

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
5 April 2007 (05.04.2007)

PCT

(10) International Publication Number
WO 2007/036071 A1

(51) International Patent Classification:
G06F 13/10 (2006.01)

(21) International Application Number:
PCT/CN2005/001622

(22) International Filing Date:
29 September 2005 (29.09.2005)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US): **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **WANG, Brett** [CN/CN]; Room 204, 89 Tianshan, No.5, Newly State, Changning District, Shanghai 200336 (CN). **WU, Min-qiang** [CN/CN]; Room 303, #2, Lane 600, Maotal Road, Shanghai 200336 (CN). **ZHANG, Winters** [CN/CN];

350-3201, Guiping Rd., Shanghai 200000 (CN). **ZHANG, Hongbing** [CN/CN]; Shanghai 200000 (CN).

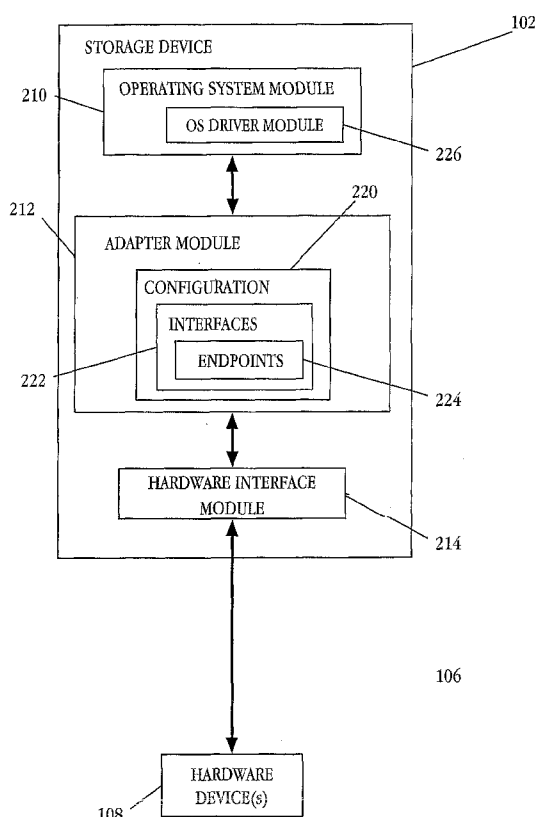
(74) Agent: **CHINA PATENT AGENT (H.K.) LTD.**; 22/F, Great Eagle Centre, 23 Harbour Road, Wanchai, Hong Kong (CN).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,

[Continued on next page]

(54) Title: SYSTEMS AND METHODS FOR IMPROVED HARDWARE DEVICE CONNECTIVITY



(57) Abstract: Systems and methods are described herein to provide for improved hardware device connectivity. Other embodiments include apparatus and system for receiving messages from an operating system and sending messages to one or more hardware devices. Further embodiments include methods for receiving messages from an operating system regarding operations on a hardware device and responding to those messages. Other embodiments include methods for receiving interrupt messages and mapping those interrupt messages to hardware devices. Other embodiments are described and claimed.

WO 2007/036071 A1



FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,
RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,
GN, GQ, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *with international search report*

SYSTEMS AND METHODS FOR IMPROVED HARDWARE DEVICE CONNECTIVITY

TECHNICAL FIELD

[0001] Various embodiments described herein relate generally to device driver operations and more particularly for providing improved connectivity for hardware devices.

BACKGROUND

[0002] Presently, a computer user extending functionality of their computing device seeks to do so by adding a hardware device to the computing device. The computer user may wish to have scanning capability and would then hook up a scanner to their device and then proceed to scan their pictures or documents. Many different standards for connectivity of these hardware devices have been proposed and adopted.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] In the drawings, which are not necessarily drawn to scale, like numerals describe substantially similar components throughout the several views. Like numerals having different letter suffixes represent different instances of substantially similar components. The drawings illustrate generally, by way of example, but not by way of limitation, various embodiments discussed in the present document.

[0004] FIG. 1 is a high level block diagram of a system according to embodiments of the present invention;

[0005] FIG. 2 is a high level block diagram of a device according to embodiments of the present invention;

[0006] FIG. 3 is a flowchart of a method according to embodiments of the present invention;

[0007] FIG. 4A is a flowchart of a method according to embodiments of the present invention;

[0008] FIG. 4B is a flowchart of a method according to embodiments of the present invention;

[0009] FIG. 5A is an exemplary dataflow diagram to be carried out on a device according to embodiments of the present invention; and

[0010] FIG. 5B is an exemplary dataflow diagram to be carried out on a device according to embodiments of the present invention.

DETAILED DESCRIPTION

[0011] In the following detailed description of, reference is made to the accompanying drawings which form a part hereof, and in which are shown, by way of illustration, specific preferred embodiments in which the subject matter may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice them, and it is to be understood that other embodiments may be utilized and that logical, mechanical, and electrical changes may be made without departing from the spirit and scope of the present disclosure. Such embodiments of the inventive subject matter may be referred to, individually and/or collectively, herein by the term "invention" merely for convenience and without intending to voluntarily limit the scope of this application.

[0012] FIG. 1 is a high level block diagram of a system according to embodiments of the present invention. In an embodiment, a computing device 100 includes a storage device 102 and a processor, 104. In an embodiment, the storage device 102 and the processor are communicatively coupled through a communications bus 106. In an embodiment, a hardware device 108 is coupled through any suitable means to the computing device 100. The communications bus 106 represent one or more busses, e.g., Universal Serial Bus (as detailed in the Universal Serial Bus Specification, Revision 1.1, published September 23, 1998 and later versions, hereinafter USB specification), FireWire, PCI, ISA (Industry Standard Architecture),

X-Bus, EISA (Extended Industry Standard Architecture), or any other appropriate bus and/or bridge (also called a bus controller).

