

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

(19) World Intellectual Property

Organization

International Bureau

(43) International Publication Date

07 April 2022 (07.04.2022)



(10) International Publication Number

WO 2022/071925 A1

(51) International Patent Classification:

G06F 1/3246 (2019.01)

G06F 1/3287 (2019.01)

Published:

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

(21) International Application Number:

PCT/US2020/053311

(22) International Filing Date:

30 September 2020 (30.09.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: HEWLETT-PACKARD DEVELOPMENT

COMPANY, L.P. [US/US]; 10300 Energy Drive, Spring,
Texas 77389 (US).

(72) Inventors: TRUONG, Binh T.; 10300 Energy Dr., Spring,

Texas 77389 (US). CHANG, Chun; No. 66 Jing Mao 2
Road, 10F-2, NanGang District, Taipei City, 11568 (TW).

MCCLELLAN, James; 10300 Energy Drive, Spring,
Texas 77389 (US).

(74) Agent: GARDINER, Austin; 3390 E. Harmony Rd., 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, DJ, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN,
KP, KR, KW, 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, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

(54) Title: WAKE SETTINGS

(57) Abstract: An example system includes a processor resource and a memory resource with instructions stored thereon, where execution of the instructions causes the system to establish a first wake setting for a first physical communication interface independent of a second wake setting for a second physical communication interface. An example method of managing wake operations may include identifying a device identifier for a communication interface, mapping a wake setting to the device identifier, and setting a state of the communication interface corresponding to the wake setting.

WO 2022/071925 A1

WAKE SETTINGS

BACKGROUND

[0001] Electronic transactions are processed with computing machines. In some examples, a transaction system includes a host device to process the transaction and a peripheral device to interact with an employee or customer. A host device with limited ports may include a hub to allow for additional peripheral devices to communicate with the host device. Transaction systems may be customized for a particular type of transaction or retail experience, in particular, by connecting a variety of peripherals to the host device (e.g., via a hub).

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Figures 1 and 2 are block diagrams depicting example compute systems.

[0003] Figure 3 depicts an example compute system in which various example communication interfaces are coupled to the example compute system.

[0004] Figure 4 an example table for use with an example compute system.

[0005] Figures 5-9 are flow diagrams depicting example methods of managing wake operations.

DETAILED DESCRIPTION

[0006] In the following description and figures, some example implementations of compute systems, peripheral devices, and/or methods of managing wake operations are described. A compute system is a combination of circuitry and executable instructions to perform an operation. The compute system includes a host device and an input/output (I/O) bus for managing connections with peripheral devices. A host device is a general computer (or a specialized computer) with a processor resource and a memory resource with instructions stored thereon that, when executed by the processor resource, cause operation of an operating system (OS). A host device, as used herein, includes particular programming to perform operations described herein.

[0007] A peripheral device is an electronic device capable of being connected to and communicating with the host device. For example, the peripheral device may be a human interface device (HID) to allow a user to input information to be processed by the host device, such as generation of signals corresponding to information input from interaction with a customer or employee corresponding to a retail transaction. Example peripheral devices include display devices, print apparatus, hubs, keyboards, computer mice, scanners, readers, cash drawers, personal identification number (PIN) entry device (e.g., PIN pads), etc.

[0008] In examples described herein, a display device is a device to present content visually. A display device, for example, may produce a visual representation of an image by operating light-emissive circuitry represented as a number of pixels based on processed image data. Example display devices may include a screen such as a liquid crystal display (LCD) panel, an organic light-emitting diode (OLED) panel, a micro light emitting diode (μ LED), or other display technology. In some examples, a display device may also include circuitry to operate the screen, such as a monitor scaler that causes an image to present (e.g., display) on a panel using source image data to determine a color to display for every pixel on the panel.

[0009] In examples described herein, a print apparatus is a device to print content on a physical medium (e.g., paper, textiles, a layer of powder-based build material, etc.) with a print material (e.g., ink or toner). In some examples, the physical medium printed on may be a web roll or a pre-cut sheet. In some examples, a print apparatus may be a three-dimensional (3D) print apparatus.

[0010] With such a wide arrange of peripheral devices available to be used with a host device, a method of managing interactions with the many peripheral devices may allow for saving power used to operate the many devices and for the host device to perform operations in a manner desired by the user. In general, some electronic devices may be operable in a low power state, in which the device operates using less power than a full working state. For example, some devices may automatically be placed into a sleep state after a time period of inactivity in order to save power. The low power state has less functionality than a fully powered working state. Example low power states are S1-S4 sleep states, where S0 is a fully powered, fully functional state and S5 is a completely off state. The low power state may allow for a signal to be received to “wake” the system, which represents causing the system to change operational mode to a state that uses more power, such as a

fully operational state. An example of such technology is known as wake-on-local area network (LAN), where a magic packet received at a network interface card (NIC) from the LAN connection will initiate a change of power state of the host device. Once power is increased, (e.g., fully restored) functionality of the device may be restored.

[0011] A peripheral device may generate a wake signal to change the power state of the host device. In a system with multiple peripherals connected, it may be desirable for a user to customize with peripherals are allowed to wake the host device and when those peripherals are allowed to wake the host device. For example, some legacy device may generate signals that appear similar to wake signals and cause the system to power on even when no processing or interaction with the host device is desired. Various examples described below relate to management of wake operations.

[0012] Figures 1 and 2 are block diagrams depicting example compute systems 100 and 200. Referring to Figure 1, the example compute system 100 of Figure 1 generally includes a first communication interface 102, a second communication interface 104, a processor resource 120, and a memory resource 110. In general, the memory resource 110 stores instructions, that when executed by the processor resource 120, manage wake settings of the communication interfaces 102 and 104.

[0013] As used herein, a communication interface is an electrical interface for communication with an electronic device. General purpose input/output (GPIO) interfaces may be used to connect controllers with other devices, for example. Example communication interfaces are input/output (I/O) ports, communication protocols useable by the host device, and controllers to control the same. Example classes of communication interfaces that may be used by a device may be serial, audio, display, and network. Example communication protocols include universal serial bus (USB), peripheral component interconnect (PCI), high-definition multimedia interface (HDMI), DISPLAYPORT, BLUETOOTH, ZIGBEE, and the like. An example component having a communication interface is a NIC, such as a PCI Express (PCIe) NIC or a USB NIC. The physical interface is capable of transmitting electrical signals between devices. A physical component of the interfaces may include a port (such as standard jacks like RJ45), a cable, and/or an antenna at which data is communicated and/or received. In some examples, a communication

interface may include multiple interface technologies, such as were a GPIO signal from a motherboard branches from a serial input/output (SIO) or embedded controller to communicate directly with a peripheral device.

[0014] The processor resource 120 is electrically coupled to the first communication interface 102 and the second communication interface 104 as well as the memory resource 110. A processor resource is any appropriate circuitry capable of processing (e.g., computing) instructions, such as one or multiple processing elements capable of retrieving instructions from a memory resource and executing those instructions. For example, the processor resource 120 may be a central processing unit (CPU) that enables managing wake operations by fetching, decoding, and executing the instructions 114 to establish wake settings. Example processor resources include at least one CPU, a semiconductor-based microprocessor, a programmable logic device (PLD), and the like. Example PLDs include an application specific integrated circuit (ASIC), a field-programmable gate array (FPGA), a programmable array logic (PAL), a complex programmable logic device (CPLD), and an erasable programmable logic device (EPLD). A processor resource may include multiple processing elements that are integrated in a single device or distributed across devices. A processor resource may process the instructions serially, concurrently, or in partial concurrence.

