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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 December 2010 (16.12.2010)

PCT

(10) International Publication Number
WO 2010/144164 A1

(51) International Patent Classification:
G06F 13/14 (2006.01) *G06F 12/02* (2006.01)
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/US2010/027003

(22) International Filing Date:
11 March 2010 (11.03.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/482,614 11 June 2009 (11.06.2009) US

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

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **ESPIG, Michael E.** [US/US]; 22955 SW HIllsboro Hwy., Newberg, Oregon 97132 (US). **FANG, Zhen** [CN/US]; 5922 NW Snowlily Dr., Portland, Oregon 97229 (US). **IYER, Ravishankar** [IN/US]; 15934 NW Andalusian Way, Portland, Oregon 97229 (US). **HARRIMAN, David J.** [US/US]; 2758 SW English Ct., Portland, California 97201 (US).

(74) Agents: **TROP, Timothy N.** et al.; Trop, Pruner & Hu, P.C., 1616 S. Voss Rd., Ste. 750, Houston, TX 77057-2631 (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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, 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, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

(54) **Title**: DELEGATING A POLL OPERATION TO ANOTHER DEVICE

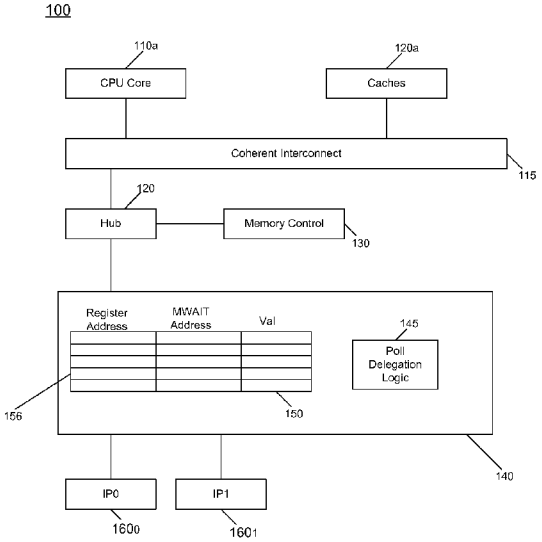

FIG. 1

(57) **Abstract**: In one embodiment, the present invention includes a method for handling a registration message received from a host processor, where the registration message delegates a poll operation with respect to a device from the host processor to another component. Information from the message may be stored in a poll table, and the component may send a read request to poll the device and report a result of the poll to the host processor based on a state of the device. Other embodiments are described and claimed.

WO 2010/144164 A1

DELEGATING A POLL OPERATION TO ANOTHER DEVICE

Background

[0001] Modern computer systems typically include a processor and various other components that are coupled together. In addition, many systems include one or more peripheral or input/output (IO) devices.

[0002] To enable communications between software that executes on the processor and operations that may be performed by the other devices, different mechanisms can be used. Common mechanisms include a polling method and an interrupt method. However, neither of these methods is optimal. Using a polling technique, software either continuously polls status registers on the IO device if the IO device's task is fine-grained, or relies on an asynchronous interrupt through the operating system (OS) if the IO device's task is coarse-grained. While a polling method may ensure good performance, it suffers from drawbacks. First, the core/thread that needs to know the completion status has to continuously check (e.g., via a busy spin operation) on a memory mapped input/output (MMIO) status register, preventing itself from entering a low power state. Second, repetitive polling on an uncacheable MMIO address results in a large amount of traffic on a system interconnect. In a word, the fast response time comes at a cost of power consumption (a major issue especially for ultra-low power environments) and waste of system resources.

[0003] An interrupt method avoids busy spinning of a processor on the status register. While waiting, the core/thread can either context switch to execute another process or enter a lower power state. Completion of the task on the IO device triggers an interrupt into the OS. However, in a typical system, several hundred cache misses and tens of thousand clock cycles are induced by a kernel interrupt handler. This performance overhead of interrupt handling is not acceptable for many fine-grained logic blocks.

[0004] Thus both polling and interrupt techniques are not satisfactory for a low power application, as polling negates a large portion of any power benefits from using an IO device, while interrupts introduce a large performance penalty.

Brief Description of the Drawings

[0005] FIG. 1 is a block diagram of a system in accordance with one embodiment of the present invention.