[0013] In an embodiment, the computing device 100 is configured to execute machine readable instructions contained on the storage device 102 implementing an operating system and other software applications, including routines and software operations other than an operating system that in some configurations are capable of controlling input/output operations, routines, or interrupt handlers on the computing device 100. Operating systems may include operating systems based on Windows®, UNIX, Linux, Macintosh®, and operating systems embedded on a processor. The computing device may include, without limitation, desktop PC, server PC, PDA, etc. In one embodiment, the machine readable instructions are executed by the processor 104. Though drawn as single blocks in FIG. 1 it is understood that the computing device 102 may contain more than one storage device 102 and more than one processor 104. Further, the machine readable instructions executing by the processor 104 of the computing device 102 may be contained on a storage device 102 external to the computing device 100, such as across a network.

[0014] In an embodiment, the hardware device 108 is any electronic device that is configured to be coupled to the computing device 100 in any suitable manner thereby extending the functionality of the computing device. Examples of hardware devices 108 include, without limitation, mass storage device, such as external hard drives; multimedia player devices, such as mp3 players; input devices, such as keyboards and mice; and display devices, such as monitors and many other internal and external devices. An example of a hardware device internal to the computing device 100 may be a modem, etc.

[0015] FIG. 2 is a high level block diagram of a device according to embodiments of the present invention. In an embodiment, the storage device 102 includes machine-readable instructions contained therein to execute the following modules: operating system module 210, adapter module 212 and hardware interface module 214. In a further embodiment, the adapter module 212 includes a configuration 220, one or more interfaces 222 and one or more endpoints 224. In one embodiment, the

configuration, one or more interfaces and the one or more endpoints are hard-coded and can not be changed. In another embodiment, the configuration, one or more interfaces, and one or more endpoints are able to be altered in any suitable manner. The operating system module 210 includes at least one OS driver module 226. The modules may include hardware circuitry, single or multi-processor circuits, memory circuits, software program modules and objects, firmware, and combinations thereof, as desired by the architect of the computing device 100 and as appropriate for particular implementations of various embodiments. The modules may be included in a system operation simulation package such as a software electrical signal simulation package, a signal transmission-reception simulation package, or any combination of software and hardware used to simulate the operation of various potential embodiments. Such simulations may be used to characterize or test the embodiments, for example.

[0016] In an embodiment, the adapter module 212 includes hardware device data corresponding to the type of hardware device 108 connected to adapter module 212 via hardware interface module 214. The hardware device data includes configuration data 220 that corresponds to the protocol type for a particular hardware device. By way of example, but not by way of limitation, adapter module 212 may include configuration data corresponding to the USB specification. The configuration data includes hardware information, interface data, endpoint data, communications protocol, and other information corresponding to the protocol type utilized by the particular hardware device.

[0017] The configuration data includes a description of how many interfaces of the same type are connected to the computing device. For each type of hardware device 108, there is one adapter module 212 that includes configuration data 220 for the corresponding type of protocol. In an embodiment, a single hardware device 108 is coupled to a single adapter module 212. In another embodiment, multiple hardware devices 108 each using a different type of communications protocol are coupled to the computing device which contains multiple adapter modules and each one of the multiple adapter modules corresponds to one type of communications protocol.

[0018] In an alternate embodiment, multiple hardware devices 108, each using the same type of communications protocol, are coupled to a single adapter module 212. In this case, the configuration data describes one or more interfaces of the same type. For each of the one or more interfaces of the same type, there may be one or more endpoints defined by the USB specification. Endpoints are the basic input/output unit in a hardware device.

[0019] In an embodiment, the adapter module 212 is coupled to and communicates with the OS driver module 226 as a USB device. In a further embodiment, the adapter module 212 maintains data about the hardware device 108 in the form of hard-coded hardware device data so that the adapter module 212 can respond to configuration and data requests from the OS driver module 226 and higher level operations. As depicted in FIG. 2, the adapter module 212 includes hard-coded configuration data 220 describing one or more hard-coded interfaces 222. The one or more hard-coded interfaces 222 describes the functions that the hardware device 108 can perform and includes one or more hard-coded endpoints 224 and are the basic I/O unit for the type of hardware device.

[0020] In an embodiment, the storage device 102 further includes a hardware interface module 214 coupled through an appropriate communications bus to the hardware device 108. The hardware interface module 214 is configured to receive messages from the adapter module 212 as executed by the processor using the appropriate communications protocol for the hardware device and send those messages to the hardware device 108. In an embodiment, the hardware interface module 214 is external to the hardware device 108.

[0021] FIG. 3 is a flowchart of a method according to embodiments of the present invention. In an embodiment, the operations contemplated in FIG. 3 are to be carried out by an adapter module 212 as depicted in FIG. 2 and described above. At block 305 a configuration request message is sent to the adapter module 212. The adapter module 212 operates with respect to the operating system module 210 and the OS driver module 226 as a hardware device. In one embodiment, the configuration request message is an USB Request Block (URB). The configuration

request message may be sent for any number of reasons to include, without limitation, system start-up, system re-start and attachment of a new hardware device 108. The adapter module 212 holds the configuration request message for processing and at block 310 retrieves the configuration data that was requested. In one embodiment, the adapter module reads the configuration data from the hard-coded configuration 222. At block 315, the adapter module encapsulates the configuration data into a response message. In one embodiment, the response message is the held URB with the configuration data encapsulated within it. At block 320, the adapter module 212 sends the response message to the driver module 228.