[0015] A memory resource 110 is operatively coupled to the processor resource 120. The memory resource 110 contains a set of instructions that are executable by the processor resource 120. A memory resource represents a medium to store data utilized and/or produced by the system 100. The medium is any non-transitory medium or combination of non-transitory media able to electronically store data, such as modules of the system 100 and/or data used by the system 100. For example, the medium may be a storage medium, which is distinct from a transitory transmission medium, such as a signal. The medium may be machine-readable, such as computer-readable. The medium may be an electronic, magnetic, optical, or other physical storage device that is capable of containing (i.e., storing) executable instructions. A memory resource may be said to store program instructions that when executed by a processor resource cause the processor resource to implement functionality of the system 100. A memory resource may be integrated in the same device as a processor resource or it may be separate but accessible to that device and the processor resource. A memory resource may be distributed across devices.

[0016] The memory resource 110 has instructions 112 and 114 stored thereon. The instructions 112 represent a device resource interface system (DRIS) to allocate device resources and connects firmware to an OS of a host device. Example DRISs include a basic input/output system (BIOS) and a unified extensible firmware interface (UEFI). For example, a BIOS may initiate a communication layer between a host device and a target peripheral device. Execution of DRIS may be performed at boot of the host device and may establish connections and operational settings of the host device during the boot process, such as during a power-on self-test (POST) stage. In some examples, the compute system 100 may be a hub, which may not include instructions 112 for a DRIS, yet include instructions 114 that when executed by the processor resource 120 to establish wake settings for the communication interfaces 102 and 104.

[0017] The instructions 114 to establish wake settings allow for managing the wake capabilities of the first communication interface 102 independent of the second communication interface 104. Rather than setting up the DRIS 112 to manage whether a host device is able to wake up or not for all communication interfaces (or none), the processor resource 120 is able execute the instructions 112 stored on the memory resource 110 to cause the compute system 100 to establish a first wake setting for the first physical communication interface 102 individually and/or independent of a second wake setting for the second physical communication interface 104. In an example, an option read-only memory (ROM), such as a NIC option ROM) accessible by the DRIS may be used to establish a wake setting of the communication interface (e.g., an interface driver or BIOS may control the wake setting by communicating with the option ROM corresponding to the particular communication interface. As used herein, a wake setting is a changeable characteristic of a communication interface corresponding to receiving a wake signal at the communication interface. For example, the first wake setting and the second wake setting corresponding to the communication interfaces 102 and 104 indicate a state of a limitation to operating a component of the compute system at a powered state. For another example, the wake setting may be a bit flip or flag that represents a state of whether a wake capability is enabled for the device. A wake setting represents a condition to trigger an action (where events like change power state and drop a packet are consider actions) and may correspond with one of a plurality of states of the component with respect to performing a wake operation, such as:

enabled wake operations, disabled wake operations, enabling wake operations based on a time or an event, and disabling wake operations based on a time or an event. In some examples, an event may be a change in power setting, such as one initiated based on a wake signal, where wake operations may be disabled during a designated time for transition after a wake signal has initiated a change in power state. The wake setting of a communication interface may be controlled via the DRIS 112, an interface controller, and/or interface driver as caused by execution of the instructions 114 to establish wake settings. For example, the interface driver or the BIOS may set the wake-on-LAN setting to off to turn off a wake-on-LAN capability, where one of the BIOS or interface driver may set wake-on-LAN to be enabled in order to support the wake capability and allow the device to wake. In other examples, multiple or all the interface management systems (e.g., the DRIS, the device controller, and the device driver) may be set to enable a wake capability in order for the wake capability to perform. The wake settings for each of the communication interfaces 102 and 104 may be different, and, thereby, the wake capability of each communication interface may be managed individually with independent settings from one another. In this manner, multiple wake signals may be ignored for a time period (or evaluated together after the time period expires) to reduce accidental repeated wake requests, for example.

[0018] A change in the wake setting may cause the communication interface to change power state (e.g., between a fully working state, a low power or sleep state, or an off state.) In some examples, the communication interface may be in a low power state that disregards any signals that do not correspond to a wake signal. In other words, the data corresponding to the signal may not be processed when the communication interface is in a low power state, though the signal may be identified by the DRIS or driver as a wake signal to cause a component to transition between power states. As used herein, a wake signal is any appropriate signal capable of meeting a wake condition as recognized by the communication interface. An example wake signal is a magic packet, such as a magic packet sent over ETHERNET to a NIC.

[0019] With the wake settings of the communication interfaces set independently, the processor resource 120 can execute instructions, such as 112 and/or 114, to manage a signal received by the first communication interface 102 based on the first wake setting and manage a signal received by the second

communication interface 104 based on the second wake setting. For example, the processor resource 120 may, via execution of the instructions representing methods described herein, block the signal received by the first physical communication interface 102 based on a user-defined disablement of receiving a magic packet and cause a component of the compute system to power up based on a signal received by the second physical communication interface 104. In that example, the execution of the DRIS 112 may allow for a user to select what wake capability is allowed for each communication interface 102 and 104 at boot and establish the settings as implemented in during instructions executed at a power-on self-test (POST) stage. For another example, the processor resource 120 may, via execution of the instructions representing methods described herein, block the signal received by the second physical communication interface 104 based on a user-defined disablement of receiving a magic packet and cause a component of the compute system to power up based on a signal received by the first physical communication interface 104. In this manner, the communication interfaces 102 and 104 may be managed individually. In some examples, the wake settings set at boot based on time or events may cause the operating system to trigger a change to wake setting in response to a time or an event as determined by the operating system. Such capabilities may allow for customization of peripheral capabilities and power saving schedules for a compute system. In this manner, any variety of peripheral devices may be used with a host device and a customized wake experience may be restricted individually, such as limiting wake operations to input received from a keyboard or computer mouse rather than a print apparatus and/or limiting to waking the host device to hours when a store is open to customers. In the later example, security may be hardened by disabling features that allow potential access to the host device during certain hours or events.

[0020] In some examples, functionalities described herein in relation to any of Figures 1-2 may be provided in combination with functionalities described herein in relation to any of Figures 3-8.

[0021] Figure 2 depicts the example system 200 may comprise a memory resource 210 operatively coupled to a memory resource 220. Referring to Figure 2, the memory resource 210 may contain a set of instructions that are executable by the memory resource 220. The set of instructions are operable to cause the memory resource 220 to perform operations of the system 200 when the set of instructions

are executed by the memory resource 220. The set of instructions stored on the memory resource 210 may be represented as a DRIS module 212, an establishment module 214, and management module 216. The DRIS module 212, the establishment module 214, and the management module 216 represent program instructions that, when executed, cause functions described with respect to the compute system 100 of Figure 1. For example, execution of the DRIS module 212 causes the processor resource 220 to execute a DRIS, execution of the establishment module 214 causes the processor resource 220 to establish wake settings for each communication interface, and execution of the management module 216 causes a processor resource to manage wake signals based on the wake settings of each communication interface. The memory resource 210 may carry out a set of instructions to execute the modules 212, 214, and 216 and/or any other appropriate operations among and/or associated with the modules of the system 200.