[0006] FIG. 2 is a flow diagram of a method in accordance with one embodiment of the present invention.

[0007] FIG. 3 is a block diagram of a system in accordance with another embodiment of the present invention.

Detailed Description

[0008] In various embodiments, a poll delegation technique may be implemented in which an interconnect serves as a delegate in a polling and notification process. In one embodiment the interconnect may be an input/output (IO) interconnect, although the scope of the present invention is not limited in this regard. Using this technique, the interconnect polls IO devices for a host processor such as a central processing unit (CPU) and notifies an application software of a given event using one of a number of techniques such as a test and hold operation, or by update to a user-selected memory location that triggers a processor's exit from a power optimized state. In one embodiment, user-level instructions such as MONITOR/MWAIT may be used to notify application software. In various embodiments, poll delegation may enable a response time as short as polling, and power consumption/resource usage as low as an interrupt-based technique, thus providing user-level notification of IO device status without the need for polling or interrupts.

[0009] In one embodiment the IO interconnect may include specific-purpose hardware to poll status register(s) of an IO device. Then upon a status change, the IO interconnect can issue a write operation (e.g., a coherent write) to the memory address that is being monitored by the host. The coherent write will be detected by this hardware, and cause the thread that is waiting on the address to resume execution. Thus in various embodiments, a processor can stay in a low power state until the IO device is done with its task, and resume execution almost as fast as if it had been busy-spinning. No change is required in the processor core, the cache, the system coherent interconnect, or the IO devices.

[0010] The MONITOR/MWAIT pair of instructions can support inter-thread synchronization. The instruction pair can be available at all privilege levels. MONITOR can be used to enable a CPU to set up monitoring hardware of the CPU to detect stores to an effective address range (typically a cacheline). This address range belongs to a coherent, write-back address range. In one embodiment, cache coherency hardware may monitor for a write to the destination address. When that write occurs the cache coherency controller will send a message to the processor to come out of the low power state. After

this set up, the succeeding MWAIT instruction puts the processor core into a selected low-power state (e.g., a clock-gated state or a power-gated state). When the monitoring hardware detects a store to any byte in the address range, the stalled thread resumes execution from the instruction following MWAIT. Architecturally, MWAIT behaves like a no operation (NOP). While these MONITOR and MWAIT instructions are designed to implement performance and power-optimized inter-thread synchronization, embodiments can leverage the instructions for IO device completion notification.

[0011] Referring now to FIG. 1, shown is a block diagram of a system in accordance with one embodiment of the present invention. As shown in FIG. 1, system 100 may be a system on a chip (SoC) that includes various components interconnected together and which may provide mechanisms to handle polling delegation in accordance with an embodiment of the present invention. Specifically, in the embodiment of FIG. 1, SoC 100 may include a plurality of processor cores only one of which, core 110_a, is shown for ease of illustration.

[0012] The one or more cores may be coupled via a coherent interconnect 115 to one or more cache memory 120_a. Coherent interconnect 115 may include various hardware, software and/or firmware to implement a cache coherency protocol, such as a modified exclusive shared invalid (MESI) protocol, to maintain a coherent view of information stored within the system. In some implementations, coherent interconnect 115 may be a layered protocol including various layers such as a protocol layer, a link layer and possibly a physical layer (where the system is not on a single die).

[0013] In turn, coherent interconnect 115 may be coupled via a hub 120 to a memory controller 130 that in turn may be coupled to a system memory, e.g., dynamic random access memory (DRAM), for example. Note that such memory is not shown in FIG. 1, as the memory may be external to the SoC.

[0014] In addition, coherent interconnect 115 may be coupled to an upstream side of an IO interconnect 140 which may be of a given communication protocol such as a Peripheral Component Interconnect Express (PCI Express™ (PCIe™)) protocol in accordance with links based on the PCI Express™ Specification Base Specification version 2.0 (published January 17, 2007) (hereafter the PCIe™ Specification), or another such protocol. IO interconnect 140 may include a polling table 150 in accordance with an embodiment of the present invention. While shown as being present in the interconnect, other implementations may locate this buffer elsewhere in close relation to the interconnect. In

turn, various devices, e.g., devices 160_0 and 160_1, which may be IO devices, intellectual property (IP) blocks or so forth may be coupled to a downstream side of IO interconnect 140.