[0022] In one embodiment, the operations depicted in FIG. 2 are carried out whenever the host operating system, or operating system module 210 in the context of the present discussion, polls for configuration of attached hardware devices 108. As discussed above this may be the result of initial system start-up or restart. Additionally, as new hardware devices are attached, the host operating system needs to properly configure itself to communicate with them. In the context of hardware devices configured to communicate with the host operating system using a communications protocol supported by the OS driver module 226, the adapter module is not used. However, when a hardware device 108 is attached that does not communicate using a protocol supported by the OS driver module 226, an adapter module 212 is executed to handle communications between the host operating system and the hardware device 108.

[0023] FIG. 4A is a flowchart of a method according to embodiments of the present invention. In an embodiment the operations depicted in FIG. 4A are carried out by an adapter module 212 as depicted in FIG. 2 and described above. At block 405, a message is received by the adapter module 212 from the OS driver module 226. In one embodiment, the message is an interrupt message. In an alternate embodiment, the message is an interrupt URB message. The adapter module 212 maps the interrupt message to the appropriate hardware device 108 at block 410. At block 415, the adapter module 212 holds the interrupt message for processing.

Through this mechanism, the adapter module provides a mechanism by which the host operating system can query attached hardware devices for input data or for status. One example of such an operation is that of an input device, such as a keyboard. As the host operating system starts up it sends an interrupt message to the keyboard setting up a communications path by which it can receive periodic input data, such as keystrokes. In the context of the present discussion, the host operating system without the adapter module 212 would be unable to communicate with a hardware device 108 that is not configured to communicate using a protocol supported by the OS driver module 226. Using the operations described here, the host operating system enables this communications path using the adapter module 212 as the destination for any messages requesting input data, or interrupt messages.

[0024] FIG. 4B is a flowchart of a method according to embodiments of the present invention. In an embodiment, the operations depicted in FIG. 4B are carried out by an adapter module 212 as depicted in FIG. 2 and described above. FIG. 4B is similar to FIG. 4A with the addition of operations to receive data from a hardware device 108.

[0025] Operations to configure the communications path for a hardware device proceed as depicted above with respect to FIG. 4A. During normal operations following configuration, the adapter module 212 queries the hardware device for input data at block 420 in one embodiment. In one embodiment, querying the hardware device 108 includes periodically polling the hardware device 108. In another embodiment, the hardware device is queried for data following the receipt of the interrupt message at block 405 and the mapping of the message to the hardware device 410. In this embodiment, the host operating system may be sending data items to the hardware device 108 in the context of the hardware device 108 being a mass storage device, such as a hard drive.

[0026] At block 425, the hardware device sends data in response to the query. In one embodiment, the data is input data, such as keystrokes. At block 430, the input data is encapsulated into a response message. In an embodiment, the response message is an interrupt URB response message. At block 435, the adapter module

sends the response message to the OS driver module 226. It will be understood that in the context of input devices or mass storage devices that the operations depicted at blocks 420 through 435 are essentially continuous facilitating the receipt of multiple keystrokes or the storing and retrieval of data on a mass storage device.

[0027] FIG. 5A is an exemplary dataflow diagram carried out on a device according to embodiments of the present invention. For the purposes of illustration the OS driver module 226 is configured to communicate using the USB protocol and the hardware device is configured to communicate using the PS/2 protocol. This is not meant to be limiting in any manner as the dataflow described herein has equal applicability to any hardware device communicating data to a host operating system through an adapter module as contemplated and described above.

[0028] The OS driver module 226 sends 502 a URB to the adapter module 212, the adapter module 212 appearing to the OS driver module 226 as a USB device. The URB is received by the adapter module 212 which maps 504 the URB to the appropriate hardware device requested by the URB and reads 506 the configuration data for that device. The configuration data is stored as hard-coded configuration data as described above with respect to FIG. 2. The configuration data is encapsulated within the URB and that URB is sent back 508 to the driver module. Through these operations the host operating system can request and receive configuration data for a hardware device, even though that hardware device can not communicate directly with the OS driver module 226 using the communications protocol supported by the driver module.

[0029] FIG. 5B is an exemplary dataflow diagram carried out on a device according to embodiments of the present invention. For the purposes of illustration the OS driver module 226 is configured to communicate using the USB protocol and the hardware device is configured to communicate using the PS/2 protocol. This is not meant to be limiting in any manner as the dataflow described herein has equal applicability to any hardware device communicating data to a host operating system through an adapter module as contemplated and described above.

[0030] The driver module generates an interrupt URB and sends 510 the interrupt URB to the adapter module 212. The adapter module maps 512 the interrupt URB to the appropriate device and holds the interrupt URB for further processing and receipt of input data from the appropriate device. The adapter module 212 and more particularly a querying module 513 of the adapter module 212 periodically polls the appropriate device, or more particularly the hardware interface module 214 coupled to the appropriate device. When input data in the form of a PS/2 packet is sent 514 from the hardware interface module 214 in response to the query it is encapsulated 512 into the original interrupt URB. This URB, the original interrupt URB with the encapsulated input data is sent 516 as a URB to the OS driver module 226. The operations and dataflow for input data may be continuous in the context of a keyboard receiving keystrokes from a user.

[0031] Thus, although specific embodiments have been illustrated and described herein, it should be appreciated that any arrangement calculated to achieve the same purpose may be substituted for the specific embodiments shown. This disclosure is intended to cover any and all adaptations or variations of various embodiments of the invention. Combinations of the above embodiments and other embodiments not specifically described herein will be apparent to those of skill in the art upon reviewing the above description.

[0032] The Abstract of the Disclosure is provided to comply with 37 C.F.R. §1.72(b), requiring an abstract that allows the reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. Additionally, in the foregoing Detailed Description, it can be seen that various features are grouped together in a single embodiment for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments of the invention require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus the following claims are hereby incorporated into the Detailed Description, with each

claim standing on its own as a separate preferred embodiment. In the appended claims, the terms "including" and "in which" are used as the plain-English equivalents of the terms "comprising" and "wherein," respectively. Moreover, the terms "first," "second," and "third," etc. are used merely as labels, and are not intended to impose numerical requirements on their objects.