[0022] For example, the processor resource 220 may carry out a set of instructions to identify a plurality of communication interfaces with a wake capability, individually establish a wake setting for each of the communication interfaces, and manage a wake signal received by a first communication interface of the plurality of communication interfaces based on the wake setting corresponding to the first communication interface of the plurality of communication interfaces. As used herein, a wake capability represents a wake operation, an attribute of a wake operation, or a combination thereof that represents a functionality or feature corresponding to changing a power state of a component or device. In this manner, the wake setting may be a data representation of the wake capability, such as where the wake setting indicates whether to enable the wake capability or disable the wake capability.

[0023] For another example, the processor resource 220 may carry out a set of instructions to retrieve the wake setting from an entry in a table corresponding to the first communication interface of the plurality of communication interfaces when a magic packet is received at the first communication interface, cause a component to enter an S0 state in response to receiving a magic packet at the first communication interface when that first communication interface is in an enabled wake-on-LAN state, transition a device between power states based on the wake signal received at the first communication interface, disable the wake capability for a time period after the transition of the device between power states, and drop the magic packet

received at the first communication interface when the first communication interface is in a disabled wake-on-LAN state.

[0024] For yet another example, the processor resource 220 may carry out a set of instructions to determine which device class received a magic packet and allow or drop the magic packet based on the device class. A processor resource, such as a I/O controller, may manage each communication interface individually even when the settings are established based on groups, such as device class, because the system allows for individualized settings for each communication interface. Indeed, a rule to disable or enable wake operations via a communication interface based on a device class is made possible by providing the capability to individually manage wake operations for each communication interface.

[0025] Although these particular modules and various other modules are illustrated and discussed in relation to Figure 2 and other example implementations, other combinations or sub-combinations of modules may be included within other implementations. Said differently, although the modules illustrated in Figure 2 and discussed in other example implementations perform specific functionalities in the examples discussed herein, these and other functionalities may be accomplished, implemented, or realized at different modules or at combinations of modules. For example, two or more modules illustrated and/or discussed as separate may be combined into a module that performs the functionalities discussed in relation to the two modules. As another example, functionalities performed at one module as discussed in relation to these examples may be performed at a different module or different modules.

[0026] In the discussion herein, compute systems 100 and 200 of Figures 1 and 2 have been described as circuitry or a combination of circuitry and executable instructions. Such components may be implemented in a number of fashions. Looking at Figure 2, the executable instructions may be processor-executable instructions, such as program instructions, stored on the memory resource 210, which is a tangible, non-transitory computer-readable storage medium, and the circuitry may be electronic circuitry, such as memory resource 220, for executing those instructions. The instructions residing on a memory resource may comprise any set of instructions to be executed directly (such as machine code) or indirectly (such as a script) by a processor resource.

[0027] In some examples, the system 200 may include the executable instructions may be part of an installation package that when installed may be executed by a processor resource to perform operations of the system 200, such as methods described with regards to Figures 5-9. In that example, a memory resource may be a portable medium such as a compact disc, a digital video disc, a flash drive, or memory maintained by a computer device, such as a data server, from which the installation package may be downloaded and installed. In another example, the executable instructions may be part of an application or applications already installed. A memory resource may be a non-volatile memory resource such as read-only memory (ROM), a volatile memory resource such as random-access memory (RAM), a storage device, or a combination thereof. Example forms of a memory resource include static RAM (SRAM), dynamic RAM (DRAM), electrically erasable programmable ROM (EEPROM), flash memory, or the like. A memory resource may include integrated memory such as a hard drive (HD), a solid-state drive (SSD), or an optical drive.

[0028] Figure 3 depicts an example compute system 300 in which various example communication interfaces 332, 338, and 340 are coupled to the example compute system 300. The compute system 300 generally includes a host computer device 330 and a plurality of interfaces including a PCIe interface and a USB interface. The example compute system 300 includes a PCIe NIC 332, a first USB NIC 338, and a second USB NIC 340 coupled to the host device 330. The example compute system 300 depicts multiple USB hubs 334 and 336. In other examples, a single hub may include multiple interface components; in other words, a single hub may include a plurality of I/O ports, such as the USB NIC B 238 and USB NIC C 240, manageable by a controller to manage wake operations for each communication interface coupled to the host device 330. A controller, as used herein, is a combination of circuitry and a control program that when executed by the circuitry causes operations of the control program to be performed.

[0029] The host device 330 includes a BIOS 322, a power deliver (PD) controller 324, and a table manager 326. The PD controller 324 represents a combination of circuitry and executable instructions to provide power to the interface components, including the controllers over each interface. The PD controller 324 manages delivery of power to components of the host device 330. The PD controller may be programmed to maintain an operational state of the devices as well as a

power state of the device. For example, the PD controller 324 may be programmed to set a state of a NIC to allow or drop a wake signal based on a setting established by the BIOS 322. For another example, the PD controller 324 may be coupled to devices PCIe NIC device ID A, USB NIC device ID B, and USB NIC device ID C and the BIOS 322 communicates with the PD controller 324 to determine which of communication interfaces received a wake signal indicating to wake the component and/or which communication interfaces are allowed to respond to a wake signal while the host device (or one of the USB hubs 334 and 336) is in a low power state.

[0030] The BIOS 322 may include instructions to setup each of the network controllers 332, 338, and 340 using the device identifiers (IDs). The BIOS 322 may establish wake settings based on entries of a table of wake settings corresponding to the available interface devices, which may be displayed as an option during the boot process. Such a table is managed by a table manager 326. In some examples, the BIOS 322 may manage the wake setting of the interfaces based on the class or type of communication interface. For example, the BIOS 322 may coordinate power state and operations of the devices with the PD controller 324 such that the PCIe interfaces are allowed to receive and react to a wake signal while USB interfaces are denied taking action in response to a wake signal.

[0031] The table manager 326 is a combination of circuitry and executable instructions to maintain (e.g., generate, update, etc.) a table for wake settings for each port of the host device 330, the hub A 334, and hub B 336 in response to detection of a device coupled to the hubs. An example representation of table, which may be an example of a visual presentation by the BIOS 322, is depicted in Figure 4. the table including a network identifier and a device class for each device coupled to each port.

[0032] The arrow lines between the host device 330 and devices 332, 334, 336, 338, and 340 may be a link that generally represents one or a combination of a cable, wireless connection, fiber optic connection, or remote connections via a telecommunications link, an infrared link, a radio frequency link, or any other connectors of systems that provide electronic communication. A link may include, at least in part, intranet, the Internet, or a combination of both. The link may also include intermediate proxies, routers, switches, load balancers, and the like.