[0015] As seen in FIG. 1, polling table 150 stores a plurality of entries including, e.g., entry 156. Each entry may include a device monitored location such as an address of a status register present in one of devices 160, a memory monitored address, which may correspond to a physical address in system memory that corresponds to a monitored location for an MWAIT instruction, and a initial value, which may be the initial value of the device monitored location, e.g., the initial value of the status register. Embodiments thus essentially associate each status register on an IO device with a corresponding memory address that is being monitored by the CPU. While shown with this particular implementation in the embodiment of FIG. 1, the scope of the present invention is not limited in this regard.

[0016] Referring now to FIG. 2, shown is a flow diagram of a method in accordance with one embodiment of the present invention. Specifically, FIG. 2 shows a flow diagram that may be performed by logic of an IO or other interconnect that handles poll delegation on behalf of a host processor. In some implementations, method 200 may be implemented by way of a state machine or other logic of an interconnect, although other implementations are possible. As shown in FIG. 2, method 200 may begin by receiving a registration message from an application (block 210). In one embodiment, the registration message may include a tuple of MMIO address, true physical memory address, and initial value. In one embodiment, this registration message may include a physical address range for a location and a monitored system memory that backs up a monitored virtual address range, a MMIO address for obtaining the status of a location in a device to be monitored, e.g., a status register, and an initial value of such register, which may also correspond to the initial value stored in the monitored address of system memory. In one embodiment, an application may make a system called to pass this information to the interconnect, e.g., using a host processor.

[0017] Still referring to FIG. 2, it may be determined whether an entry is available in the polling table of the interconnect (diamond 215). If not, a failure message may be sent back to the host (block 220). Otherwise, the information may be stored into an entry of the polling table (block 230). This information thus may correspond to a device monitored location (i.e., corresponding to a MMIO address of a completion status register of the

device), an initial value of the register, and a memory monitored address (which may be a physical address location within the system memory that is monitored, e.g., by a MONITOR/MWAIT instruction pair). Accordingly, a success message may be sent from the interconnect to the host processor (block 240).

[0018] During operation, the application then initializes the monitored location, issuing the MONITOR and MWAIT instructions, thus enabling the device to begin executing its task. Various such tasks may be realized, including offloading of specialized functions, graphics processing, physics processing or so forth. As one example, the function may be a specialized calculation such as a fast fourier transform (FFT). The application thus may pass various information regarding the FFT such as the number of points, the starting address and so forth, prior to execution of the MONITOR/MWAIT instructions.

[0019] Accordingly, at this time one or more cores of the host processor may enter a low power state, which may be configurable depending on a type of operation that the device is to perform.

[0020] Referring still to FIG. 2, after the device begins executing its operation, the interconnect may perform polling at the device monitored locations (block 245). In one embodiment, the polling operation may issue a read operation to each register address in the polling table, and compare the fetched value against the initial value in the polling table entry, as discussed with regard to diamond 250. Understand that such polling may poll a number of locations present in one or more devices, namely whatever addresses are indicated in a polling table. It may be determined on each polling operation whether there is a change in any entries' value from its initial value (diamond 250). If not, a further polling iteration may occur. If a change occurs, control instead passes to block 260, where a write operation may be issued to the memory monitored address with the new value. In one embodiment, the interconnect may issue a coherent write to this address, which may be realized using a message signaling interrupt (MSI), in one embodiment. This coherent write to the monitored address in memory will cause the processor core to wake up from the low power state, enabling the halted thread to proceed. With reference to FIG. 2, it may be determined after issuing the write operation whether a release of the entry has been received (diamond 270). In one embodiment, such a release may be received from the OS when a given application associated with the poll delegation entry terminates. Accordingly, if the device knows the task is done, the device could issue an eviction request to the IO interconnect. Control passes from diamond 270, if the release is not

indicated, back to block 245, where further polling may be performed. While shown with this particular implementation in the embodiment of FIG. 2, the scope of the present invention is not limited in this regard.

[0021] The generation of the registration message and release message may use help from the OS. The user-level MONITOR and MWAIT instructions may be performed completely in user mode, and the poll delegation operation can be purely hardware. Note that the registration and release are usually only performed at the application initialization and cleanup phases. Therefore their power and performance do not matter. In contrast, the user-level setup and poll detection usually are executed a large number of times. The efficiency of these two steps thus enables efficient system power and performance characteristics.