What is claimed is:

- 1 1. Apparatus, comprising:
2 a driver module to generate request for data and to receive responses to the
3 request; and
4 an adapter module separately coupled to the driver module and a hardware
5 interface module, the hardware interface module for coupling to a hardware device,
6 the adapter module to receive requests from the driver module and send responses to
7 the driver module.
- 1 2. The apparatus of claim 1, wherein the adapter module is to receive data from
2 the hardware interface module and encapsulate the data in the response sent to the
3 driver module.
- 1 3. The apparatus of claim 1, wherein the hardware device is an input device.
- 1 4. The apparatus of claim 3, wherein the input device is a PS/2 input device.
- 1 5. The apparatus of claim 1, wherein the hardware device is a mass storage
2 device.
- 1 6. A method, comprising:
2 receiving a configuration request from a driver module;
3 retrieving configuration data related to the configuration request;
4 encapsulating the configuration data into a response message; and
5 sending the response message to the driver module.
- 1 7. The method of claim 6, wherein the configuration data is hard-coded
2 configuration data stored previously.

3 8. The method of claim 6, wherein the driver module is executed as part of an
4 operating system executed on a computing device.

1 9. The method of claim 8, wherein the operating system is a Linux operating
2 system.

3
4 10. A method, comprising:
5 receiving an interrupt message from a driver module;
6 mapping the interrupt message to one or more devices; and
7 holding the interrupt message for processing.

1 11. The method of claim 10, further comprising:
2 querying the one or more devices for input data;
3 receiving input data from the one or more devices;
4 encapsulating the input data into a response message; and
5 sending the response message to the driver module.

1 12. The method of claim 11, wherein querying the one or more devices for input
2 data includes periodically polling the one or more devices.

1 13. The method of claim 11, wherein the input data is received within a PS/2
2 packet.

1 14. The method of claim 11, wherein the one or more devices is an input device.

1 15. The method of claim 11, wherein the one or more devices is a mass storage
2 device.

1 16. A system, comprising:
2 a PCI bus; and

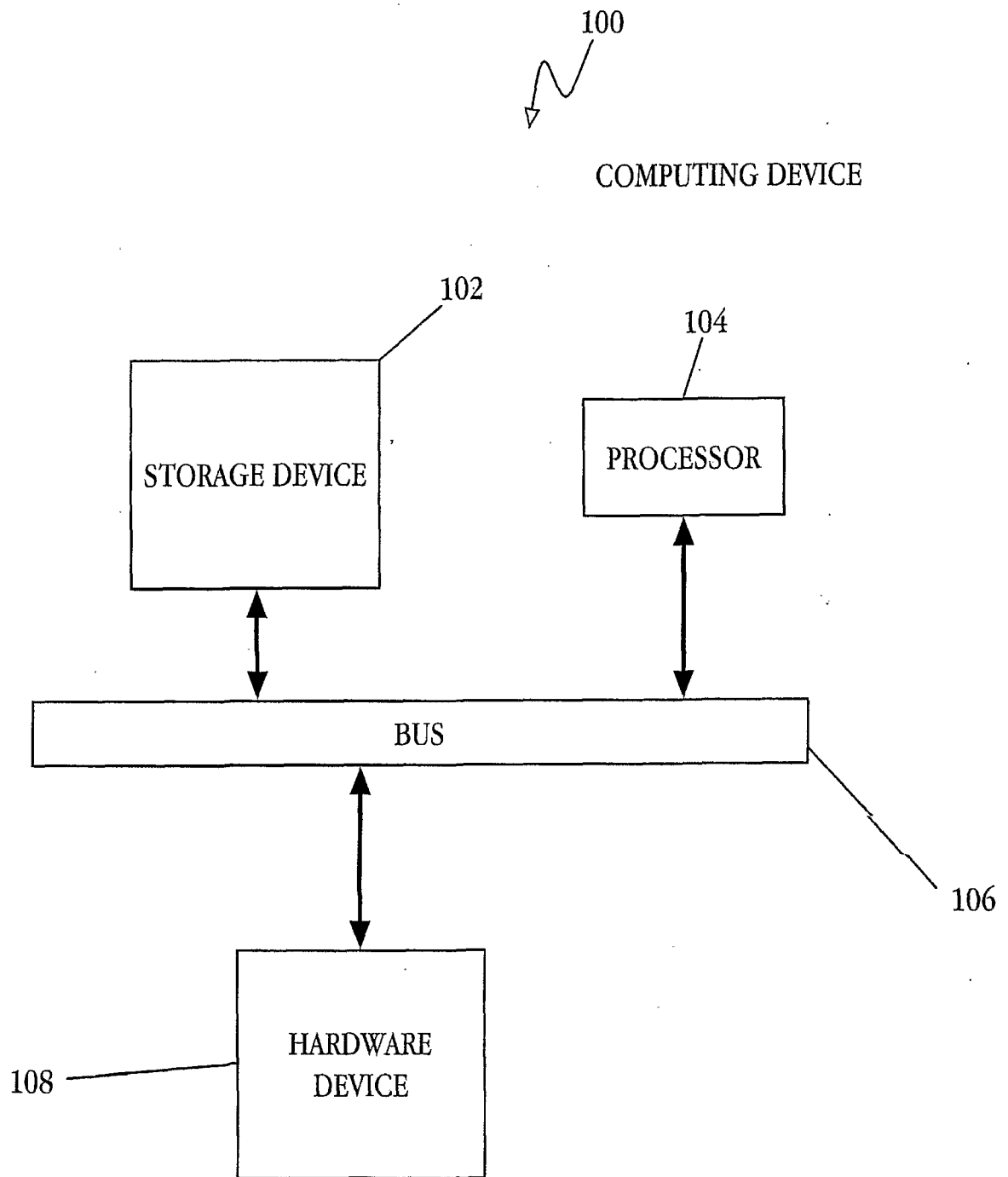
3 a storage device coupled to the PCI bus, the storage device having
4 instructions contained therein which when executed cause to be operated the
5 following software modules:
6 a driver module to generate request for data and to receive responses
7 to the request; and
8 an adapter module separately coupled to the driver module and a
9 hardware interface module, the hardware interface module for coupling to a
10 hardware device, the adapter module to receive requests from the driver module and
11 send responses to the driver module.