[0033] Figure 4 an example table 400 for use with an example compute system. In the example table 400, the columns include an interface identifier, an

interface class, and a port status. The interface identifier A refers to device ID A 332 of Figure 3, the interface identifier B refers to device ID B 338 of Figure 3, and the interface identifier C refers to device ID C 340 of Figure 3. A class of the interface is listed with references to the type of interface corresponding to the unique identifier listed in the interface identifier column. A port status is also listed in the row of the corresponding unique identifier. In the example of Figure 4, device ID A has wake-on-LAN enabled, device ID B has wake-on-LAN disabled, and device ID C has wake-on-LAN disabled between 5 am and 6 pm. In this manner, the DRIS may establish the wake setting for the boot session of the host device for devices ID A and B (enabled and disabled, respectively) and device ID C changes wake setting based on a time of day. In other examples, the device ID C could be allowed to change wake settings based on an event, such as performing a first financial transaction or printing a summary report of transactions for the day.

[0034] Figures 5-9 are flow diagrams depicting example methods 500, 600, 700, 800, and 900 of managing wake operations. Referring to Figure 5, example method 500 of managing wake operations may generally comprise identifying a device identifier for each communication interface, determining a wake capability for each communication interface, mapping a wake setting to each wake capability, and setting a state of each communication interface corresponding to the mapped wake setting. The method 500 may be performable by a processor resource, such as processor resource 120 of Figure 1 or a PD controller, such as PD controller 324 of Figure 3.

[0035] At block 502, a device identifier is identified for each communication interface. A host device may include a number of communication interfaces and may be operated, via the DRIS, to collect a unique identifier for each communication interface device. This may also include a hierarchy of communication interfaces, such as information regarding devices coupled to hubs directly coupled to the host device.

[0036] At block 504, a wake capability is determined for each communication interface. For example, a NIC option ROM may be checked for a wake capability of a communication interface. Each communication interface may have a set of capabilities to allow features corresponding to wake operations (e.g., wake capabilities). For example, a wake capability of a device may be to have wake-on-LAN enabled or disabled. In other examples, the device may be more advanced to

allow for time-based or event-based enablement or disablement of wake-on-LAN. In further examples, some communication interfaces may have different wake capabilities based on the power state of the component and, yet some others may not have any wake capabilities regardless of the power state of the component.

[0037] At block 506, a wake setting is mapped to each wake capability with the corresponding device identifier. A processor resource, such as processor resource 120, may manage the mapping of a wake setting to a device identifier. For example, if a device can be enabled with wake-on-LAN or disabled, then an enabled wake setting and a disabled wake setting may be mapped to the corresponding device identifier. In this manner, the wake setting may switch among multiple wake capabilities to invoke a state of the component to receive and handle a wake signal. The mapping may be exemplified by a table, such as table 400 of Figure 4.

[0038] At block 508, a state of each communication interface is set corresponding to the mapped wake setting of block 506 based on the corresponding device identifier. For example, a controller of the communication interface may set the component power state in accordance with a corresponding wake setting (e.g., as selected at boot or by default setting). In this manner, each communication interface with wake capabilities may be individually set to enable or disable wake-on-LAN independent of the wake-on-LAN state of the other communication interfaces.

[0039] Figure 6 includes blocks similar to blocks of Figure 5 and provides additional blocks and details. In particular, Figure 6 depicts additional blocks and details generally regarding causing display of wake-on-LAN capabilities, determining a time or event to enable or disable a wake capability, managing a signal received at the communication interface, and transition the power state of a component. Blocks 602, 604, 610, and 612 are similar to blocks 502, 504, 506, and 508 of Figure 5 and, for brevity, their respective descriptions are not repeated in their entirety. The method 600 may be performable by a processor resource, such as processor resource 120 of Figure 1 or a PD controller, such as PD controller 324 of Figure 3.

[0040] At block 606, a wake-on-LAN field is caused to display for each communication interface corresponding to each wake capability determined at block 604. Such a presentation at boot may allow for a user to select and verify which interfaces are allowed to perform a wake operation in response to receiving a wake signal at the communication interface.

[0041] At block 608, a time (or an event) to enable or disable the corresponding wake capability of one of the communication interfaces is determined. Such an event or time may be monitored by an operating system or BIOS, and the processor resource executing the method 600 may receive a signal triggered by the monitored time or event to cause transition of the operational state of the communication interface.

[0042] At block 610, a state (e.g., a power state and/or an operation state) corresponding to each wake capability is mapped based on a device type and a user selection received in response to a change to the wake-on-LAN field displayed at block 606. In this manner, the user may customize the interface capabilities with regards to wake-on-LAN, individually for each interface.

[0043] At block 614, a signal received by a communication interface is managed based on the wake setting for that communication interface, as set at block 612. For example, the wake signal may be dropped for any device identifier with a wake setting set to disable.

[0044] At block 616, a power state of a component is transitioned based on a signal received at the communication interface and a wake setting for that communication interface. For example, a processor resource may cause a component to transition from a low power state to a fully powered state based on receiving a wake signal when the wake setting for that communication interface is set to enable. In some examples, the wake operation could also be a sleep operation to transition the component or host device to a low power state from a fully powered state, for example.

[0045] Referring to Figure 7, example method 700 of managing wake operations may generally comprise causing a table to display wake information for a communication interface and updating a state of the communication interface. The method 700 may be performable by a processor resource, such as processor resource 120 of Figure 1 or a PD controller, such as PD controller 324 of Figure 3.

[0046] At block 702, a table is caused to be displayed, such as on a display device coupled to a host device. A processor resource may cause a table to display with a matrix including columns for device identifiers, a corresponding device class, and a corresponding wake capability and/or wake setting for each communication interface.

[0047] At block 704, the state corresponding to each communication interface is updated in response to a change in the table for the corresponding communication interface of the change. In this manner, the user may have direct customization of the status and wake capabilities of the communication interfaces by adjusting what is displayed at block 702, such as using a keyboard or mouse input device.

[0048] Referring to Figure 8, example method 700 of managing wake operations may generally comprise identifying which one of the plurality of communication interfaces received a magic packet, determining if the communication interfaces is authorized to change a power state, and causing the host device to power on when the communication interface is authorized. The method 800 may be performable by a processor resource, such as processor resource 120 of Figure 1 or a PD controller, such as PD controller 324 of Figure 3.

[0049] At block 802, one of the plurality of communication interfaces that received a magic packet is identified, such as in response to a processor resource identifying a magic packet was sent to the host device from an I/O port. In this manner, the processor resource may collect all signals from the communication interfaces and determine how to manage those signals based on the source of the signals and the wake setting associated with the source. A hub connected to the host device may provide an indication that a wake signal was received. In some examples, the host device may interrogate the hub to provide further information about which port received the wake signal to determine whether to change the power state of a component of the host device.

[0050] At block 804, a determination is made by a processor resource as to whether the one of the plurality of communication interfaces is authorized to change a power state of the host device in response to a magic packet. The processor resource may review a wake setting corresponding to the source of the magic packet to determine whether the interface is authorized to cause a change in power state (e.g., a wake signal received at that particular NIC is allowed to wake the host device or not).

[0051] At block 806, the host device is caused to power on when the one of the plurality of communication interfaces is authorized to change the power state of the host device in response to the magic packet (as determined at block 804). Indeed, once the source of the magic packet is identified as authorized to perform a wake operation in response to a wake signal, then the processor resource may send

a signal to cause the host device to switch from a low power state to a fully functional, fully powered state.

