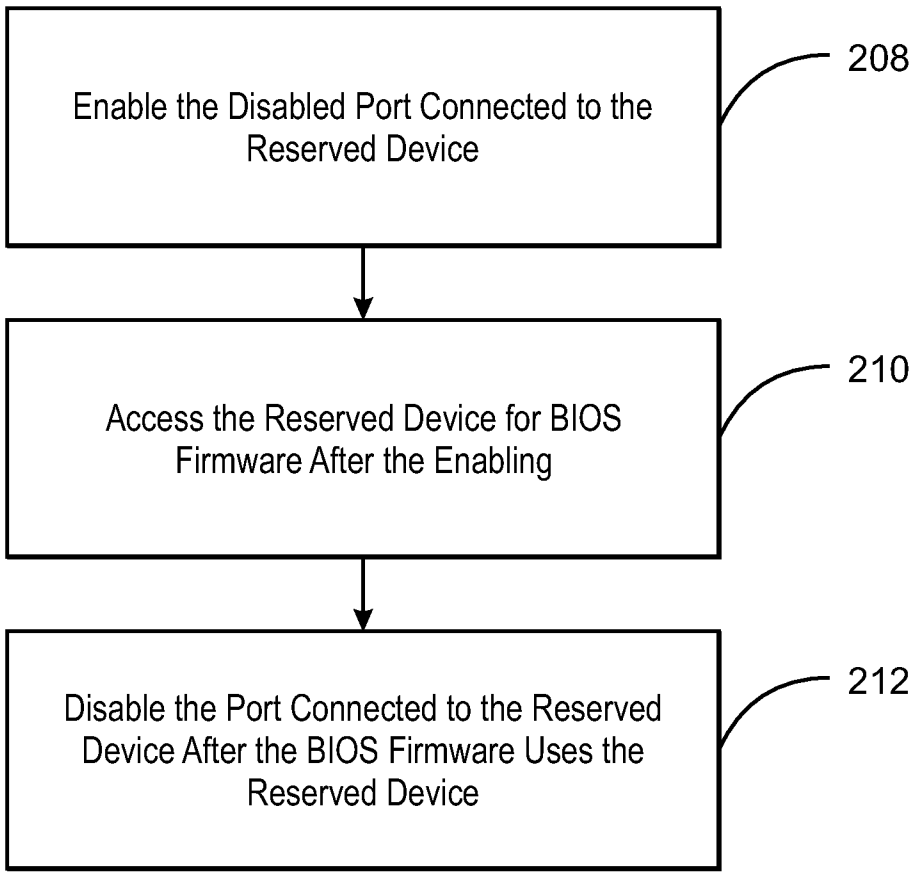
100

FIG. 1B


Access a Reserved Device
202
Initialize a Register to
Disable a Port Connected to the Reserved Device
204
Disable the Port Connected to the Reserved
Device
206

200A

FIG. 2A


Enable the Disabled Port Connected to the Reserved Device
208
Access the Reserved Device for BIOS Firmware After the Enabling
210
Disable the Port Connected to the Reserved Device After the BIOS Firmware Uses the Reserved Device
212

200B

FIG. 2B

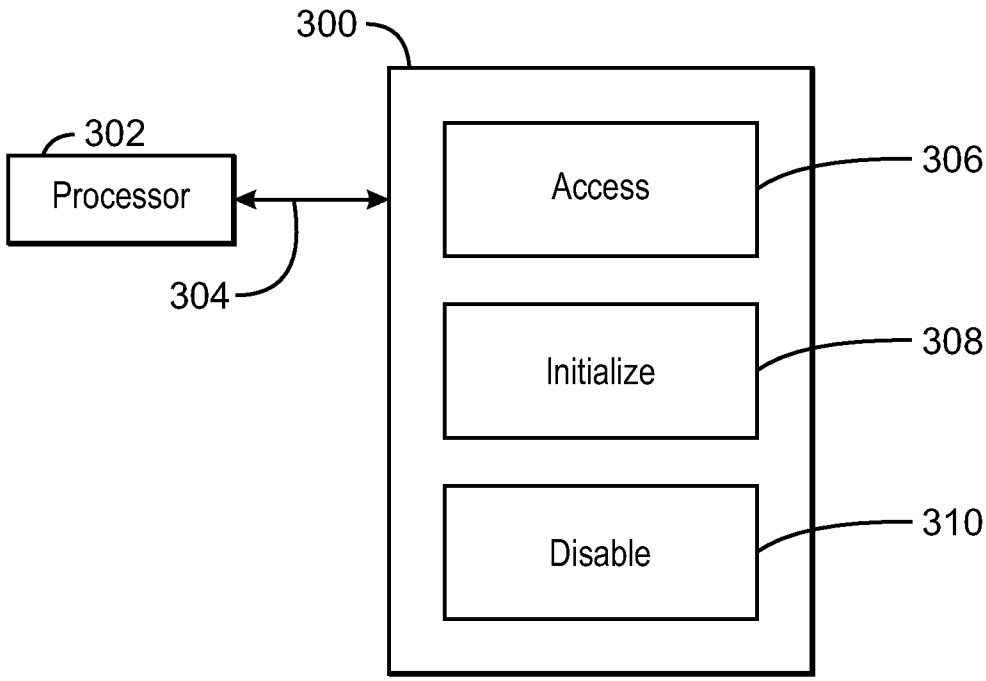

300
302
Processor
304
Access
306
Initialize
308
Disable
310

FIG. 3

**INTERNATIONAL SEARCH REPORT**

International application No.
**PCT/US2015/027901**

**A. CLASSIFICATION OF SUBJECT MATTER**

**G06F 9/24(2006.01)i, G06F 9/22(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 9/24; G06F 11/14; G06F 1/26; G06F 9/00; G06F 9/22

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: firmware, BIOS

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | US 2014-0173267 A1 (HON HAI PRECISION INDUSTRY CO., LTD.) 19 June 2014<br>See abstract, paragraphs [0013]-[0026], and figures 2-3. | 1-15 |
| A | US 2012-0017101 A1 (SO CHI W. et al.) 19 January 2012<br>See abstract, paragraphs [0019]-[0027], and figures 2-5. | 1-15 |
| A | US 2004-0030877 A1 (ALEKSANDR FRID) 12 February 2004<br>See abstract, paragraphs [0005]-[0008], and figures 1-3. | 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 | Date of mailing of the international search report |
|---|---|
| 29 October 2015 (29.10.2015) | **30 October 2015 (30.10.2015)** |

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
JI, Jeong Hoon
Telephone No. +82-42-481-5688

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

**PCT/US2015/027901**

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| None | | | |