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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 May 2003 (08.05.2003)

PCT

(10) International Publication Number
WO 03/038646 A2

(51) International Patent Classification[7]: G06F 15/78

(21) International Application Number: PCT/US02/28980

(22) International Filing Date:
12 September 2002 (12.09.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/033,142 1 November 2001 (01.11.2001) US

(71) Applicant: ADVANCED MICRO DEVICES, INC. [US/US]; One AMD Place, Mail Stop 68, P.O. Box 9453, Sunnyvale, CA 94088-3453 (US).

(72) Inventor: GULICK, Dale, E.; 11715 Astoria Drive, Austin, TX 78738 (US).

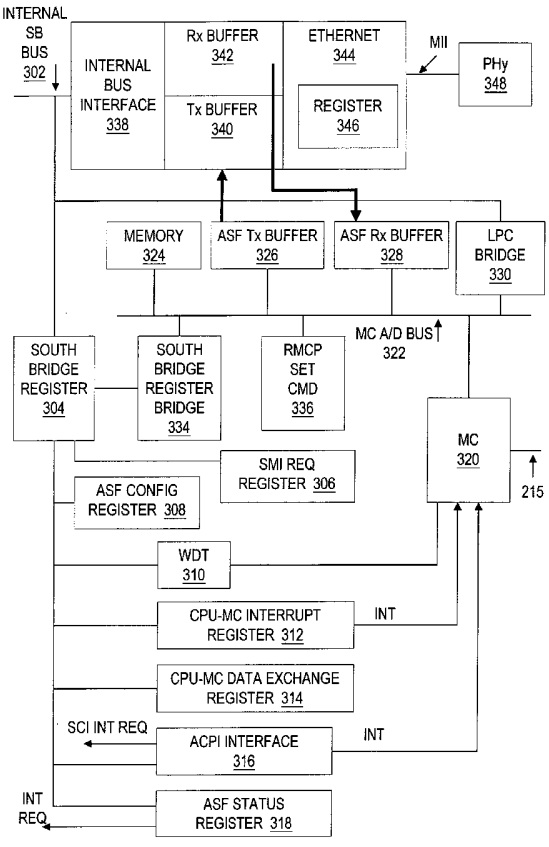
(74) Agent: DRAKE, Paul, S.; Advanced Micro Devices, Inc., 5204 East Ben While Boulevard, Mail Stop 562, Austin, TX 78741 (US).

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

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: MICROCOMPUTER BRIDGE ARCHITECTURE WITH AN EMBEDDED MICROCONTROLLER



(57) Abstract: An integrated circuit, a computer system, and a method of operating the computer system. The integrated circuit includes an internal bus and a microcontroller connected to the internal bus. The microcontroller is configured to master the internal bus. An Ethernet controller may also be coupled to the internal bus with a plurality of buffers coupled between the microcontroller and the Ethernet controller for buffering data between the microcontroller and the Ethernet controller. The Ethernet controller and the microcontroller are then configured to exchange data over the internal bus.

WO 03/038646 A2

Published:

— *without international search report and to be republished upon receipt of that report*

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.

MICROCOMPUTER BRIDGE ARCHITECTURE WITH AN EMBEDDED MICROCONTROLLER

TECHNICAL FIELD

This invention relates generally to computing systems, and, more particularly, to a chipset architecture for remote manageability, such as in a personal computer system.

BACKGROUND ART

Fig. 1A illustrates an exemplary computer system 100. The computer system 100 includes a processor 102, a north bridge 104, memory 106, Advanced Graphics Port (AGP) device 108, a network interface card (NIC) 109, a Peripheral Component Interconnect (PCI) bus 110, a PCI connector 111, a south bridge 112, a battery 113, an AT Attachment (ATA) interface 114 (more commonly known as an Integrated Drive Electronics (IDE) interface), an SMBus 115, a universal serial bus (USB) interface 116, a Low Pin Count (LPC) bus 118, an input/output controller chip (SuperI/O™) 120, and BIOS memory 122. It is noted that the north bridge 104 and the south bridge 112 may include only a single chip or a plurality of chips, leading to the collective term "chipset." It is also noted that other buses, devices, and/or subsystems may be included in the computer system 100 as desired, e.g. caches, modems, parallel or serial interfaces, SCSI interfaces, etc.

The processor 102 is coupled to the north bridge 104. The north bridge 104 provides an interface between the processor 102, the memory 106, the AGP device 108, and the PCI bus 110. The south bridge 112 provides an interface between the PCI bus 110 and the peripherals, devices, and subsystems coupled to the IDE interface 114, the SMBus 115, the USB interface 116, and the LPC bus 118. The battery 113 is shown coupled to the south bridge 112. The Super I/O™ chip 120 is coupled to the LPC bus 118.

The north bridge 104 provides communications access between and/or among the processor 102, memory 106, the AGP device 108, devices coupled to the PCI bus 110, and devices and subsystems coupled to the south bridge 112. Typically, removable peripheral devices are inserted into PCI "slots," shown here as the PCI connector 111, that connect to the PCI bus 110 to couple to the computer system 100. Alternatively, devices located on a motherboard may be directly connected to the PCI bus 110. The SMBus 115 may be "integrated" with the PCI bus 110 by using pins in the PCI connector 111 for a portion of the SMBus 115 connections.

The south bridge 112 provides an interface between the PCI bus 110 and various devices and subsystems, such as a modem, a printer, keyboard, mouse, etc., which are generally coupled to the computer system 100 through the LPC bus 118, or one of its predecessors, such as an X-bus or an Industry Standard Architecture (ISA) bus. The south bridge 112 includes logic used to interface the devices to the rest of computer system 100 through the IDE interface 114, the USB interface 116, and the LPC bus 118. The south bridge 112 also includes the logic to interface with devices through the SMBus 115, an extension of the two-wire inter-IC bus protocol.

Fig. 1B illustrates certain aspects of the south bridge 112, including reserve power by the battery 113, so-called "being inside the RTC (real time clock) battery well" 125. The south bridge 112 includes south bridge (SB) RAM 126 and a clock circuit 128, both inside the RTC battery well 125. The SB RAM 126 includes CMOS RAM 126A and RTC RAM 126B. The RTC RAM 126B includes clock data 129 and checksum data 127. The south bridge 112 also includes, outside the RTC battery well 125, a CPU interface 132, power and system management units 133, and various bus interface logic circuits 134.

Time and date data from the clock circuit 128 are stored as the clock data 129 in the RTC RAM 126B. The checksum data 127 in the RTC RAM 126B may be calculated based on the CMOS RAM 126A data and stored by BIOS during the boot process, such as is described below, e.g. block 148, with respect to Fig. 2. The CPU interface 132 may include interrupt signal controllers and processor signal controllers.

Fig. 1C illustrates a prior art remote management configuration for the computer system 100. A motherboard 101 provides structural and base electrical support for the south bridge 112, the PCI bus 110, the PCI connector 111, the SMBus 115, and sensors 103A and 103B. The NIC 109, a removable add-in card, couples to the motherboard 101, the PCI bus 110, and the SMBus 115 through the PCI connector 111. The NIC 109 includes an Ethernet controller 105 and an ASF microcontroller 107. The Ethernet controller 105 communicates with a remote management server 90, passing management data and commands between the ASF microcontroller 107 and the remote management server 90. The remote management server 90 is external to the computer system 100

An industry standard specification, generally referred to as the Alert Standard Format (ASF) Specification, defines one approach to "system manageability" using the remote management server 90. The ASF Specification defines remote control and alerting interfaces capable of operating when an operating system of a client system, such as the computer system 100, is not functioning. Generally, the remote management server 90 is configured to monitor and control one or more client systems. Typical operations of the ASF alerting interfaces include transmitting alert messages from a client to the remote management server 90, sending remote control commands from the remote management server 90 to the client(s) and responses from the client(s) to the remote management server 90, determining and transmitting to the remote management server 90 the client-specific configurations and assets, and configuring and controlling the client(s) by interacting with the operating system(s) of the client(s). In addition, the remote management server 90 communicates with the ASF NIC 109 and the client(s)' ASF NIC 109 communicates with local client sensors 103 and the local client host processor.

When the client has an ACPI-aware operating system functioning, configuration software for the ASF NIC 109 runs during a "one good boot" to store certain ASF, ACPI, and client configuration data.

The transmission protocol in ASF for sending alerts from the client to the remote management server 90 is the Platform Event Trap (PET). A PET frame consists of a plurality of fields, including GUID (globally unique identifier), sequence number, time, source of PET frame at the client, event type code, event level, sensor device that caused the alert, event data, and ID fields.

Many events may cause an alert to be sent. The events may include temperature value over or under a set-point, voltage value over or under a set-point, fan actual or predicted failure, fan speed over or under a set-point, and physical computer system intrusion. System operation errors may also be alerts, such as memory errors, data device errors, data controller errors, CPU electrical characteristic mis-matches, etc. Alerts may also correspond to BIOS or firmware progression during booting or initialization of any part of the client. Operating system (OS) events may also generate alerts, such as OS boot failure or OS timeouts. The ASF Specification provides for a "heartbeat" alert with a programmable period typically one minute but not to exceed 10 minutes, when the client does not send out the heartbeat, or "I am still here," message.

Client control functions are implemented through a remote management and control protocol (RMCP) that is a user datagram protocol (UDP) based protocol. RMCP is used when the client is not running the

operating system. RMCP packets are exchanged during reset, power-up, and power-down cycles, each having a different message type. The remote management server 90 determines the ASF-RMCP capabilities of the client(s) by a handshake protocol using a presence-ping-request that is acknowledged by the client(s) and followed-up with a presence-pong that indicates the ASF version being used. The remote management server 90 then sends a request to the client to indicate the configuration of the client, which the client acknowledges and follows with a message giving the configuration of the client as stored in non-volatile memory during the "one good boot." The RMCP packets include a contents field, a type field, an offset field, and a value field.

RMCP message transactions involve a request from the remote management server 90, a timed wait for an acknowledgement followed by a second timed wait for a response. If either of the time limits for the acknowledgement or the response is exceeded, then the remote management server 90 knows that either the client needs some of the packets resent or the client has lost contact due to failure of either the client or the communications link.

The ASF NIC 109 must be able to report its IP (Internet protocol) address (or equivalent) without the intervention of the operating system. Thus, the ASF NIC 109 must be able to receive and reply to ARP (Address Resolution Protocol) requests with the operating system, not interfere with ARP packets when the operating system is running, and wake-up for ARP packets when configured to do so. Note that ACPI includes waking-up for ARP packets as a standard configuration.

The following information is sent to the remote management server 90 from the client as an indication of the configuration of the client: an ACPI description table identifying sensors and their characteristics, ASF capabilities and system type for PET messages, and the client's support for RMCP and the last RCMP command; how the client configures an optional operating system boot hang watchdog timer; and the SMBIOS identification of the UUID/GUID for PET messages. ASF objects follow the ASL (ASF Software Language) naming convention of ACPI.