[0052] Referring to Figure 9, example method 900 of managing wake operations may generally comprise setting a state of each communication interface to a wake setting, managing signals received at the communication interfaces, transition a power state, and disabling wake-on-LAN capability for a time period after transitioning the power state. The method 900 may be performable by a processor resource, such as processor resource 120 of Figure 1 or a PD controller, such as PD controller 324 of Figure 3.

[0053] At block 902, a state of each communication interface is set to a corresponding wake setting based on the device identifier of the communication interface. At block 904, a signal received at the communication interface is managed based on the wake setting set at block 904.

[0054] At block 906, the communication interface is transitioned between power states based on the signal received at the communication interface and the wake setting for that communication interface (e.g., when the communication interface is authorized to cause a power transition and has received a wake signal associated with performing the power transition).

[0055] At block 908, the wake capability is disabled for a time period after a transition between power states at block 906. By disabling wake capabilities (such as wake-on-LAN) for a transition time period, the compute system may allow for smooth transitions between states without interruptions to power state change requests, causing devices to quickly turn on or off again in a manner that exceeds the technical capabilities of one of the host device components, or otherwise providing a disapproved user experience. By being able to manage wake signal operations individually for each communication interface, and customize to times or events, any variety of peripheral devices may be used with a host device and a customized wake experience may be had.

[0056] Although the flow diagrams of Figures 5-9 illustrate specific orders of execution, the execution order may differ from that which is illustrated. For example, the execution order of the blocks may be scrambled relative to the order shown. Also, the blocks shown in succession may be executed concurrently or with partial concurrence. All such variations are within the scope of the present description.

[0057] All the features disclosed in this specification (including any accompanying claims, abstract and drawings), and/or all the elements of any method or process so disclosed, may be combined in any combination, except combinations where at least some of such features and/or elements are mutually exclusive.

[0058] The terms “include,” “have,” and variations thereof, as used herein, mean the same as the term “comprise” or appropriate variation thereof. Furthermore, the term “based on,” as used herein, means “based at least in part on.” Thus, a feature described as based on some stimulus may be based only on the stimulus or a combination of stimuli including the stimulus. The article “a” as used herein does not limit the element to a single element and may represent multiples of that element. Furthermore, use of the words “first,” “second,” or related terms in the claims are not used to limit the claim elements to an order or location, but are merely used to distinguish separate claim elements.

[0059] The present description has been shown and described with reference to the foregoing examples. It is understood that other forms, details, and examples may be made without departing from the spirit and scope of the following claims.

CLAIMS

What is claimed is:

1. A compute system comprising:
 - a first physical communication interface;
 - a second physical communication interface;
 - a memory resource with instructions stored thereon, the instructions representing a device resource interface system to allocate device resources to an operating system; and
 - a processor resource, the processor resource to execute the instructions stored on the memory resource to cause the compute system to:
 - establish a first wake setting for the first physical communication interface independent of a second wake setting for the second physical communication interface, the first wake setting and the second wake setting to indicate a state of a limitation to operating a component of the compute system at a powered state;
 - manage a signal received by the first physical communication interface based on the first wake setting; and
 - manage a signal received by the second physical communication interface based on the second wake setting.
2. The compute system of claim 1, wherein:
 - the signal received by the first physical communication interface is blocked based on a user-defined disablement of receiving a magic packet and the signal received by the second physical communication interface causes the component of the compute system to be powered; or
 - the signal received by the second physical communication interface is blocked based on a user-defined disablement of receiving a magic packet and the signal received by the first physical communication interface causes the component of the compute system to be powered.
3. The compute system of claim 1, further comprising:
 - a host device;

a hub coupled to the host computer, the hub having a plurality of input/output ports; and

a table manager to generate a table for wake settings for each port of the hub in response to detection of a device coupled to the hub, the table including a network identifier and a device class for each device coupled to each port.

4. The compute system of claim 1, wherein:

the wake setting represents one of a plurality of states of the component including:

enabling wake-on-LAN based on a time or an event; and

disabling wake-on-LAN based on a time or an event;

the instructions to establish the settings is implemented in instructions executed at a power-on self-test (POST) stage; and

the operating system triggers a change to wake setting in response to a time or an event determined by the operating system.

5. The compute system of claim 1, comprising:

a power delivery controller coupled to the first electronic device and the second electronic device, the device resource interface system to communicated via the power delivery controller to determine which of the first physical communication interface and the second physical communication interface received the signal indicating to wake the component.

6. A non-transitory computer-readable storage medium (NCRSM) comprising a set of instructions executable by a processor resource to:

identify a plurality of communication interfaces with a wake capability;

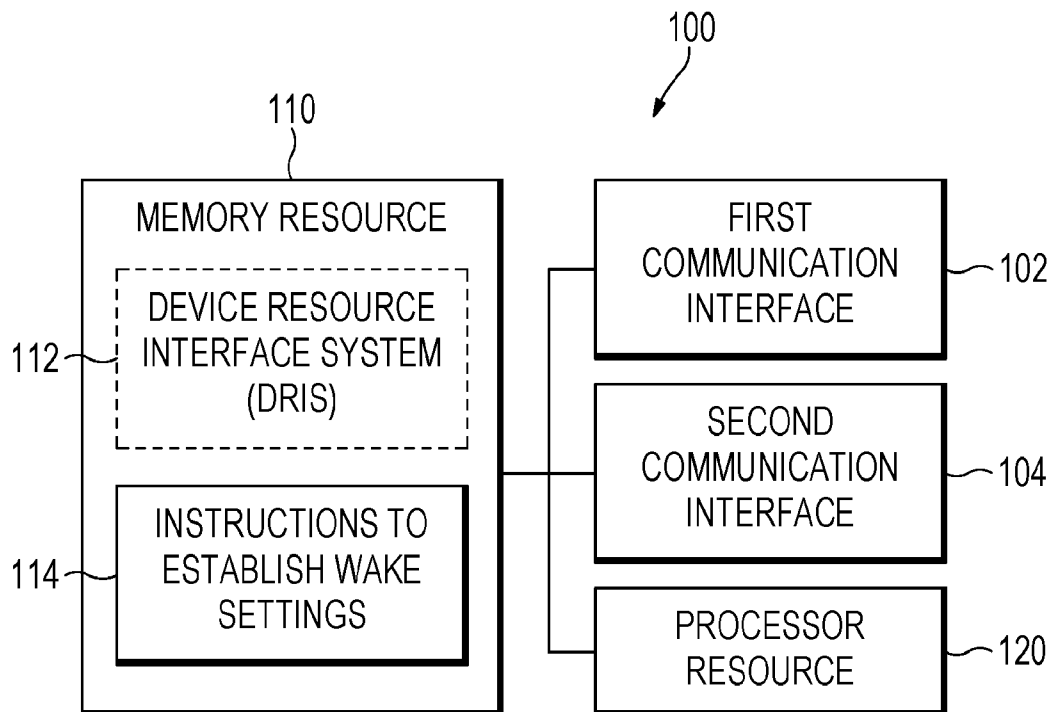
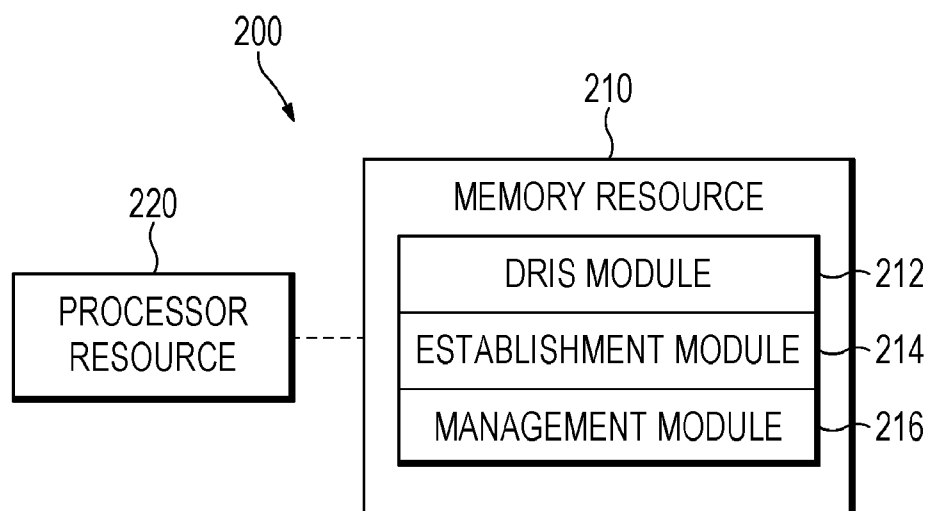
individually establish a wake setting for each of the communication interfaces, the wake setting to indicate whether to enable the wake capability or disable the wake capability, and