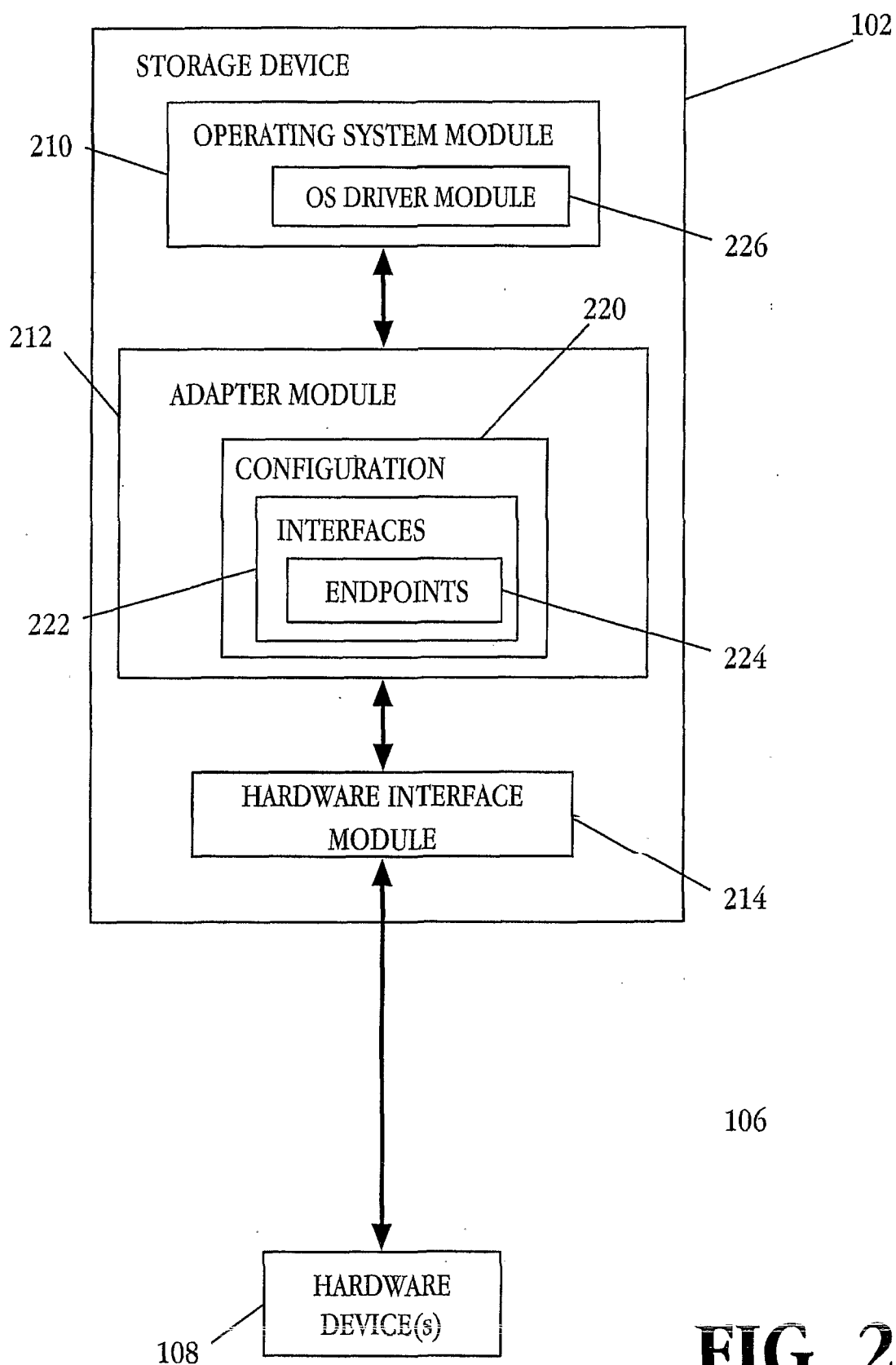
1 17. The system of claim 16, wherein the adapter module includes a hard-coded
2 configuration for the hardware device.

1 18. The system of claim 17, wherein the hard-coded configuration includes one
2 or more hard-coded interfaces