In Fig. 2, a flowchart of a conventional method of initializing a computer system using code stored in the BIOS 122 is shown. During initialization of the power supply, the power supply generates a power good signal to the north bridge 104, in block 136. Upon receiving the power good signal from the power supply, the south bridge 112 (or north bridge 104) stops asserting the reset signal for the processor 102, in block 138.

During initialization, the processor 102 reads a default jump location, in block 140. The default jump location in memory is usually at a location such as FFFF0h. The processor 102 performs a jump to the appropriate BIOS code location (e.g. FFFF0h) in the ROM BIOS 122, copies the BIOS code to the RAM memory 106, and begins processing the BIOS code instructions from the RAM memory 106, in block 142. The BIOS code, processed by the processor 102, performs a power-on self test (POST), in block 144.

The BIOS code next looks for additional BIOS code, such as from a video controller, IDE controller, SCSI controller, etc. and displays a start-up information screen, in block 146. As examples, the video controller BIOS is often found at C000h, while the IDE controller BIOS code is often found at C800h. The BIOS code may perform additional system tests, such as a RAM memory count-up test, and a system inventory, including identifying COM (serial) and LPT (parallel) ports, in block 148. The additional system tests may include ASF, ACPI, and Ethernet initializations, including initiating a communications link with the remote management server 90. The BIOS code also identifies plug-and-play devices and other similar devices and then displays a summary screen of devices identified, in block 150.

The BIOS code identifies the boot location, and the corresponding boot sector, in block 152. The boot location may be on a floppy drive, a hard drive, a CDROM, a remote location, etc. The BIOS code next calls the boot sector code at the boot location to boot the computer system, such as with an operating system, in block 154.

It is noted that for a cold boot or a hard (re)boot, all or most of the descriptions given in blocks 136-154 may occur. During a warm boot or a soft (re)boot the BIOS code usually jumps from block 142 into block 148, skipping the POST, memory tests, etc.

Remote management techniques such as ASF are predicated on the NIC 109 being installed for "one good boot" of the operating system so that initialization of the remote management hardware and/or firmware can be supervised by the operating system. Improvements in remote management for personal computers may speed the initialization of remote management hardware and/or firmware and may lessen the dependence on the operating system. A computer system 100 with a long boot time slows productivity and, at a minimum, irritates users. It would be desirable to shorten boot times if possible.

DISCLOSURE OF INVENTION

In one aspect of the present invention, an integrated circuit is disclosed. The integrated circuit includes an internal bus and a microcontroller connected to the internal bus. The microcontroller is configured to master the internal bus. In other embodiments, the integrated circuit includes an Ethernet controller coupled to the internal bus, and a plurality of buffers coupled between the microcontroller and the Ethernet controller for buffering data between the microcontroller and the Ethernet controller. The Ethernet controller and the microcontroller are configured to exchange data over the internal bus.

In another aspect of the present invention, another integrated circuit is disclosed. The integrated circuit includes an internal bus, a microcontroller connected to the internal bus, and a system management interrupt request register coupled to the internal bus. The microcontroller is configured to master the internal bus. The system management interrupt request register is configured to generate a request for a system management interrupt in response to a system management interrupt vector written to the system management interrupt request register. The microcontroller is configured to write the system management interrupt vector to the system management interrupt request register.

In still another aspect of the present invention, still another integrated circuit is disclosed. The integrated circuit includes an internal bus, a microcontroller connected to the internal bus, and an interrupt register coupled to the internal bus. The microcontroller is configured to master the internal bus. A microcontroller interrupt is generated in response to a microcontroller interrupt vector written to the interrupt register.

In another aspect of the present invention, a computer system is disclosed. The computer system includes an external bus, an integrated circuit, and a processor coupled to the external bus. The integrated circuit includes an internal bus, a microcontroller connected to the internal bus, an Ethernet controller coupled to the internal bus, a plurality of buffers coupled between the microcontroller and the Ethernet controller for buffering data, and a bus interface logic connected to the external bus. The microcontroller is configured to master the internal bus. The Ethernet controller and the microcontroller are configured to exchange data over the internal bus. The processor is configured to communicate over a network using the Ethernet controller.

In yet another aspect of the present invention, a method for operating a computer system is disclosed. The method includes receiving a Alert Standard Format message at a Ethernet controller in an Alert Standard Format south bridge and transmitting the Alert Standard Format message from the Ethernet controller in the Alert Standard Format south bridge to a microcontroller in the Alert Standard Format south bridge over an internal bus in the Alert Standard Format south bridge. When operating the microcontroller in the Alert Standard Format south bridge in an Alert Standard Format slave mode, the method includes transmitting the Alert Standard Format message from the microcontroller in the Alert Standard Format south bridge over an external bus to an the Alert Standard Format network interface card. When operating the microcontroller in the Alert Standard Format south bridge in an Alert Standard Format master mode, the method includes sending an acknowledgement to the Alert Standard Format message from the microcontroller in the Alert Standard Format south bridge to the Ethernet controller in the Alert Standard Format south bridge.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may be understood by reference to the following description taken in conjunction with the accompanying drawings, in which like reference numerals identify similar elements, and in which:

Fig. 1A illustrates a block diagram of a prior art computer system, Fig. 1B illustrates a block diagram of a prior art south bridge, and Fig. 1C illustrates a prior art remote management arrangement;

Fig. 2 illustrates a flowchart of a prior art method for booting a computer system using code stored in ROM;

Figs. 3A and 3B illustrate block diagrams of embodiments of computer systems having remote management arrangements, according to various aspects of the present invention;

Fig. 4 illustrates a block diagram of an embodiment of an ASF south bridge including integrated ASF, ACPI, and/or Ethernet capabilities, according to various aspects of the present invention;

Fig. 5 illustrates a block diagram of an embodiments of the ASF south bridge including ASF registers in the RTC battery well of the ASF south bridge, according to various aspects of the present invention;

Fig. 6 illustrates a flowchart an embodiment of a method for booting a computer system including the ASF south bridge of Fig. 4, according to one aspect of the present invention; and

Figs. 7A and 7B illustrate flowcharts of embodiments of method for operating a computer system including the ASF south bridge of Fig. 4, according to various aspects of the present invention.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof have been shown by way of example in the drawings and are herein described in detail. It should be understood, however, that the description herein of specific embodiments is not intended to limit the invention to the particular forms disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the appended claims.

MODE(S) FOR CARRYING OUT THE INVENTION

Illustrative embodiments of the invention are described below. In the interest of clarity, not all features of an actual implementation are described in this specification. It will, of course, be appreciated that in the development of any such actual embodiment, numerous implementation-specific decisions must be made to achieve the developers' specific goals, such as compliance with system-related and business-related constraints, which will vary from one implementation to another. Moreover, it will be appreciated that such a development

effort might be complex and time-consuming, but would nevertheless be a routine undertaking for those of ordinary skill in the art having the benefit of this disclosure. The use of a letter in association with a reference number is intended to show alternative embodiments or examples of the item to which the reference number is connected.

The following non-patent documents are hereby incorporated by reference in their entirety, without prejudice and without disclaimer, as if set forth fully herein:

[ASF] Alert Standard Format Specification, 1.03, 20 June 2001, DSP0114, and earlier version, at http://www.dmtf.org/spec/asf.html;

[ACPI] *Advanced Configuration and Power Interface Specificatio*n, 2.0, 27 July 2000, and earlier version, http://www.teleport.com/~acpi/spec.htm;

[RFC1157] *A Simple Network Management Protocol*, http://www.ietf.org/rfc/rfc1157.txt;

[CIM] CIM Standards, http://www.dmtf.org/spec/cims.html;

[IPMI] *Intelligent Platform Management Interface Specification v1.0, rev 1.1,* August 26, 1999, and earlier versions, http://developer.intel.com/design/servers/ipmi/;

[RFC1188] *IP and ARP on FDDI Networks*, http://www.ietf.org/rfc/rfc1180.txt;

[FRU] *IPMI Field Replaceable Unit (FRU) Information Storage Definitio*n, v1.0, 16 September 1998, and earlier versions, ftp://download.intel.com/design/servers/ipmi/fru1010.pdf;

[MTLS] *Metolious ACPI/Manageability Specificatio*n, v1.0, 30 April 1999, http://developer.intel.com/ial/metolious/index.htm;

[NDCPM] *Network Device Class Power Management Reference Specificatio*n, v1.0a, 21 November 1997, http://www.microsoft.com/hwdev/specs/PMref/PMnetwork.htm;

[PET] *Platform Event Trap Specificatio*n, v1.0, 7 December 1998, and earlier versions, ftp://download.intel.com/design/servers/ipmi/pet100.pdf;

[SCMIS] *SMBus Control Method Interface Specificatio*n, v1.0, 10 December 1999, and earlier versions, http://www.smbus.org/specs/index.html;

[SMBIOS] *System Management BIOS Reference Specificatio*n, v2.3.1, 16 March 1999, and earlier versions, ftp://download.intel.com/ial/wfm/smbios.pdf;

[SMBUS_2.0] *System Management Bus (SMBus) Specificatio*n, v2.0, 03 August 2000, and earlier versions, http://www.smbus.org/specs/index.html; and

[RFC_UDP] *User Datagram Protocol*, RFC 768, http://www.ietf.org/rfc/rfc0768.txt

Turning now to Figs. 3A and 3B, block diagrams of embodiments of computer systems 200A and 200B having remote management arrangements are shown, according to various aspects of the present invention. In Fig. 3A, an ASF south bridge 212 may include integrated ASF, ACPI, and/or Ethernet capabilities for improved remote manageability.

The computer system 200A of Fig. 3A includes a processor 202, a north bridge 204, memory 206, Advanced Graphics Port (AGP) device 208, a PCI bus 210, a PCI connector 211, the ASF south bridge 212, a battery 213, an AT Attachment (ATA) interface 214, an SMBus 215, a USB interface 216, an LPC bus 218, an input/output controller chip (SuperI/O™) 220, extended BIOS memory 222, and, optionally, a crypto-processor

224 and protected storage 230. It is noted that the north bridge 204 and the ASF south bridge 212 may include only a single chip or a plurality of chips in the "chipset." It is also noted that other buses, devices, and/or subsystems may be included in the computer system 200A as desired, e.g. caches, modems, parallel or serial interfaces, SCSI interfaces, etc.

The processor 202 is coupled to the north bridge 204. The north bridge 204 provides an interface between the processor 202, the memory 206, the AGP device 208, and the PCI bus 210. The ASF south bridge 212 provides an interface between the PCI bus 210 and the peripherals, devices, and subsystems coupled to the IDE interface 214, the SMBus 215, the USB interface 216, and the LPC bus 218. The battery 213 is shown coupled to the ASF south bridge 212. The Super I/O™ chip 220, the extended BIOS 222, and the crypto-processor 224 are coupled to the LPC bus 218. The protected storage 230 is coupled through the crypto-processor 224.