manage a wake signal received by a first communication interface of the plurality of communication interfaces based on the wake setting corresponding to the first communication interface of the plurality of communication interfaces.

7. The NTCRSM of claim 6, wherein the set of instructions is executable by the processor resource to:
 - cause a component to enter an S0 state in response to receiving a magic packet at the first communication interface when that first communication interface is in an enabled wake-on-LAN state; and
 - drop the magic packet received at the first communication interface when the first communication interface is in a disabled wake-on-LAN state.
8. The NTCRSM of claim 6, wherein the set of instructions is executable by the processor resource to:
 - retrieve the wake setting from an entry in a table corresponding to the first communication interface of the plurality of communication interfaces when a magic packet is received at the first communication interface.
9. The NTCRSM of claim 6, wherein the set of instructions is executable by the processor resource to:
 - determine which device class received a magic packet; and
 - allow or drop the magic packet based on the device class.
10. The NTCRSM of claim 6, wherein the set of instructions is executable by the processor resource to:
 - transition a device between power states based on the wake signal received at the first communication interface; and
 - disable the wake capability for a time period after the transition of the device between power states.
11. A method of managing wake operations with a plurality of communication interfaces coupled to a host device, the method comprising:
 - identifying a device identifier for each communication interface;
 - determining a wake capability for each communication interface;
 - mapping a wake setting to each wake capability with the corresponding device identifier; and
 - setting a state of each communication interface corresponding to the mapped wake setting based on the corresponding device identifier.

12. The method of claim 11, comprising:
- causing a wake-on-LAN field to display for each communication interface corresponding to each wake capability; and
 - mapping the state corresponding to each wake capability based on a device type and a user selection received in response to a change to the wake-on-LAN field displayed.
13. The method of claim 11, comprising:
- determining a time to enable or disable the corresponding wake capability of one of the plurality of communication interfaces; or
 - determining an event to trigger a change to the corresponding wake capability of one of the plurality of communication interfaces.
14. The method of claim 11, comprising:
- causing a table to display, the table including the corresponding device identifier, a corresponding device class, and the corresponding wake capability for each communication interface; and
 - updating the state corresponding to each communication interface in response to a change in the table for the corresponding communication interface.
15. The method of claim 11, comprising:
- identifying which one of the plurality of communication interfaces received a magic packet; and
 - determining whether the one of the plurality of communication interfaces is authorized to change a power state of the host device in response to a magic packet; and
 - causing the host device to power on when the one of the plurality of communication interfaces is authorized to change the power state of the host device in response to the magic packet.