[0022] To support multiple IP accelerators, polling table 150 may be a multi-entry translation table. In one embodiment, each entry 156 contains an MMIO address and the physical memory address that is linked to it. In some implementations, the number of entries in the polling table can be a small number, e.g., N=8. In the extreme case when more than N registers need to be checked at the same time, the user thread can always directly poll the registers instead of using the above method, although a virtualized polling table could instead be used.

[0023] In one embodiment, a message signaling interrupt (MSI-X) feature of PCI that may be in IO interconnect 140 provides hardware that allows devices 160 to issue writes to system memory locations. In MSI-X, the target memory locations are special addresses that will lead to interrupts. In embodiments instead addresses in writeback memory space can be used as targets of such writes. The property of the target memory location is transparent to MSI-X hardware, it simply delivers a packet from the IO interconnect to the memory system. In some implementations the polling performed by interconnect 140 may cause a poll of registered status register addresses, even if the devices that some of the registers represent are not actively computing. This is because in some implementations the poll delegation logic 145 has no knowledge of whether or not a valid entry in its mapping table represents an inactive device. Different mechanisms can be used to provide this information to the logic. For example, a system call can be provided by the OS to allow an application to release a particular entry in the polling table. Alternatively, the IO interconnect 140 could intercept power-state transition commands that are sent to the IP blocks 160 so that it will know which status register will not be updated any time soon.

This information may be included in a status portion of the entries of polling table 150 in such embodiments. It is noted however, that the cost for the interconnect 140 to poll IO registers is rather low, and as such the power and performance impact of indiscriminate polling may be minimal.

[0024] While described herein for a system-on-chip (SoC) configuration, which may be the primary processing component for a computing device such as an embedded, portable or mobile device, other implementations may be used in other systems such as multiprocessor computer systems having a processor coupled to a coherent interconnect, that in turn may be coupled to an IO interconnect via one or more chipsets or other components. Still further, embodiments may be implemented in a multi-chip architecture for a computing device.

[0025] Referring now to FIG. 3, shown is a block diagram of a system in accordance with another embodiment of the present invention. As shown in FIG. 3, system 300 may be multi-chip architecture, namely a system 300 including a first chip 100, which may be a first SoC, and which may be configured the same as that of FIG. 1, a second integrated circuit 310, which may provide peripheral functionality, and a memory 370, which may be a DRAM coupled to one or more of the chips. As shown in FIG. 3, SoC 100 may communicate with IC 310 via an interconnect 305 that in turn is coupled to a first bridge 170 of SoC 100 and a second bridge 320 of IC 310. In turn, second bridge 320 may be coupled to an IO interconnect 330 that in turn can be coupled to one or more peripheral devices, e.g., a PCIe device 340 and a universal serial bus (USB) device 350. Still further, IO interconnect 330 may be coupled via a third bridge 360 to off chip components via a serial over network interface (SONIC)/external fabric. In the embodiment of FIG. 3, note that both IO interconnect 140 and IO interconnect 330 may include polling tables and poll delegation logic in accordance with an embodiment of the present invention. While shown with this particular implementation in the embodiment of FIG. 3, other implementations are of course possible.

[0026] Using an embodiment of the present invention, a process can avoid either suffering from long latency for interrupt handling, or have to busy-spin in a high power state. On a processor that supports MONITOR/MWAIT or similar test and set functions, poll delegation allows the processor to enter power and performance-optimized states while still achieving the quick response time of busy spins. For low-power SoCs that

include finter-grained IP blocks, embodiments provide a near-optimal completion notification solution in terms of power and performance.

[0027] Embodiments may be implemented in code and may be stored on a storage medium having stored thereon instructions which can be used to program a system to perform the instructions. The storage medium may include, but is not limited to, any type of disk including floppy disks, optical disks, compact disk read-only memories (CD-ROMs), compact disk rewritables (CD-RWs), and magneto-optical disks, semiconductor devices such as read-only memories (ROMs), random access memories (RAMs) such as dynamic random access memories (DRAMs), static random access memories (SRAMs), erasable programmable read-only memories (EPROMs), flash memories, electrically erasable programmable read-only memories (EEPROMs), magnetic or optical cards, or any other type of media suitable for storing electronic instructions.

[0028] While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.