The north bridge 204 provides communications access between and/or among the processor 202, memory 206, the AGP device 208, devices coupled to the PCI bus 210 and devices and subsystems coupled to the ASF south bridge 212. Typically, removable peripheral devices are inserted into PCI "slots," shown here as the PCI connector 211, that connect to the PCI bus 210 to couple to the computer system 200A. Alternatively, devices located on a motherboard may be directly connected to the PCI bus 210. The SMBus 215 is shown "integrated" with the PCI bus 210 by using pins in the PCI connector 211 for a portion of the SMBus 215 connections.

The ASF south bridge 212 provides an interface between the PCI bus 210 and various devices and subsystems, such as a modem, a printer, keyboard, mouse, etc., which are generally coupled to the computer system 200A through the LPC bus 218 (or its predecessors, such as the X-bus or the ISA bus). The ASF south bridge 212 includes logic used to interface the devices to the rest of computer system 200A through the IDE interface 214, the SMBus 215, preferably supporting masters external to the ASF south bridge 212, the USB interface 216, and the LPC bus 218.

It is also noted that the operations of the LPC bus 218 may correspond to the prior art Low Pin Count Interface Specification Revision 1.0 of September 29, 1997. The operations of the LPC bus 218 may also use an extended LPC bus transaction mechanism. By using multiple START signals that decode to an extended memory address, additional memory addresses may be utilized beyond those in the standard LPC address range. The use of multiple START values is allowed with a standard LPC bus transaction, but most of the decode values are not used.

The extended BIOS 222 includes additional memory locations different from or in addition to those memory locations in the BIOS memory 122. The additional memory locations may have specific read/write permissions and/or be secure memory locations. Memory addressing for the extended BIOS 222 may use the extended LPC bus transaction mechanism described above. The crypto-processor 224 may provide security for the protected storage 230 with access to the protected storage 230 being through the crypto-processor 224.

As mentioned above, the ASF south bridge 212 may include integrated ASF, ACPI, and/or Ethernet functionality, according to various aspects of the present invention. As there is no ASF NIC 109 in the computer system 200A, according to one aspect of the present invention, the ASF south bridge 212 recognizes that it must be a master ASF controller for the computer system 200A, during a power-up cycle. The computer system 200A may advantageously boot faster than the computer system 100 by initiating the ASF and/or ACPI

assets in the ASF south bridge 212 during the main portion of the BIOS loading since the ASF, ACPI, and/or Ethernet hardware are known to the BIOS code writer before the BIOS code is written. The BIOS code itself may then be enlarged to include any or all ASF, ACPI, and/or Ethernet initialization data and/or firmware. Additional details of various embodiments of the present invention are given below.

In Fig. 3B, the computer system 200B differs from the computer system 200A in that the computer system 200B includes the ASF NIC 109 at the PCI connector 211. In the computer system 200B, the ASF south bridge 212, according to one aspect of the present invention should recognize that it should be an ASF slave to the ASF NIC 109.

Power management functions may be performed inside a secure execution mode (SEM), including using security hardware integrated into the south bridge. One current standard for power management and configuration is the ACPI Specification. According to the ACPI specification, control methods, a type of instruction, tell the computer system to perform an operation. The ACPI specification does not know how to carry out any of the instructions. The ACPI specification only defines the calls, and the software must be written to carry out the calls in a proscribed manner. The proscribed manner of the ACPI specification is very restrictive. One cannot access some registers in your hardware. To access those registers, one can generate an SMI# (System Management Interrupt) to enter SMM and read these registers. As power management has the potential to be abused e.g. change the processor voltage and frequency, raised above operating limits to destroy the processor, or lowered below operating limits leading to a denial of service, ACPI calls should be carried out in a secure manner, such as inside SEM.

Inside SEM, each ACPI request can be checked against some internal rules for safe behavior. The ACPI request may be placed in an "inbox" (incoming-only memory locations in the south bridge) of a "mailbox" (one-direction-only memory locations in the south bridge), parameter values read from the inbox, the ACPI request evaluated using the inbox parameters for acceptability, and then either fulfill the request or not, based on the evaluation results.

System Management Mode (SMM) is a mode of operation in the computer system that was implemented to conserve power. The SMM was created for the fourth generation x86 processors, and is different from x86 operating mode. As newer x86 generation processors have appeared, the SMM has become relatively transparent to the operating system. That is, computer systems enter and leave the SMM with little or no impact on the operating system.

In Fig. 4, one embodiment of the ASF south bridge 212 is illustrated, according to various aspects of the present invention. As shown, an internal south bridge bus 302 couples a south bridge register 304 with an internal bus interface 338 of an Ethernet controller 344 and an LPC bridge 330. The south bridge register 304 also couples to an SMI request register 306, an ASF configuration register 308, a watchdog timer (WDT) 31, a CPU-MC (microcontroller) interrupt register 312, a CPU-MC data exchange register 314, an ACPI interface 316, an ASF status register 318, and a south bridge register bridge 334. The south bridge register bridge 334 also couples to an MC address/data (A/D) bus 322.

Also coupled to the MC A/D bus 322 are a memory 324, an ASF transmit (Tx) buffer 326, an ASF receive (Rx) buffer 328, the LPC bridge 330, an RMCP set command unit 336, and an embedded microcontroller 320. The MC 320 is also coupled to the WDT 310 and coupled to receive an interrupt (INT) from the CPU-MC interrupt register 312 and the ACPI interface 316. The ACPI interface 316 also generates an

SCI interrupt request. The ASF status register 318 also generates an interrupt request. The embedded Ethernet controller also includes a Rx buffer coupled to the ASF Rx buffer 328, a Tx buffer 340 coupled to the ASF Tx buffer 326, and an Ethernet core 344, including a register 346. The Ethernet core 344 is shown coupled to a PHy 348 through an MII (Machine Independent Interface). The PHy 348 may be external to the ASF south bridge 212.

The MC 320 couples to the SMBus 215, not shown. The MC 320 may use software-drive I/O ports for the SMBus protocol, according to one aspect of the present invention, using so-called "chapter 13 interfaces" of the ACPI Specification, named from their definition given in chapter 13 of the ACPI Specification. In this embodiment and other embodiments, the processor (CPU) 202 can master the SMBus 215. The MC 320 may store assignable addresses in the memory 324, with fixed motherboard-resident legacy sensor addresses store in the BIOS ROM 122 or the extended BIOS 222. When the ASF NIC 109 is present and the ASF south bridge 212 is operating in slave mode, any sensors internal to the ASF south bridge 212 should be visible to the ASF NIC 109.

The embedded Ethernet controller, including the Ethernet core 344, may be configured at boot time from either BIOS code stored in the extended BIOS or by the MC 320 reading values from an EEPROM, not shown, and writing the register 346. It is noted that the register 346 may include a plurality of storage locations or a plurality of registers each with one or more storage locations.

Note that the MC 320 may have some number of general-purpose I/O pins, not shown. The input pins may be used to generate panic interrupts to the MC 320. The output pins may be used to control motherboard functions that are desired when the processor 202 may be "hung" and for ASF slave mode panic generation. The ASF slave mode panic generation may substitute for "pushes" of sensor 103 outputs. The general-purpose I/O inputs may generate an interrupt to the MC 320 or be polled by the MC 320, as desired.

Also note that the MC 320 may be configured to manage, control, monitor, and/or provide other functionality for the ASF south bridge 212 besides ASF. Other functionality may include security, including SEM functionality, system health checking, including ACPI, or other functionality consistent with the teachings herein.

The SMI request register 306 is configured to generate an SMI interrupt when an interrupt vector is written to the SMI request register 306. The interrupt vector is passed to an interrupt controller, not shown.

The memory 324 may include ROM and/or RAM, as desired. The MC 320 may read configuration data from ROM in the memory 324 and shadow the configuration data in RAM in the memory 324. The configuration data may be stored in the extended BIOS 222 and shadowed in the RAM. Note that the ACPI interface 316 couples to the power/system management core 233, shown in Fig. 3, in the ASF south bridge 212.

In one embodiment, the ASF configuration register 308 is a plug and play configuration register for the MC 320 configured for ASF. While ASF is primarily used when the operating system is absent (*e.g.*, not yet loaded at boot time or hung), ASF does interact with the operating system.

In one embodiment, the MC 320 is a conventionally available microcontroller, such as an embedded 8051 microcontroller. The 8051 microcontroller and related microcontrollers have well-known functionality in the art. Typical functionality of the 8051 microcontroller includes a central processing unit with a Boolean processor optimized for one-bit operations, five or six interrupts, with two external and two priority levels, two or three timers or counters, often 16-bit, a programmable full-duplex serial port with data rate defined by one of

the timers, 32 I/O lines often as four 8-bit ports, RAM, and optional ROM. The 8051 microcontroller is known to exist in a multitude of varieties, each variation being embraced herein. Other microcontroller and microprocessor designs are also contemplated as the MC 320.

Fig. 5 illustrates the RTC battery well 225 of the ASF south bridge 212, according to the present invention. In addition to SB RAM 226, divided into CMOS RAM 226A and RTC RAM 226B, the RTC battery well 225 includes a clock circuit 228, a status register 250, and an enable register 252. The RTC RAM 226B includes checksum data 227 and clock data 229. The battery 213 is coupled to provide power to the contents of the RTC battery well 225. The status register 250 is configured to store status information for the ASF capabilities of the computer system 200. The enable register 252 is configured to store a master bit that, when set, indicates that the ASF NIC 109 is not present. A slave bit may alternatively be stored that, when set, indicates that the ASF NIC 109 is present. It is noted that ASF registers 250 and 252 shown in Fig. 5 may each separately include one or more storage locations or a plurality of registers each having one or more storage locations.

The ASF south bridge 212 also includes, outside the RTC battery well 225, a CPU interface 232, power and system management units 233, and various bus interface logic circuits 234. Time and date data from the clock circuit 228 are stored as the clock data 229 in the RTC RAM 226B. The checksum data 227 in the RTC RAM 226B may be calculated based on the CMOS RAM 226A data and stored by the BIOS code during the boot process. The CPU interface 232 may include interrupt signal controllers and processor signal controllers. The power and system management units 233 may include an ACPI (Advanced Configuration and Power Interface) controller.

Fig. 6 illustrates a flowchart of an embodiment of a method of initializing a computer system including the ASF south bridge. Various steps shown in Fig. 2 that are not shown or replaced in Fig. 6 are also contemplated as included in Fig. 6.

During initialization, the processor 202 reads the default jump location. The default jump location in memory is usually at a location such as FFFF0h. The processor 202 performs a jump to the appropriate BIOS code location (e.g. FFFF0h) in the ROM BIOS 222, copies the BIOS code to the RAM memory 206, and begins processing the BIOS code instructions from the RAM memory 206, in block 405. Processing the BIOS code instructions includes checking for the presence of an ASF NIC 109.