1/6

**FIG. 1****FIG. 2**

2/6

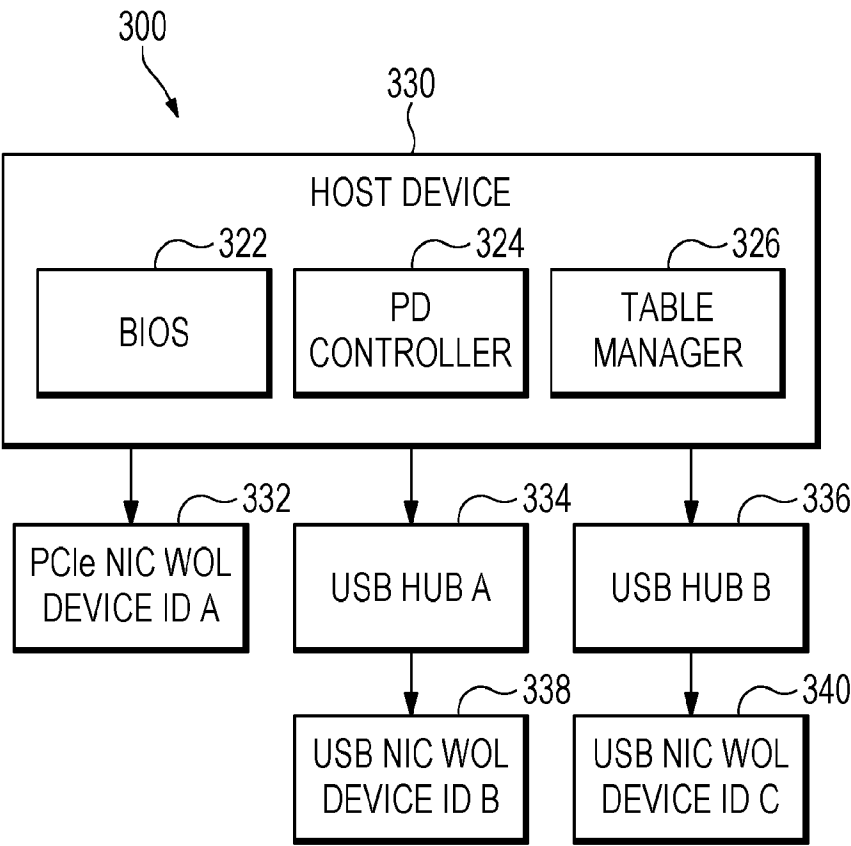


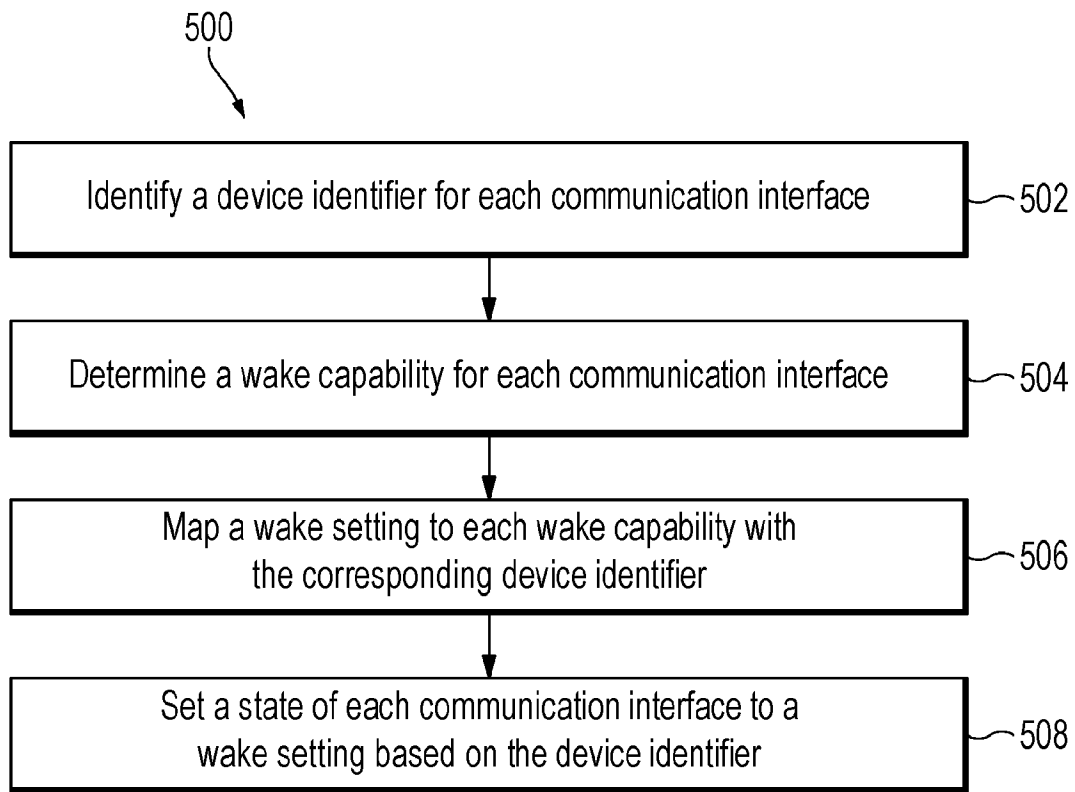
FIG. 3

400

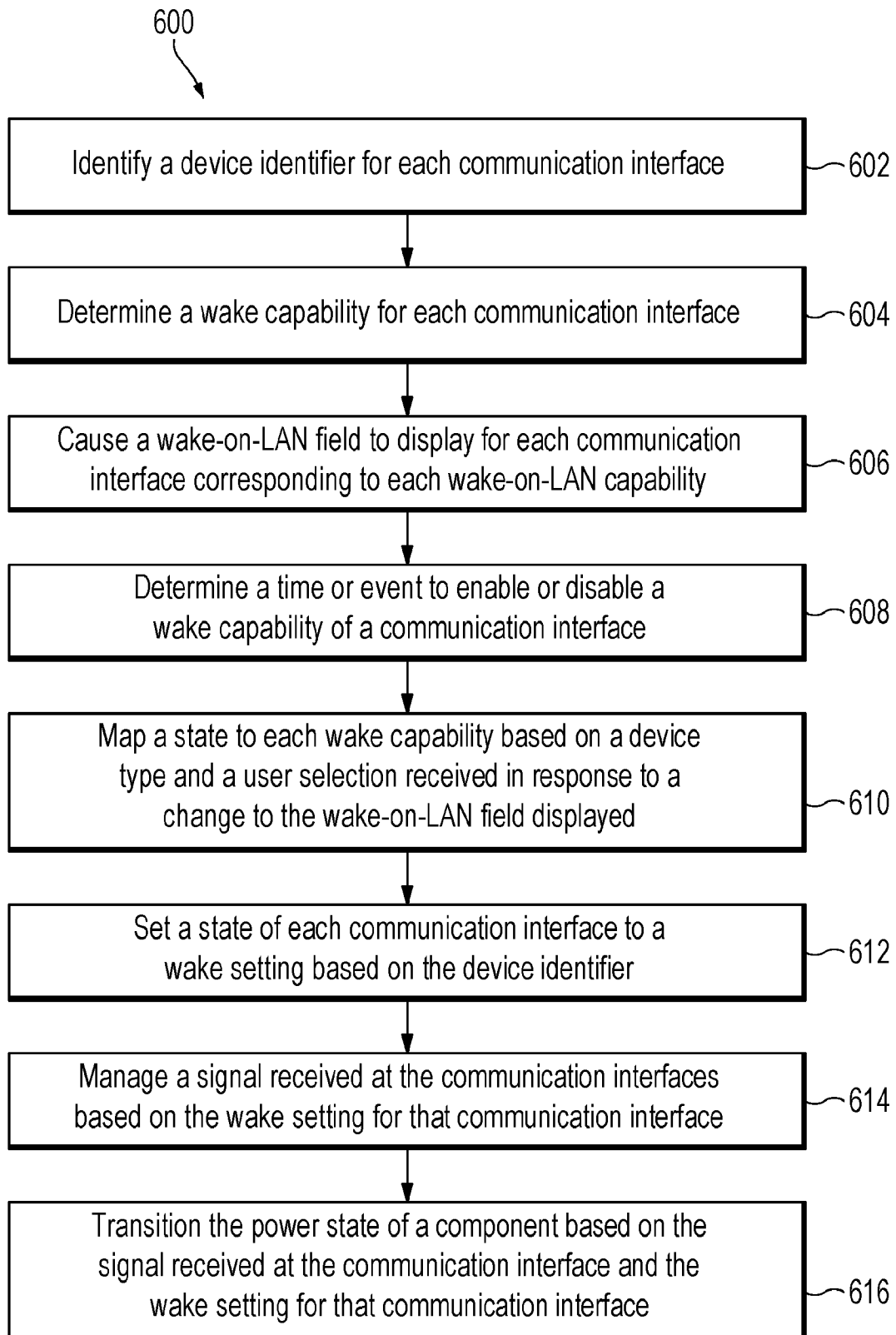
Interface Identifier	Interface Class	Port Status
A	PCIe	Enable
B	USB	Disable
C	USB	Disable between 5:00 and 18:00