What is claimed is:

1. An apparatus comprising:

a core to generate a registration message to delegate a poll operation to an input/output (IO) interconnect;

a coherent interconnect coupled to the core;

the IO interconnect coupled to the coherent interconnect, the IO interconnect including a poll table having a plurality of entries each having a register address field to store a register address received in a registration message, a destination address field to store a destination address in a system memory received in the registration message, and an initial value field to store an initial value associated with the register address received in the registration message; and

at least one device coupled to the IO interconnect to perform an operation for an application executing on the core and including at least one status register, the IO interconnect to poll the at least one status register responsive to information in a poll table entry, and to issue a write transaction to the destination address if a polled value of the at least one status register differs from the initial value.

2. The apparatus of claim 1, wherein the at least one device is to update the initial value to the polled value upon completion of the operation.

3. The apparatus of claim 1, wherein the apparatus comprises a system on a chip (SoC) formed on a single semiconductor die, and the at least one device comprises an intellectual property (IP) block.

4. The apparatus of claim 1, wherein the IO interconnect includes a poll delegation logic.

5. The apparatus of claim 4, wherein the poll delegation logic is to issue a read request to the at least one device at a predetermined interval to perform the poll.

6. The apparatus of claim 5, wherein the poll delegation logic is to perform a comparison between data received from the at least one device responsive to the read request and the initial value and to perform the write transaction when the data and the initial value differ.

7. The apparatus of claim 1, wherein the core is to send an eviction message to the IO interconnect to delete an entry in the poll table when an application corresponding to the entry is terminated.

8. The apparatus of claim 1, wherein the core is to execute a first instruction to set up the destination address and a second instruction to cause the core to enter a low power state until the destination address is updated.

9. The apparatus of claim 8, wherein the core is to send the registration message to the IO interconnect prior to execution of the first and second instructions.

10. A system comprising:

a first integrated circuit including:

at least one core including a first logic to execute a first instruction to set up a monitored address in a memory and a second logic to cause the at least one core to enter a low power state when a predetermined instruction follows the first instruction;

a first coherent interconnect coupled to the at least one core;

a first input/output (IO) interconnect including a poll table to store a tuple including a register identifier of a register in an intellectual property (IP) block coupled to the first IO interconnect, the monitored address in the memory, and an initial value associated with the register, and a delegation logic to receive a delegation message from the at least one core, and based on the tuple, obtain a current value of the register until the current value differs from the initial value and responsive to the difference write data to the monitored address; and

the IP block coupled to the first IO interconnect including the register and to perform a function for an application executing on the at least one core; and

the memory coupled to the first integrated circuit via a memory interconnect, wherein the at least one core is to exit the low power state responsive to the data being written to the destination address of the memory and to continue execution of the application at a next instruction following the predetermined instruction.

11. The system of claim 10, wherein the IP block is to update the initial value to the current value upon completion of the operation.

12. The system of claim 10, wherein the at least one core is to send the delegation message to the first IO interconnect with the tuple prior to execution of the first instruction.

13. A method comprising:

receiving a registration message from a host processor in an interconnect coupled between the host processor and a device, the registration message to delegate a poll operation with respect to the device to the interconnect;

storing information regarding a device monitored location, a memory monitored address, and an initial value of the device monitored location in a poll table associated with the interconnect; and

sending a read request from the interconnect to the device to poll the device, and reporting a result of the poll to the host processor if a device value obtained from the device monitored location is different than the initial value.

14. The method of claim 13, further comprising issuing a write request from the interconnect to the memory monitored address in a system memory to report the result.

15. The method of claim 14, further comprising issuing a message signaling interrupt from the interconnect to the system memory to issue the write request.

16. The method of claim 15, wherein the host processor is placed into a low power state after sending the registration message, and responsive to the write to the memory monitored address, the host processor is to exit the low power state.

17. The method of claim 13, wherein the poll table includes a plurality of entries each including a device monitored location, a memory monitored address, and an initial value, and further comprising sending a read request to the device for each of the plurality of entries.

18. The method of claim 13, further comprising using a poll delegation logic of the interconnect to send the read request and to compare the initial value with the device value.

19. The method of claim 13, wherein the device monitored location corresponds to a status register of the device, and associating the status register with the memory monitored address via the poll table.