If the ASF NIC 109 is present, in decision block 410, then the method continues with block 415. If the ASF NIC 109 is not present, in decision block 410, then the method continues with block 420.

If the ASF NIC 109 is present, then the ASF south bridge 212 is configured as a slave to the ASF NIC 109, in block 415. If the ASF NIC 109 is not present, then the ASF south bridge 212 is configured as a master ASF device, in block 420. Blocks 415 and 420 are each followed by block 425.

The BIOS code, processed by the processor 202, performs a power-on self test (POST), in block 425. The BIOS code next looks for additional BIOS code, such as from a video controller, IDE controller, SCSI controller, etc. and displays a start-up information screen, in block 430. The BIOS code may perform additional system tests, such as a RAM memory count-up test, and a system inventory, including identifying COM (serial) and LPT (parallel) ports, in block 435. The BIOS code also identifies plug-and-play devices and other similar devices and then displays a summary screen of devices identified, in block 440. The BIOS code identifies the boot location, and the corresponding boot sector, in block 445.

Configuring the ASF south bridge 212 as a slave to the ASF NIC 109, in block 415, may include setting a bit indicating the slave condition in the ASF enable register 252. Configuring the ASF south bridge 212 as the ASF master, in block 420, may include setting a bit indicating the master condition in the ASF enable register 252.

Fig. 7A illustrates a flowchart of an embodiment of a method 500 for operating a computer system including the ASF south bridge 212 in slave mode, according to one aspect of the present invention. In slave mode, the ASF south bridge 212 responds to reads of internal sensor status by the ASF NIC 109, in block 505. The ASF south bridge 212 in slave mode responds to SMBus 215 polls originating on the ASF NIC 109, in block 510. The ASF south bridge 212 in slave mode also provides control points for the ASF NIC 109, allowing the ASF NIC 109 to reset the computer system 200 and cycle the power to the computer system 200.

Fig. 7B illustrates a flowchart of an embodiment of a method 600 for operating a computer system including the ASF south bridge 212 in master mode, according to one aspect of the present invention. In master mode, the ASF south bridge 212 actively polls external sensors coupled to the SMBus 215 at a programmable polling rate, in block 605. The ASF south bridge 212 in master mode actively polls or otherwise monitors internal sensor states, in block 610. The ASF south bridge 212 in master mode may generate interrupts and/or respond to interrupts, in block 615. Resulting external sensor status values are combined with internally monitored sensor values and reported to the remote management server 90 via the Ethernet core 344 in the ASF south bridge 212, in block 620.

For the purposes of this disclosure, references to ROM are to be construed as also applying to flash memory and other substantially non-volatile memory types. Note that while the methods of the present invention disclosed herein have been illustrated as flowcharts, various elements of the flowcharts may be omitted or performed in different order in various embodiments. Note also that the methods of the present invention disclosed herein admit to variations in implementation.

Some aspects of the invention as disclosed above may be implemented in hardware or software. Thus, some portions of the detailed descriptions herein are consequently presented in terms of a hardware implemented process and some portions of the detailed descriptions herein are consequently presented in terms of a software-implemented process involving symbolic representations of operations on data bits within a memory of a computing system or computing device. These descriptions and representations are the means used by those in the art to convey most effectively the substance of their work to others skilled in the art using both hardware and software. The process and operation of both require physical manipulations of physical quantities. In software, usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantifies. Unless specifically stated or otherwise as may be apparent, throughout the present disclosure, these descriptions refer to the action and processes of an electronic device, that manipulates and transforms data represented as physical (electronic, magnetic, or optical) quantities within some electronic device's storage into other data similarly represented as physical quantities within the storage, or in transmission or display devices. Exemplary of the terms denoting

such a description are, without limitation, the terms "processing," "computing," "calculating," "determining," "displaying," and the like.

Note also that the software-implemented aspects of the invention are typically encoded on some form of program storage medium or implemented over some type of transmission medium. The program storage medium may be magnetic (*e.g.*, a floppy disk or a hard drive) or optical (*e.g.*, a compact disk read only memory, or "CD ROM"), and may be read only or random access. Similarly, the transmission medium may be twisted wire pairs, coaxial cable, optical fiber, or some other suitable transmission medium known to the art. The invention is not limited by these aspects of any given implementation.

The particular embodiments disclosed above are illustrative only, as the invention may be modified and practiced in different but equivalent manners apparent to those skilled in the art having the benefit of the teachings herein. Furthermore, no limitations are intended to the details of construction or design herein shown, other than as described in the claims below. It is therefore evident that the particular embodiments disclosed above may be altered or modified and all such variations are considered within the scope and spirit of the invention. Accordingly, the protection sought herein is as set forth in the claims below.

CLAIMS

1. An integrated circuit, comprising:

an internal bus;

a first bus interface logic for coupling to a first external bus;

a second bus interface logic for coupling to a second external bus; and

a microcontroller connected to the internal bus, wherein the microcontroller is configured to master the internal bus.

2. The integrated circuit of claim 1, further comprising:

an Ethernet controller coupled to the internal bus, wherein the Ethernet controller and the microcontroller are configured to exchange data over the internal bus; and

a plurality of buffers coupled between the microcontroller and the Ethernet controller for buffering the data.

3. The integrated circuit of claim 1 or 2, wherein the microcontroller is configured as an Alert Standard Format master, and wherein the Ethernet controller is configured to route Alert Standard Format messages to the microcontroller.

4. The integrated circuit of claim 2, wherein the microcontroller is configured as an Alert Standard Format slave, and wherein the Ethernet controller is configured to route Alert Standard Format messages to an external Alert Standard Format master.

5. The integrated circuit of any of claims 1, 2 or 4, further comprising:

a remote management and control protocol set command unit connected to the internal bus, wherein the remote management and control protocol set command unit is configured to execute remote management and control protocol commands received from an external management server through the Ethernet controller.

6. The integrated circuit of any of claims 1-5, further comprising:

a system management interrupt request register coupled to the internal bus, wherein the system management interrupt request register is configured to generate a request for a system management interrupt in response to a system management interrupt vector written to the system management interrupt request register, wherein the microcontroller is configured to write the system management interrupt vector to the system management interrupt request register.

7. The integrated circuit of claim 6, further comprising:

a timer coupled to the internal bus, wherein the microcontroller is configurable to write the system management interrupt vector to the system management interrupt request register in response to an expiration of the timer.

8. The integrated circuit of any of claims 1-6, further comprising:

an interrupt register coupled to the internal bus, wherein a microcontroller interrupt is generated in response to a microcontroller interrupt vector written to the interrupt register.

9. The integrated circuit of any of claims 1-8, further comprising:

a data exchange register coupled to the internal bus, wherein the data exchange register is configured to store data from the microcontroller or for the microcontroller, wherein the microcontroller is configured to write data to the data exchange register, and wherein the microcontroller is configured to read data from the data exchange register.

10. The integrated circuit of any of claims 8 or 9, further comprising:

a system management interrupt request register coupled to the internal bus, wherein the system management interrupt request register is configured to generate a request for a system management interrupt in response to a system management interrupt vector written to the system management interrupt request register, wherein the microcontroller is configured to write the system management interrupt vector to the system management interrupt request register.

11. The integrated circuit of any of claims 2 and 4, wherein the integrated circuit is incorporated into a computer system, the computer system further comprising an external bus, wherein the integrated circuit is coupled to the external bus.

12. The integrated circuit of claim 11, wherein the computer system further comprises a processor coupled to the external bus, wherein the processor is configured to communicate over a network using the Ethernet controller in the integrated circuit.

13. The integrated circuit of any of claims 11 or 12, wherein the computer system further comprises one or more sensors coupled to the external bus, wherein the Ethernet controller is configured to transmit data from the sensors over a network.

14. The integrated circuit of any of claims 11-13, wherein the microcontroller is configured as an Alert Standard Format master, and wherein the Ethernet controller is configured to route Alert Standard Format messages to the microcontroller.

15. The integrated circuit of any of claims 11-13, wherein the computer system further comprises a network interface card coupled to the integrated circuit and to the processor, wherein the network interface card is configured as an Alert Standard Format master, wherein the microcontroller is configured as an Alert Standard Format slave, and wherein the Ethernet controller is configured to route Alert Standard Format messages to the network interface card.

16. A method for operating a computer system, the method comprising:

receiving a Alert Standard Format message at a Ethernet controller in an Alert Standard Format south bridge;

transmitting the Alert Standard Format message from the Ethernet controller in the Alert Standard Format south bridge to a microcontroller in the Alert Standard Format south bridge over an internal bus in the Alert Standard Format south bridge;

when operating the microcontroller in the Alert Standard Format south bridge in an Alert Standard Format slave mode, transmitting the Alert Standard Format message from the microcontroller in the Alert Standard Format south bridge over an external bus to an the Alert Standard Format network interface card; and

when operating the microcontroller in the Alert Standard Format south bridge in an Alert Standard Format master mode, sending an acknowledgement to the Alert Standard Format message from the microcontroller in the Alert Standard Format south bridge to the Ethernet controller in the Alert Standard Format south bridge.

17. The method of claim 16, further comprising:

reading data from one or more of a plurality of Alert Standard Format south bridge registers in response to the Alert Standard Format message;

when operating the microcontroller in the Alert Standard Format south bridge in the Alert Standard Format slave mode, then transferring the data from the one or more of the plurality of Alert Standard Format south bridge registers from the network interface card to the microcontroller in the Alert Standard Format south bridge over the external bus; and

transferring the data from the one or more of the plurality of Alert Standard Format south bridge registers from the microcontroller in the Alert Standard Format south bridge to the Ethernet controller in the Alert Standard Format south bridge over the internal bridge.

18. The method of claim 17, wherein operating the microcontroller in the Alert Standard Format south bridge in the Alert Standard Format master mode comprises microcontroller in the Alert Standard Format south bridge polling Alert Standard Format sensors in the computer system for Alert Standard Format sensor status values and responding to requests from an external management server for the Alert Standard Format sensor status values.

19. The method of claim 17, wherein operating the microcontroller in the Alert Standard Format south bridge in the Alert Standard Format slave mode comprises responding to Alert Standard Format requests from the Alert Standard Format network interface card by the microcontroller in the Alert Standard Format south bridge.