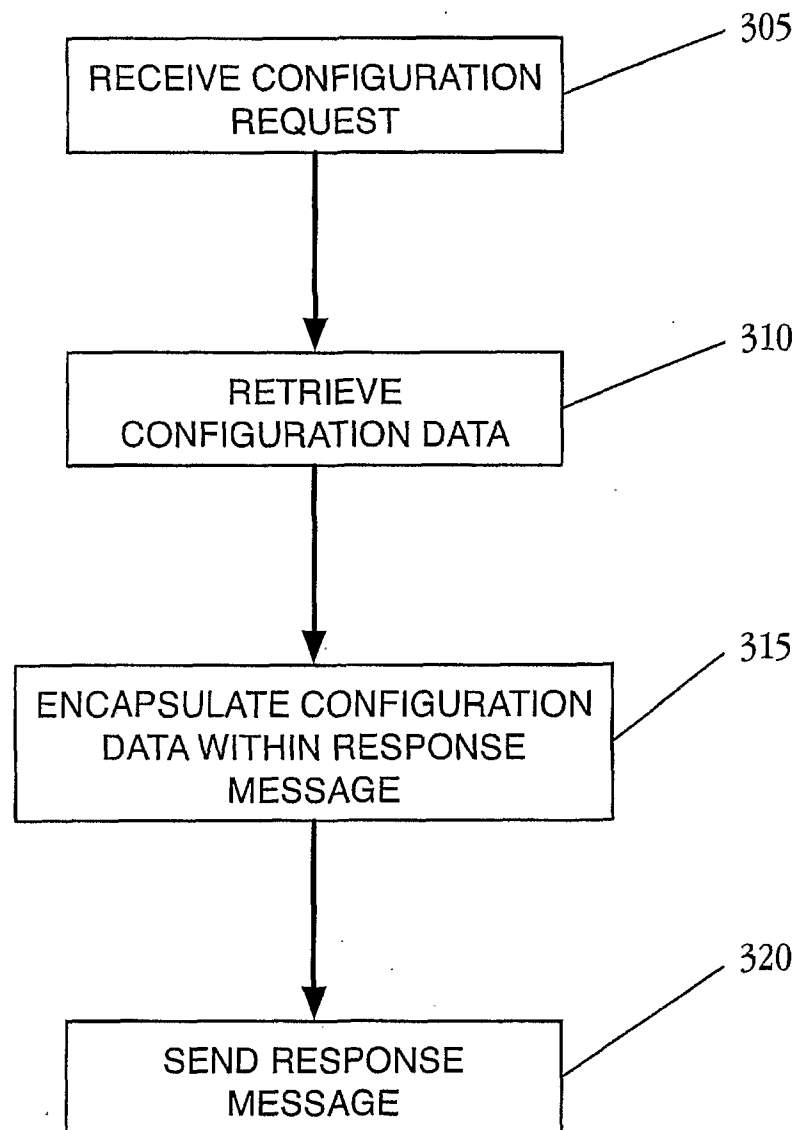
1 19. The system of claim 18, wherein the one or more hard-coded interfaces
2 includes one or more hard-coded endpoints.