FIG. 4

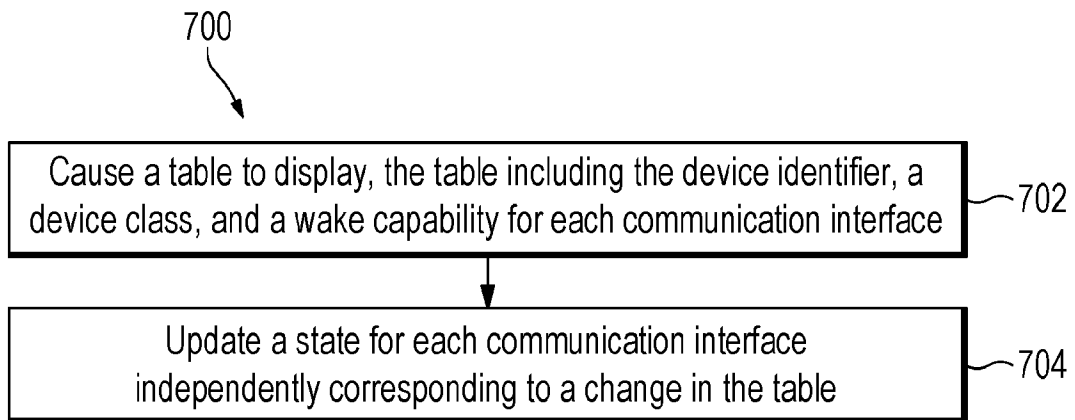
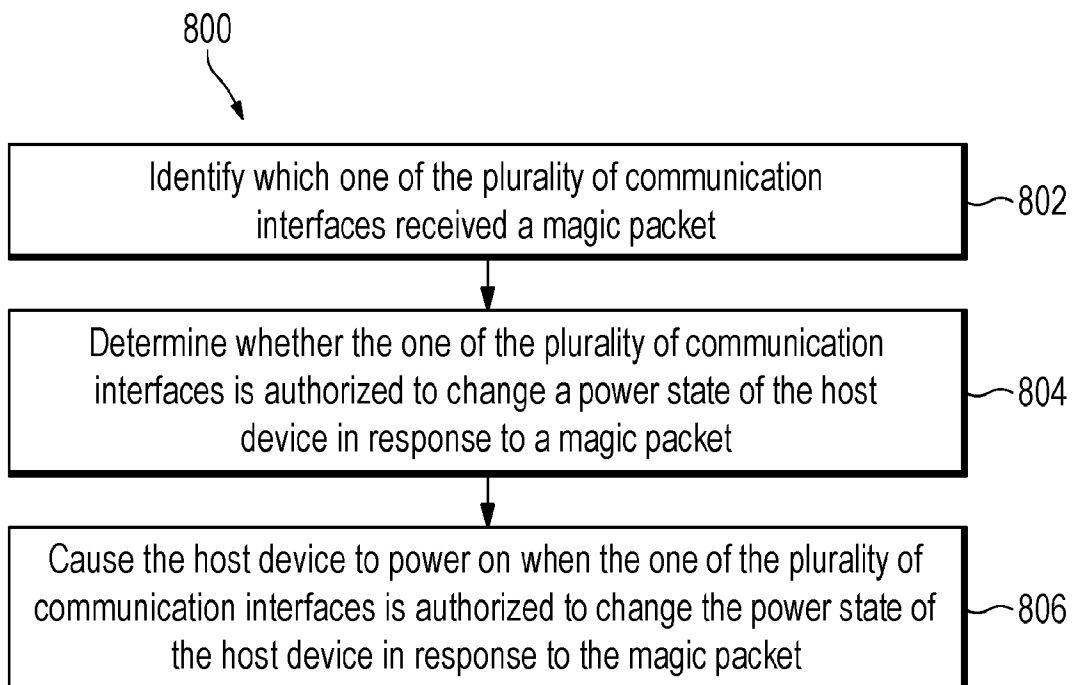
3/6

**FIG. 5**

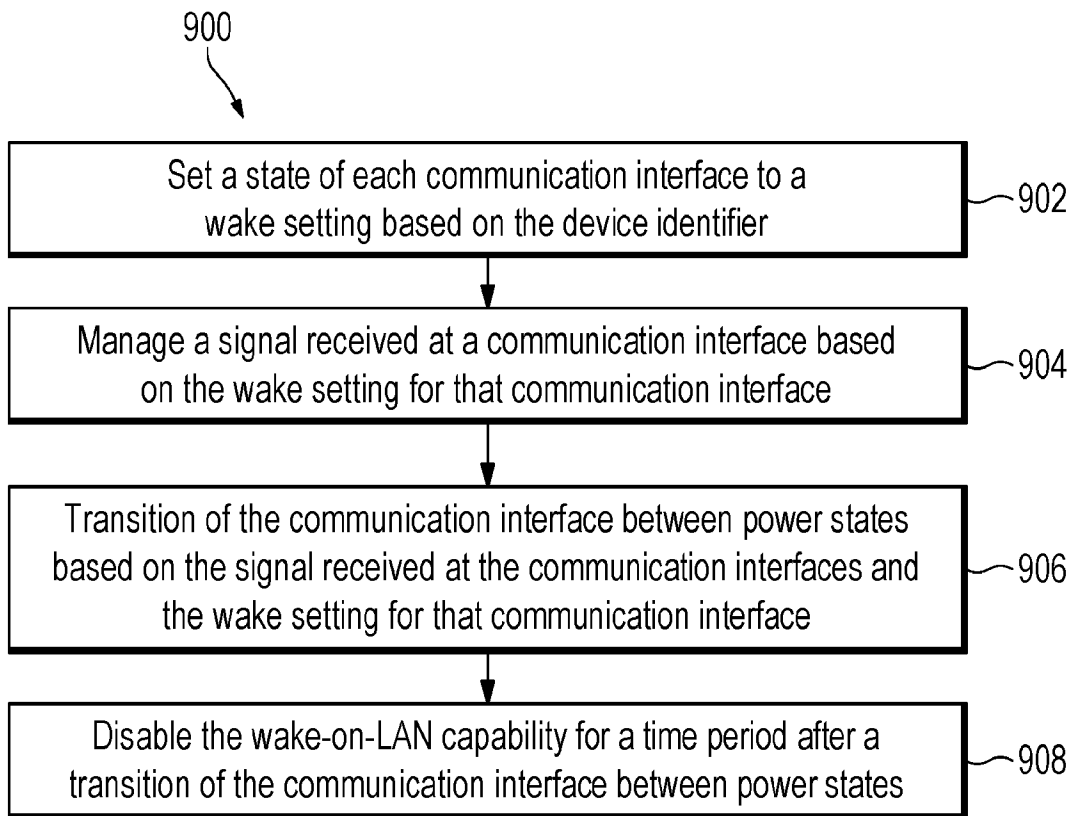
4/6

**FIG. 6**

5/6

**FIG. 7****FIG. 8**

6/6

**FIG. 9**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2020/053311

A. CLASSIFICATION OF SUBJECT MATTER		
G06F 1/3246 (2019.01) G06F 1/3287 (2019.01) 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 1/00-1/3296, H04N 1/00-1/64		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
PatSearch (RUPTO Internal), USPTO, PAJ, Espacenet, Information Retrieval System of FIPS		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2011/0090830 A1 (CANON KABUSHIKI KAISHA) 21.04.2011, paragraphs [0016], [0023], [0025], [0040]-[0043], [0045]-[0046], [0062], figure 1	1-15
A	KR 100997892 B1 (HYUNARM ENTI LTD.) 03.12.2010	1-15
A	US 2011/0185370 A1 (ELIEZER TAMIR et al.) 28.07.2011	1-15
A	JP 2005297487 A (CANON INC) 27.10.2005	1-15
☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.		
* Special categories of cited documents:	"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	
"A" document defining the general state of the art which is not considered to be of particular relevance		
"D" document cited by the applicant in the international application		
"E" earlier document but published on or after the international filing date		
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)		
"O" document referring to an oral disclosure, use, exhibition or other means		
"P" document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search	Date of mailing of the international search report	
13 May 2021 (13.05.2021)	27 May 2021 (27.05.2021)	
Name and mailing address of the ISA/RU: Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37	Authorized officer A. Grigoriev Telephone No. 8(495)531-65-15	