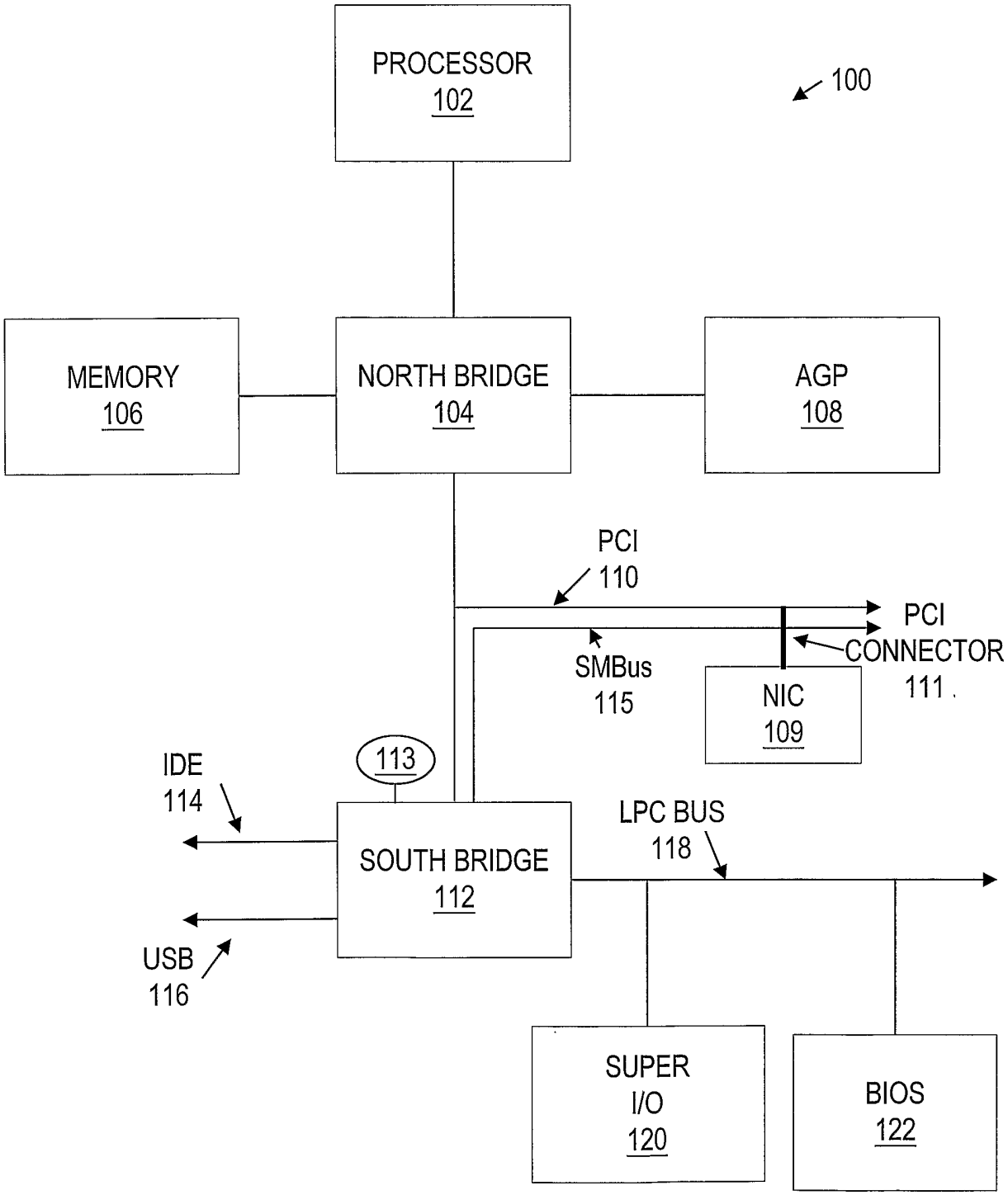


Fig. 1A
(Prior Art)

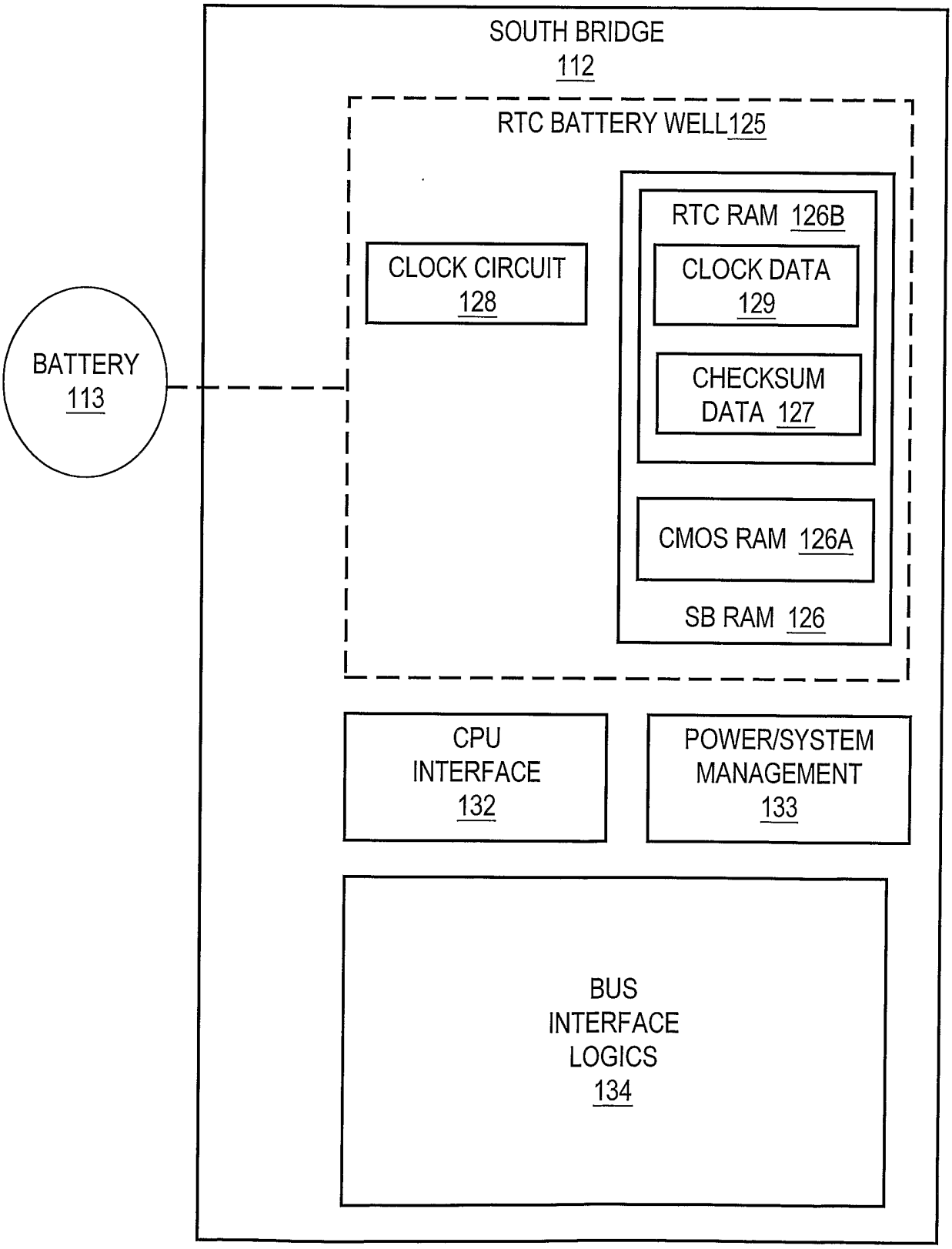


Fig. 1B
(Prior Art)

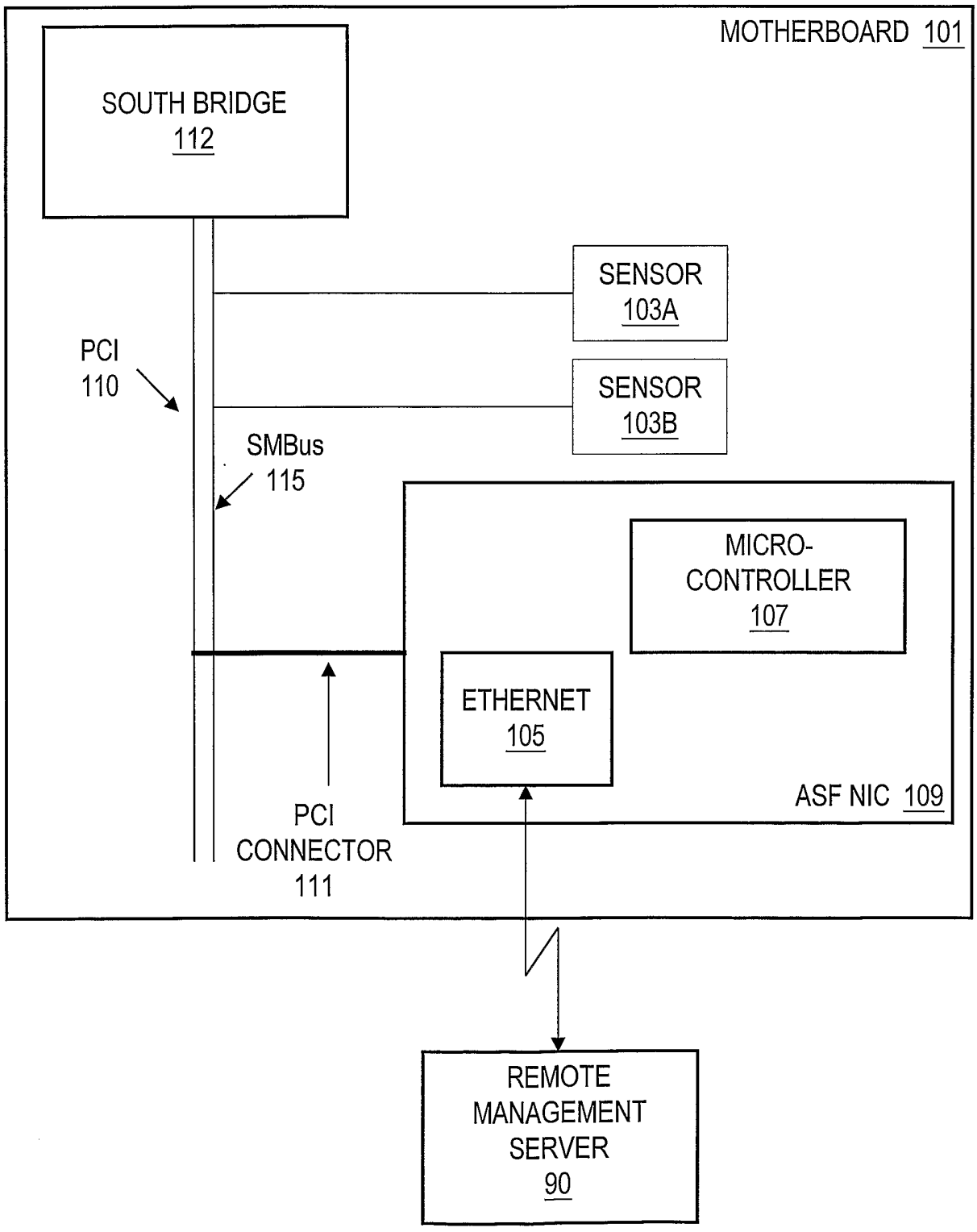


Fig. 1C
(Prior Art)