1/7

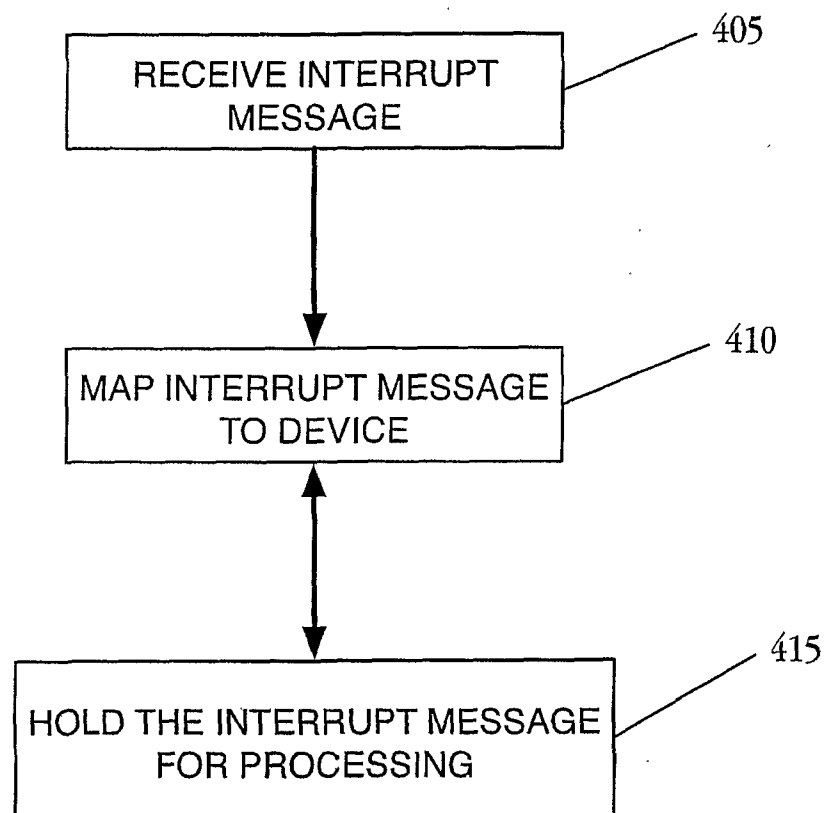
**FIG. 1**

**FIG. 2**

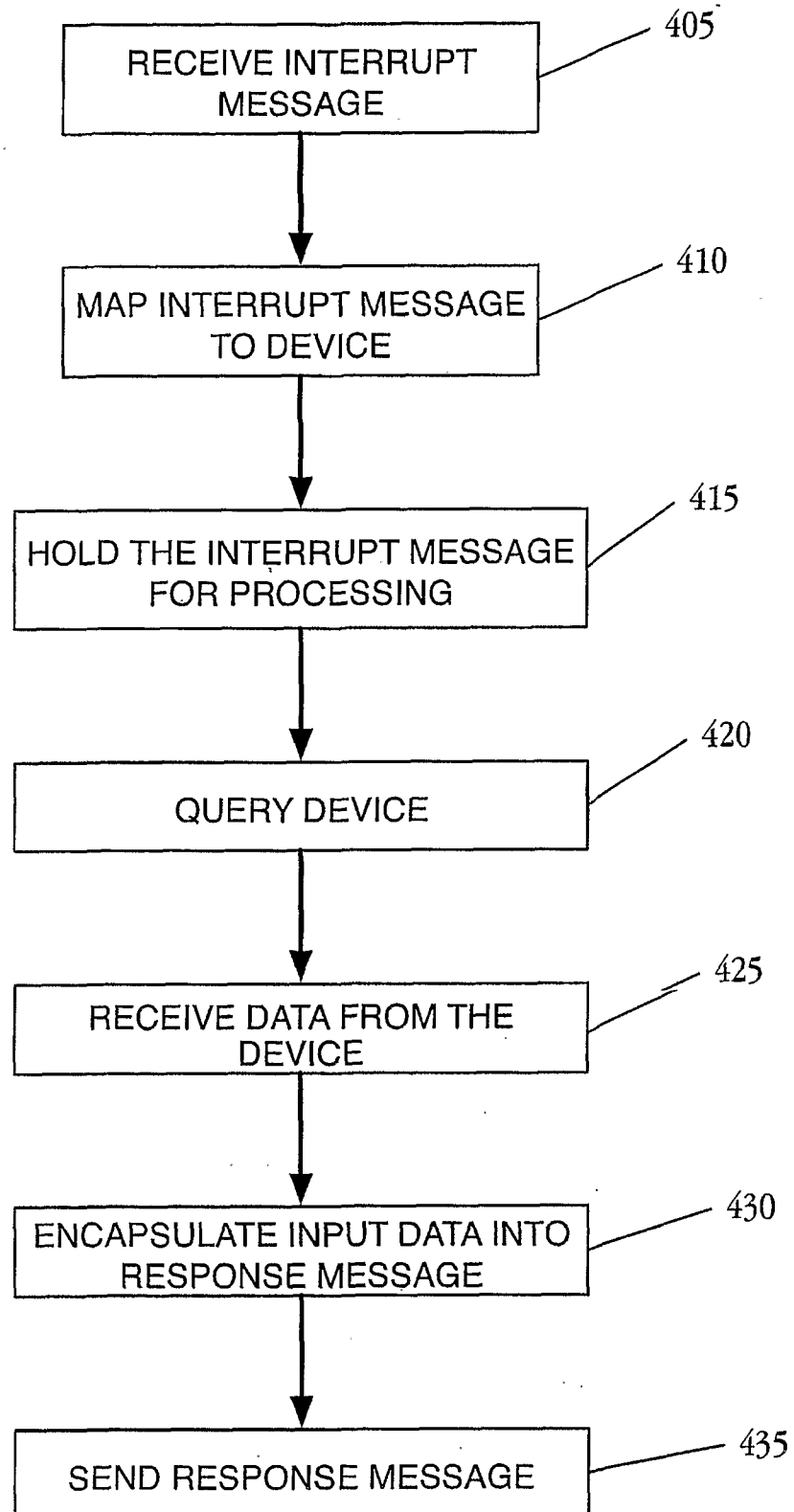
3/7

**FIG. 3**

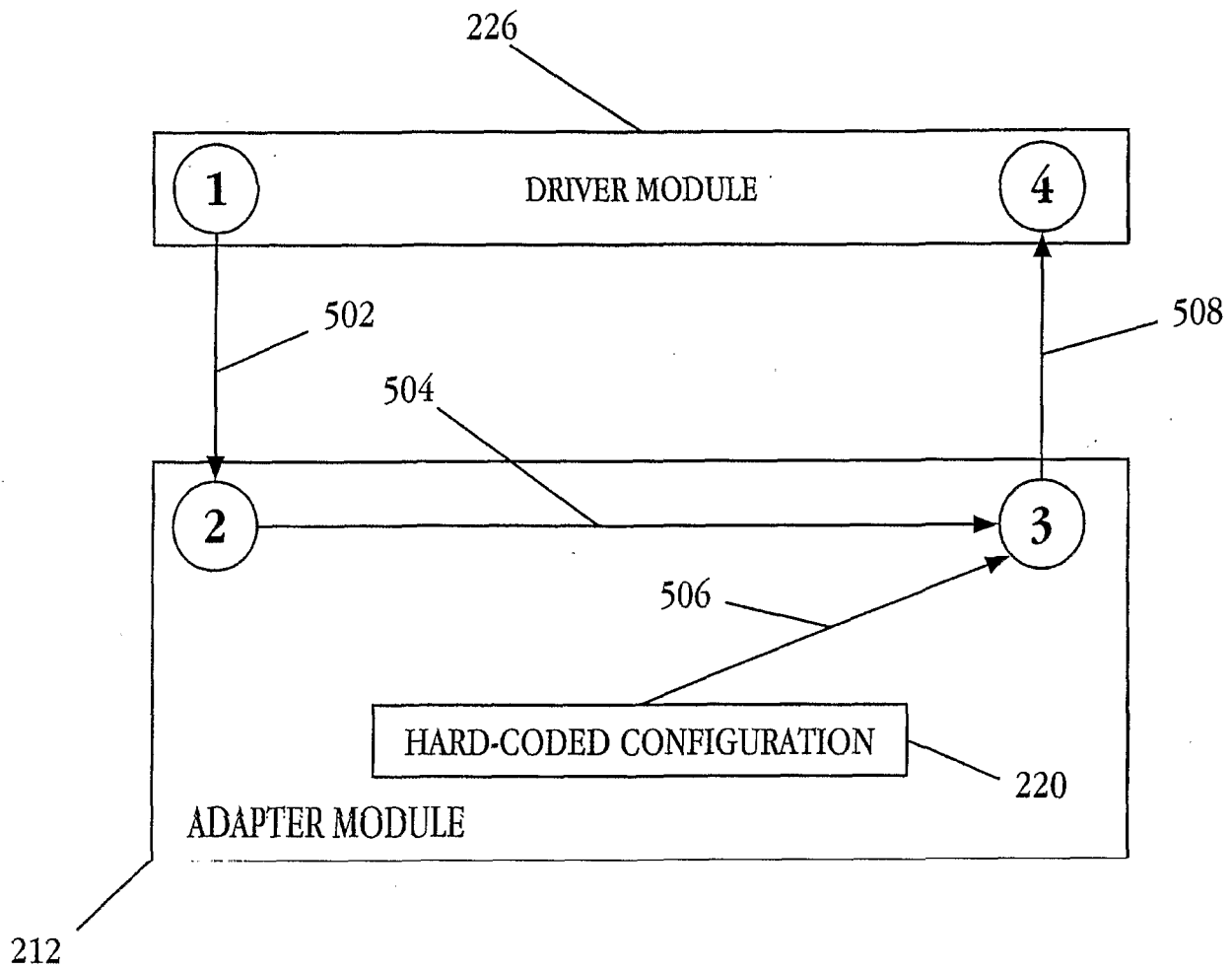
4/7

**FIG. 4A**

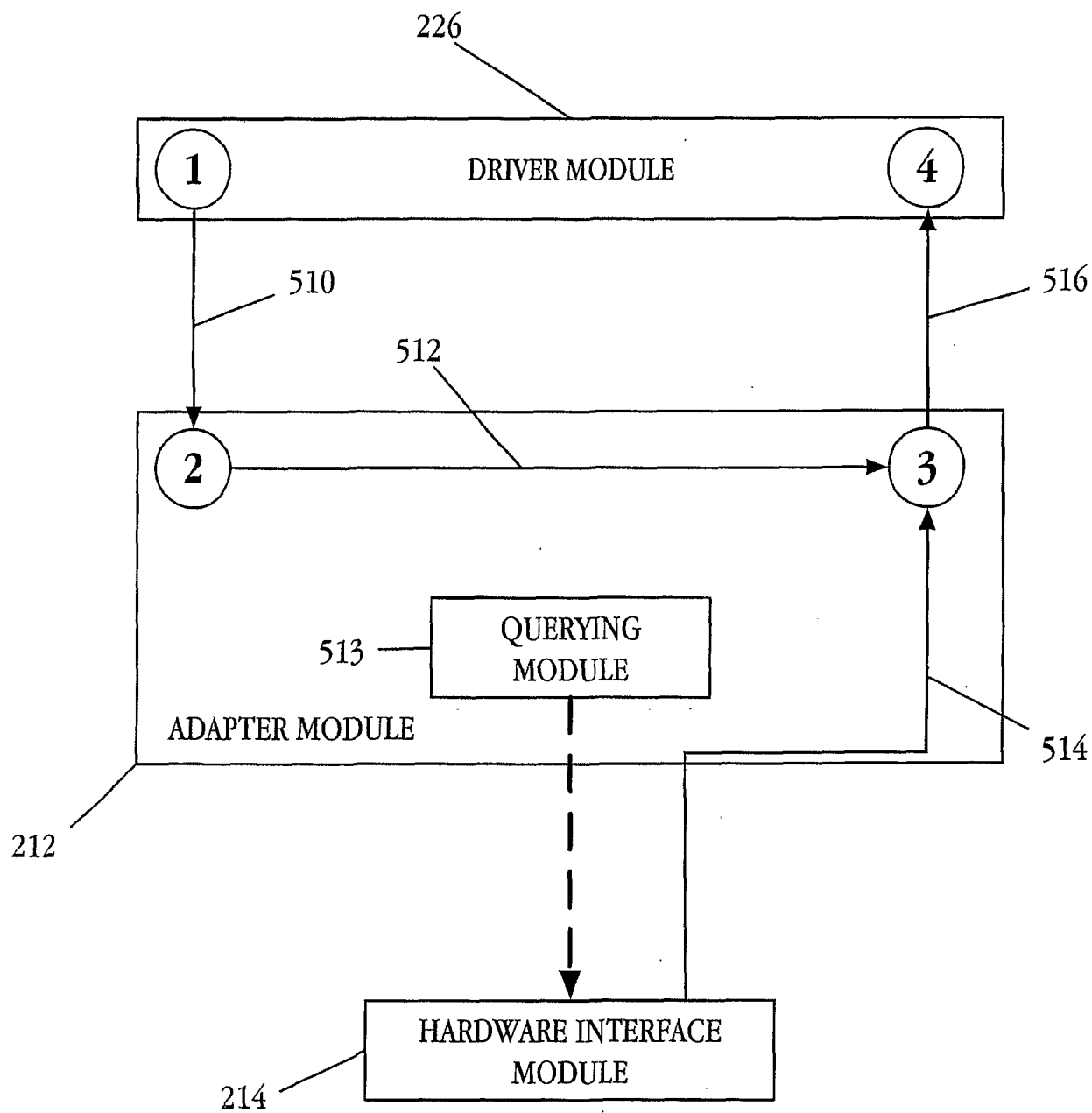
5/7

**FIG. 4B**

6/7

**FIG. 5A**

7/7

**FIG. 5B**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2005/001622

A. CLASSIFICATION OF SUBJECT MATTER

G06F 13/10 (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 13/ (2006.01)

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)

Database:EPODOC,WPI,PAJ,CNPAT

Key: hardware, peripheral, driver, adapter, connectivity, encapsulat+, receiv+, retriev+, send+, map+, hold+,

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US,A1,2004/0230710 (INLINE CONNECTION CORP) 18. Nov 2004 (18. 11. 2004) , see the whole document	1-19
A	US,B1,6324542 (WRIGHT STRATEGIES INC) 27. Nov 2001 (27. 11. 2001) , see the whole document	1-19

☐ 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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"&"document member of the same patent family
"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
22.Feb 2006 (22.02.2006)

Date of mailing of the international search report

Name and mailing address of the ISA/CN
The State Intellectual Property Office, the P.R.China
6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China
100088
Facsimile No. 86-10-62019451

09 · MAR 2006 (09 · 03 · 2006)
Authorized officer
Telephone No. (86-10) 62085025



INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2005/001622

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
US,A1,2004/0230710	18. 11. 2004	WO, A1,2005015412	17. 02. 2005
US,B1,6324542	27. 11. 2001	None	