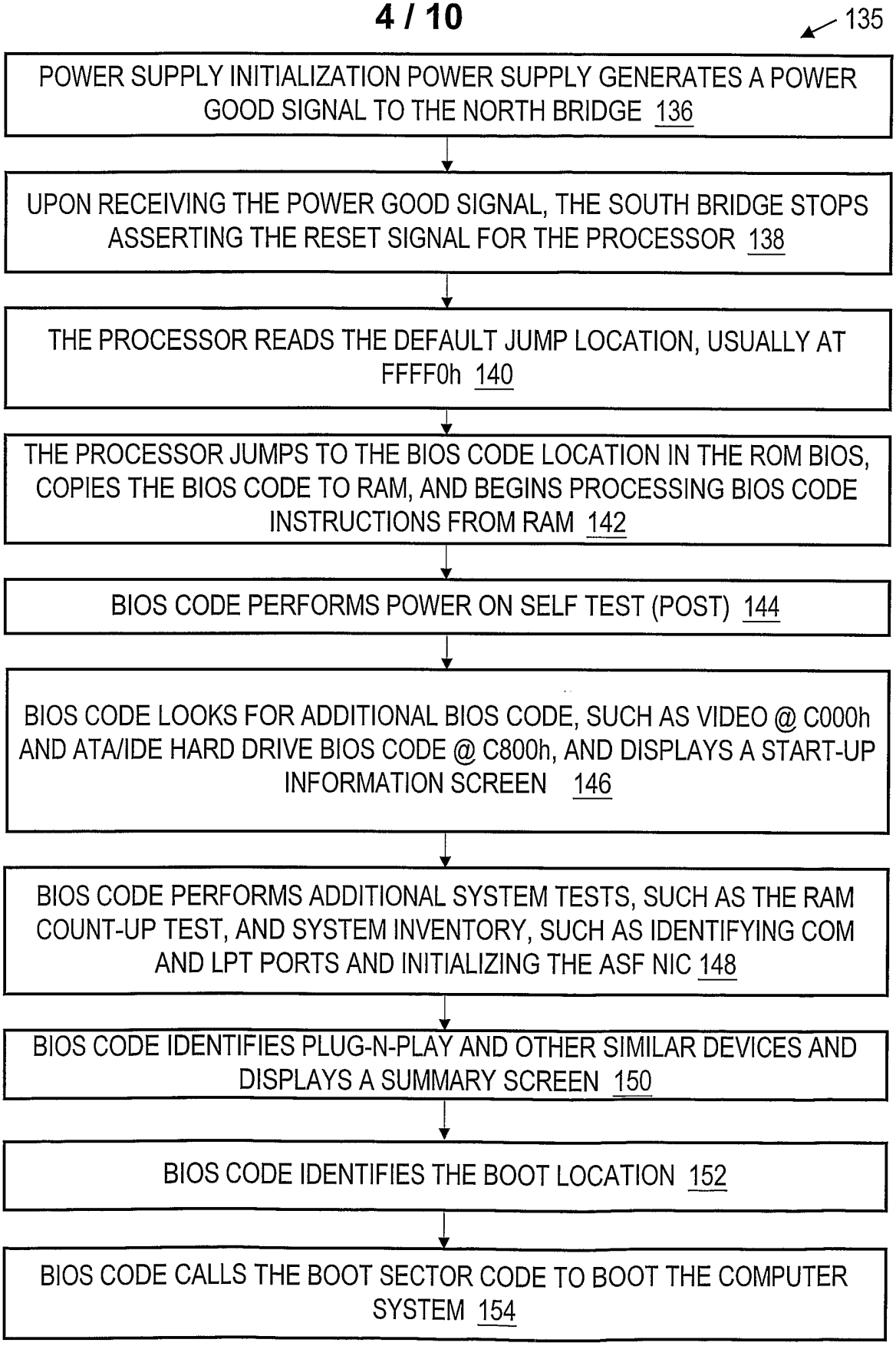


Fig. 2
(Prior Art)

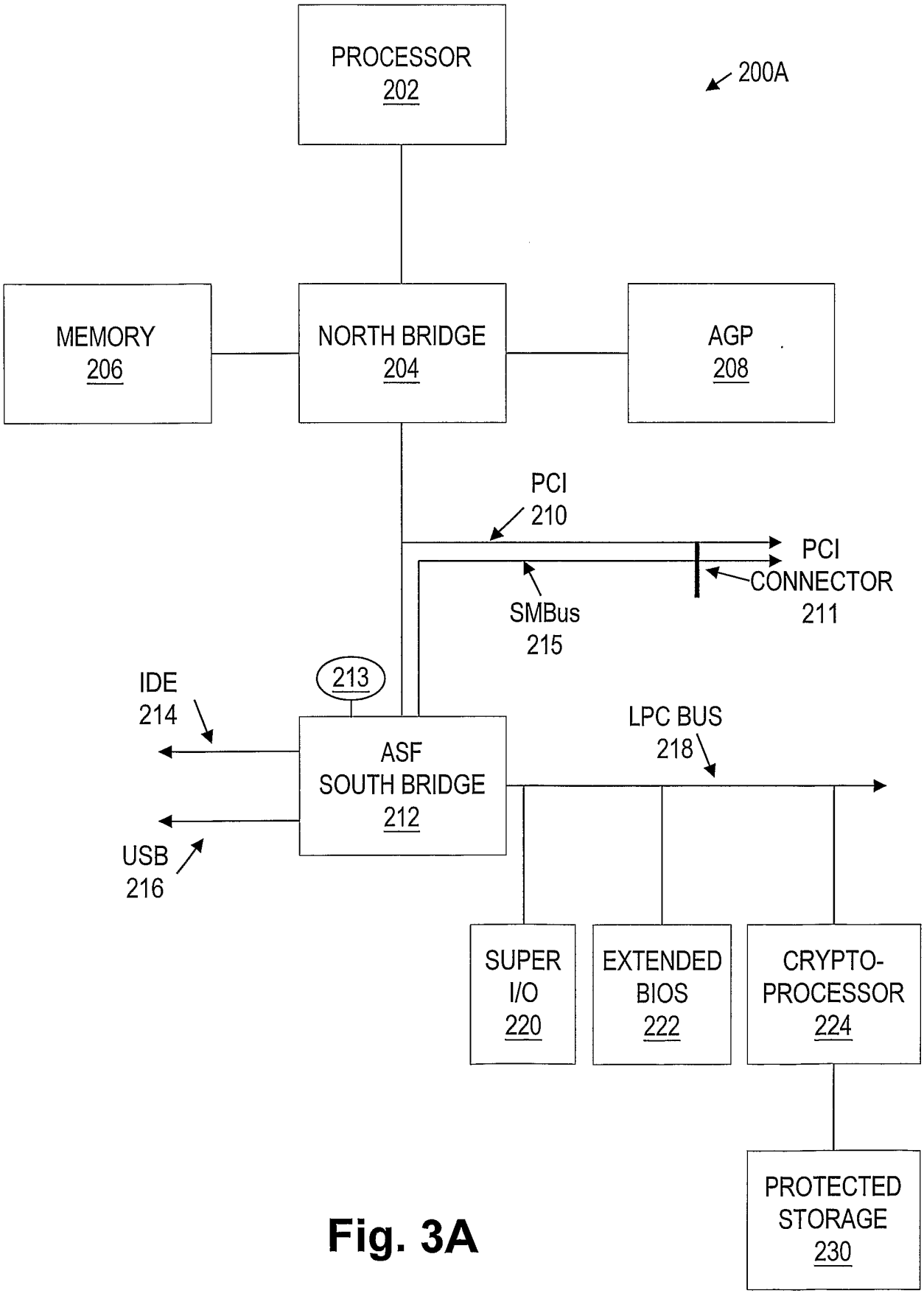

PROCESSOR
202
200A
MEMORY
206
NORTH BRIDGE
204
AGP
208
PCI
210
PCI
CONNECTOR
211
SMBus
215
213
IDE
214
ASF
SOUTH BRIDGE
212
LPC BUS
218
USB
216
SUPER
I/O
220
EXTENDED
BIOS
222
CRYPTO-
PROCESSOR
224
PROTECTED
STORAGE
230

Fig. 3A

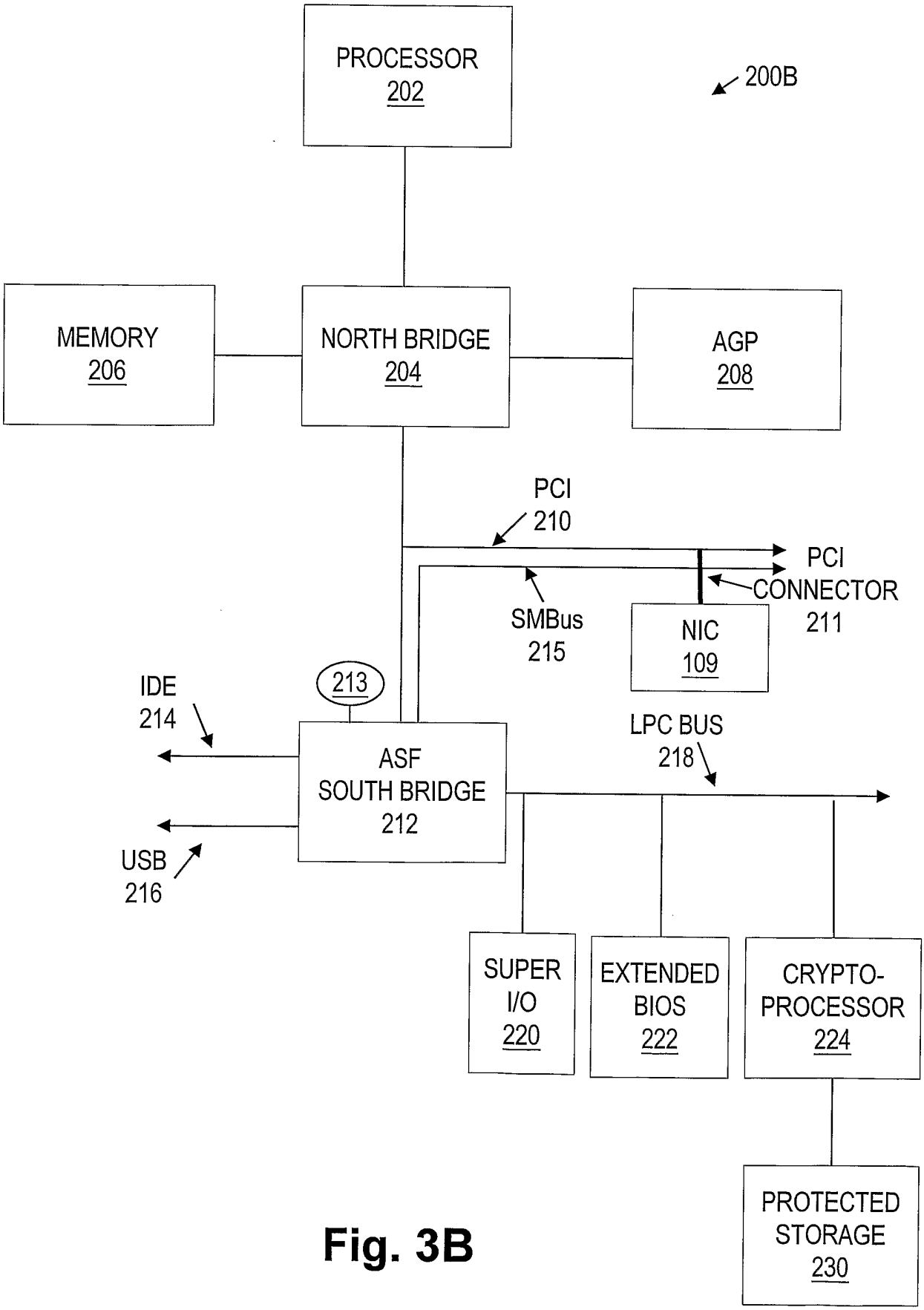

PROCESSOR
202
200B
MEMORY
206
NORTH BRIDGE
204
AGP
208
PCI
210
PCI
CONNECTOR
211
SMBus
215
NIC
109
213
IDE
214
ASF
SOUTH BRIDGE
212
LPC BUS
218
USB
216
SUPER
I/O
220
EXTENDED
BIOS
222
CRYPTO-
PROCESSOR
224
PROTECTED
STORAGE
230

Fig. 3B

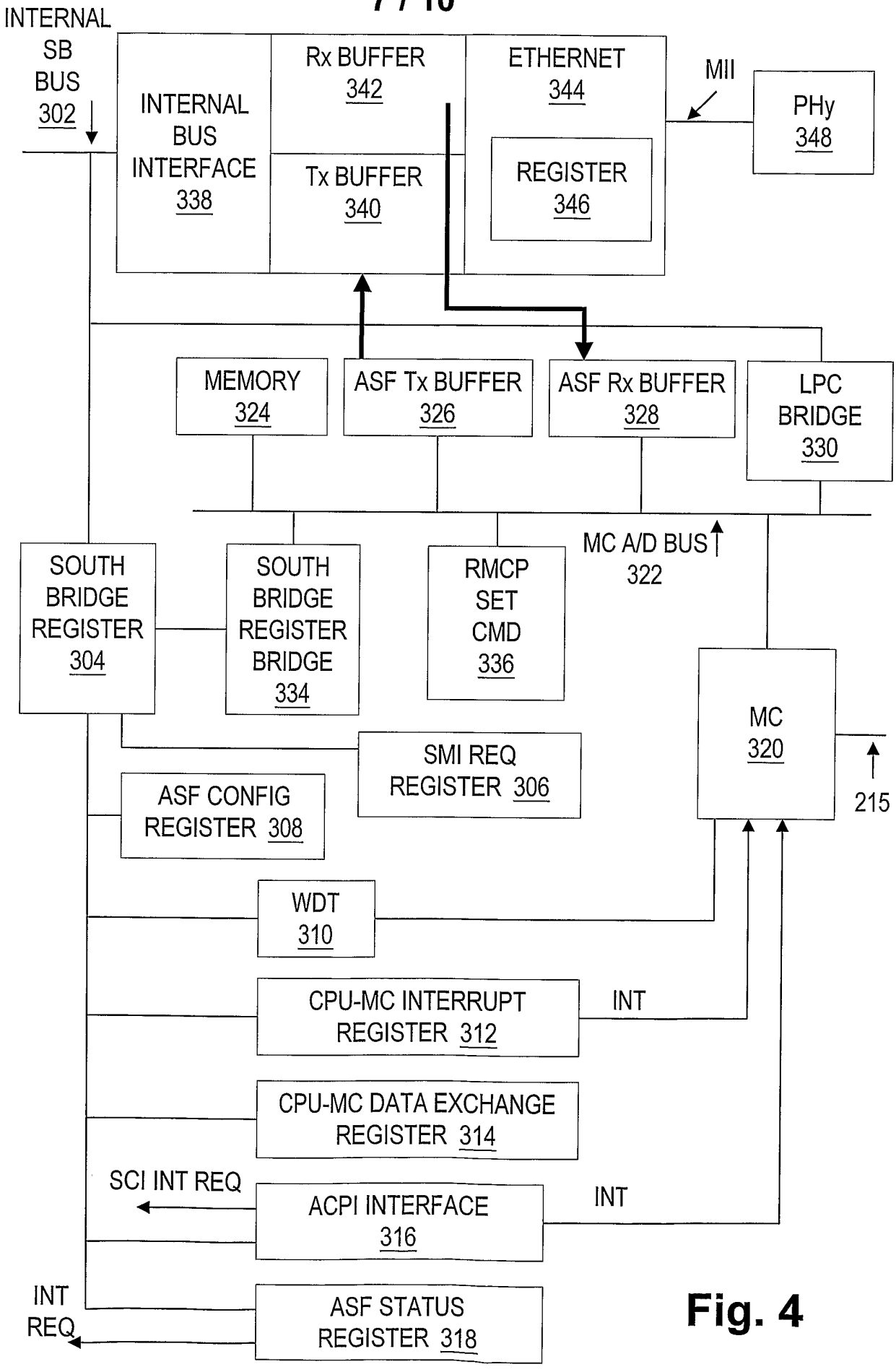

INTERNAL SB BUS 302
INTERNAL BUS INTERFACE 338
Rx BUFFER 342
Tx BUFFER 340
ETHERNET 344
REGISTER 346
MII
PHy 348
MEMORY 324
ASF Tx BUFFER 326
ASF Rx BUFFER 328
LPC BRIDGE 330
MC A/D BUS 322
SOUTH BRIDGE REGISTER 304
SOUTH BRIDGE REGISTER BRIDGE 334
RMCP SET CMD 336
MC 320
215
SMI REQ REGISTER 306
ASF CONFIG REGISTER 308
WDT 310
CPU-MC INTERRUPT REGISTER 312
INT
CPU-MC DATA EXCHANGE REGISTER 314
SCI INT REQ
ACPI INTERFACE 316
INT
INT REQ
ASF STATUS REGISTER 318

Fig. 4

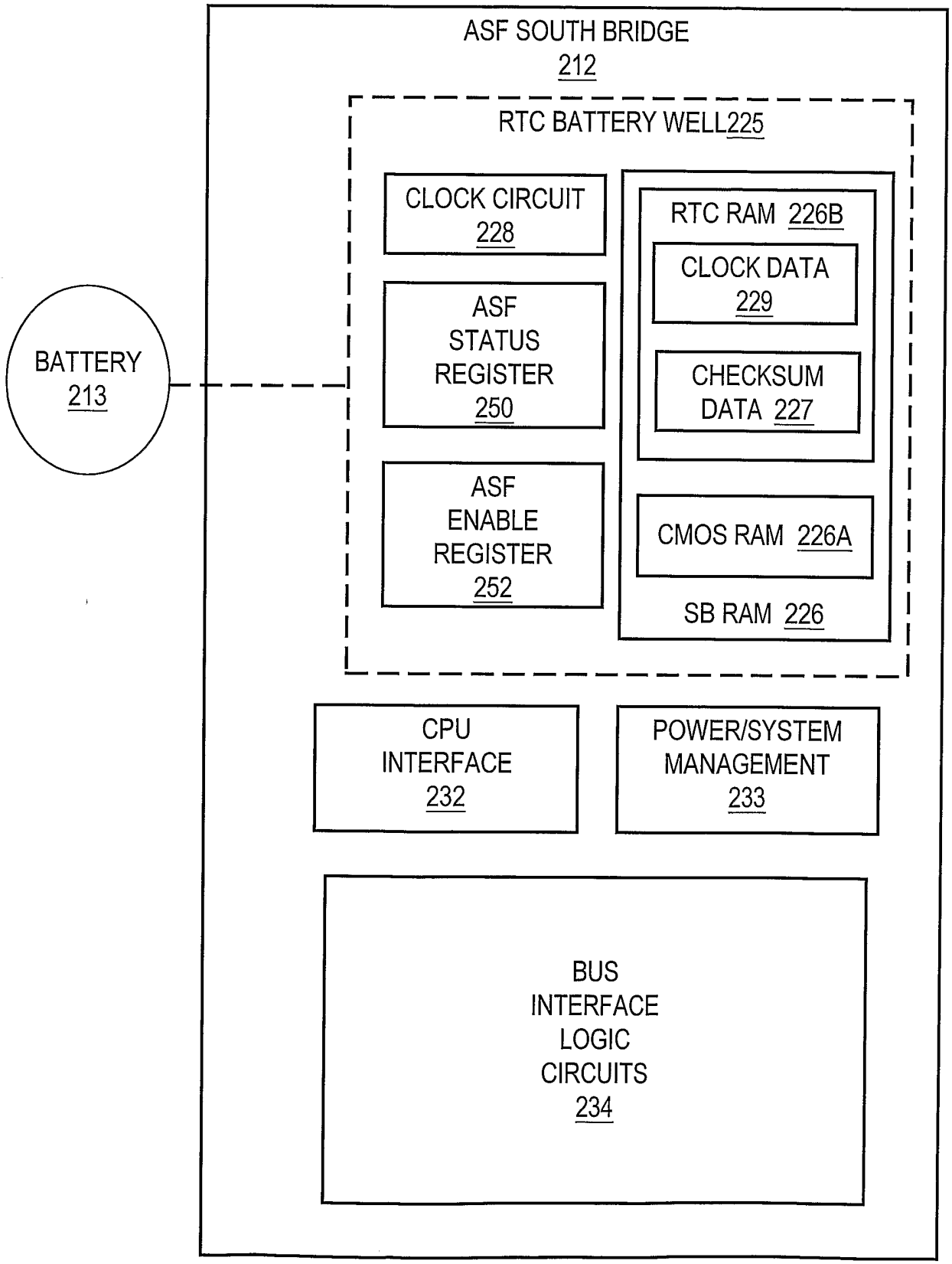

ASF SOUTH BRIDGE
212
RTC BATTERY WELL225
CLOCK CIRCUIT
228
ASF
STATUS
REGISTER
250
ASF
ENABLE
REGISTER
252
RTC RAM 226B
CLOCK DATA
229
CHECKSUM
DATA 227
CMOS RAM 226A
SB RAM 226
BATTERY
213
CPU
INTERFACE
232
POWER/SYSTEM
MANAGEMENT
233
BUS
INTERFACE
LOGIC
CIRCUITS
234

Fig. 5

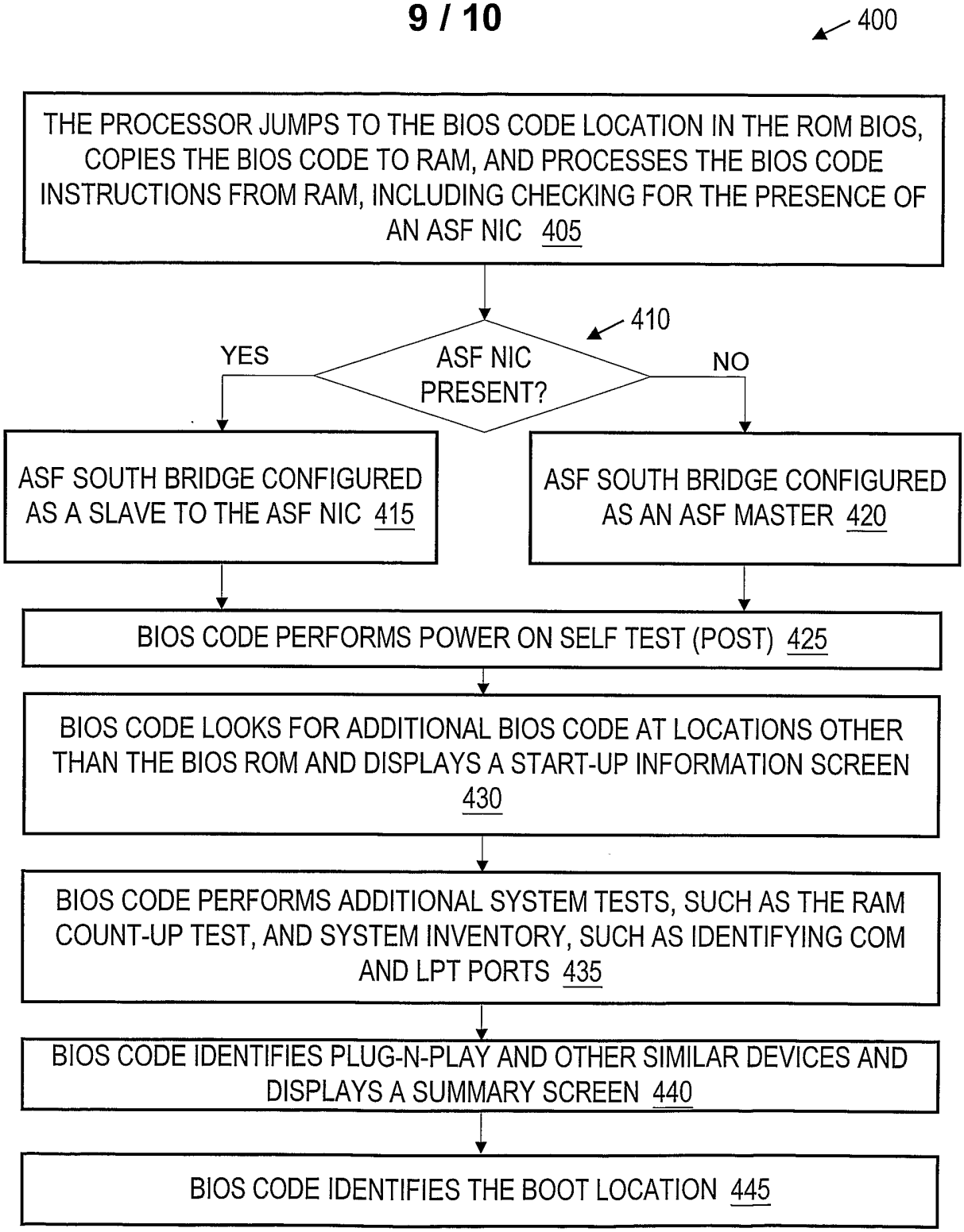

400
THE PROCESSOR JUMPS TO THE BIOS CODE LOCATION IN THE ROM BIOS, COPIES THE BIOS CODE TO RAM, AND PROCESSES THE BIOS CODE INSTRUCTIONS FROM RAM, INCLUDING CHECKING FOR THE PRESENCE OF AN ASF NIC 405
410
ASF NIC PRESENT?
YES
NO
ASF SOUTH BRIDGE CONFIGURED AS A SLAVE TO THE ASF NIC 415
ASF SOUTH BRIDGE CONFIGURED AS AN ASF MASTER 420
BIOS CODE PERFORMS POWER ON SELF TEST (POST) 425
BIOS CODE LOOKS FOR ADDITIONAL BIOS CODE AT LOCATIONS OTHER THAN THE BIOS ROM AND DISPLAYS A START-UP INFORMATION SCREEN 430
BIOS CODE PERFORMS ADDITIONAL SYSTEM TESTS, SUCH AS THE RAM COUNT-UP TEST, AND SYSTEM INVENTORY, SUCH AS IDENTIFYING COM AND LPT PORTS 435
BIOS CODE IDENTIFIES PLUG-N-PLAY AND OTHER SIMILAR DEVICES AND DISPLAYS A SUMMARY SCREEN 440
BIOS CODE IDENTIFIES THE BOOT LOCATION 445

Fig. 6

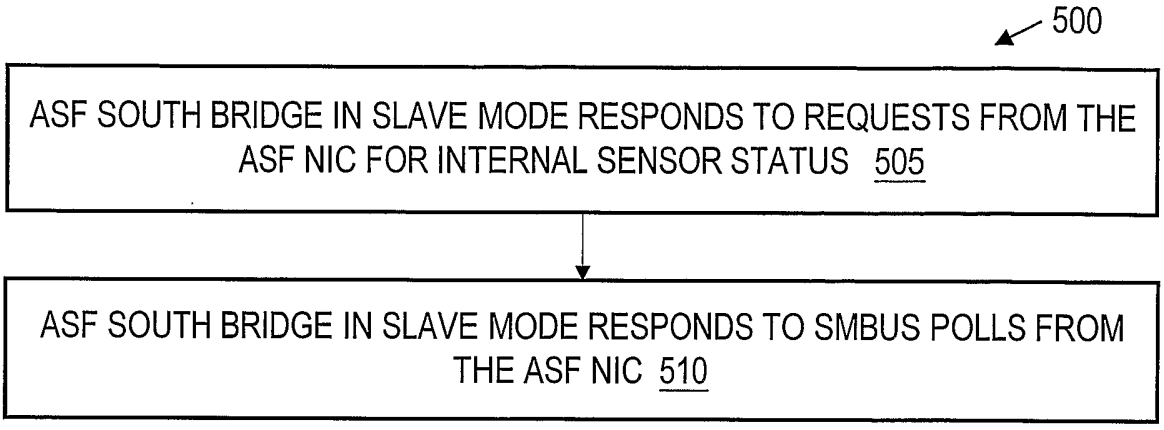

500
ASF SOUTH BRIDGE IN SLAVE MODE RESPONDS TO REQUESTS FROM THE ASF NIC FOR INTERNAL SENSOR STATUS 505
ASF SOUTH BRIDGE IN SLAVE MODE RESPONDS TO SMBUS POLLS FROM THE ASF NIC 510

Fig. 7A

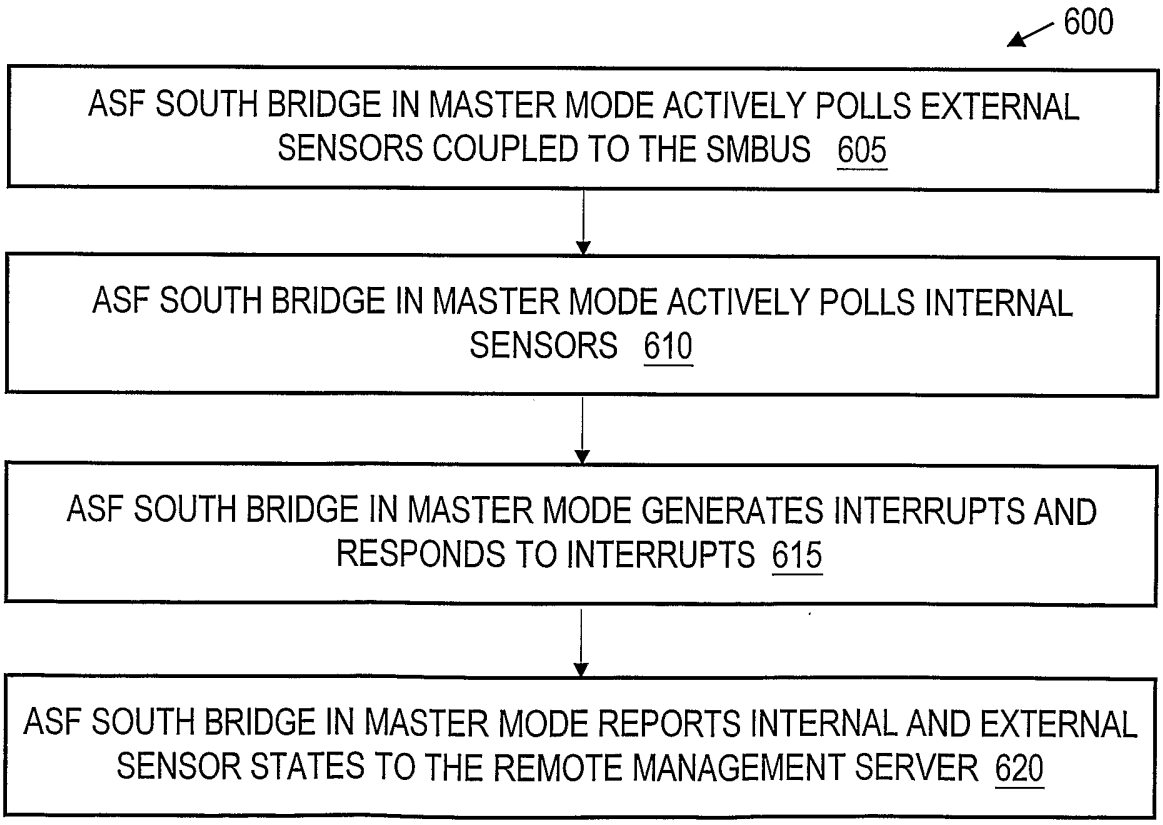

600
ASF SOUTH BRIDGE IN MASTER MODE ACTIVELY POLLS EXTERNAL SENSORS COUPLED TO THE SMBUS 605
ASF SOUTH BRIDGE IN MASTER MODE ACTIVELY POLLS INTERNAL SENSORS 610
ASF SOUTH BRIDGE IN MASTER MODE GENERATES INTERRUPTS AND RESPONDS TO INTERRUPTS 615
ASF SOUTH BRIDGE IN MASTER MODE REPORTS INTERNAL AND EXTERNAL SENSOR STATES TO THE REMOTE MANAGEMENT SERVER 620

Fig. 7B