20. The method of claim 13, further comprising initiating the registration message via an application, and performing a first function on the device for the application, wherein the application is to initiate the first function after the information is stored in the poll table, and to cause the host processor to enter into a low power state.

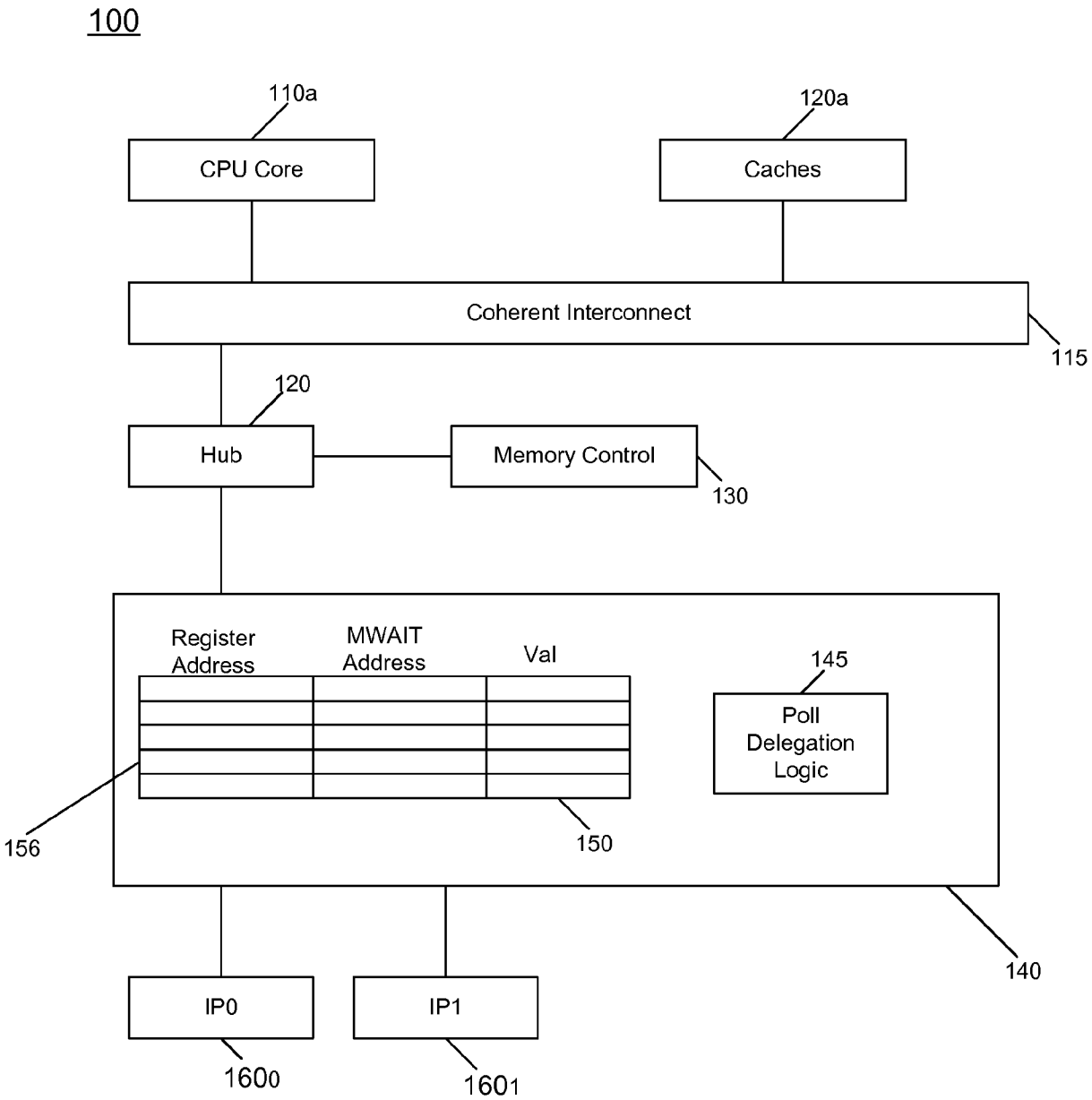
100
110a
CPU Core
120a
Caches
Coherent Interconnect
115
120
Hub
Memory Control
130
Register Address
MWAIT Address
Val
145
Poll Delegation Logic
156
150
140
IP0
IP1
1600
1601

FIG. 1

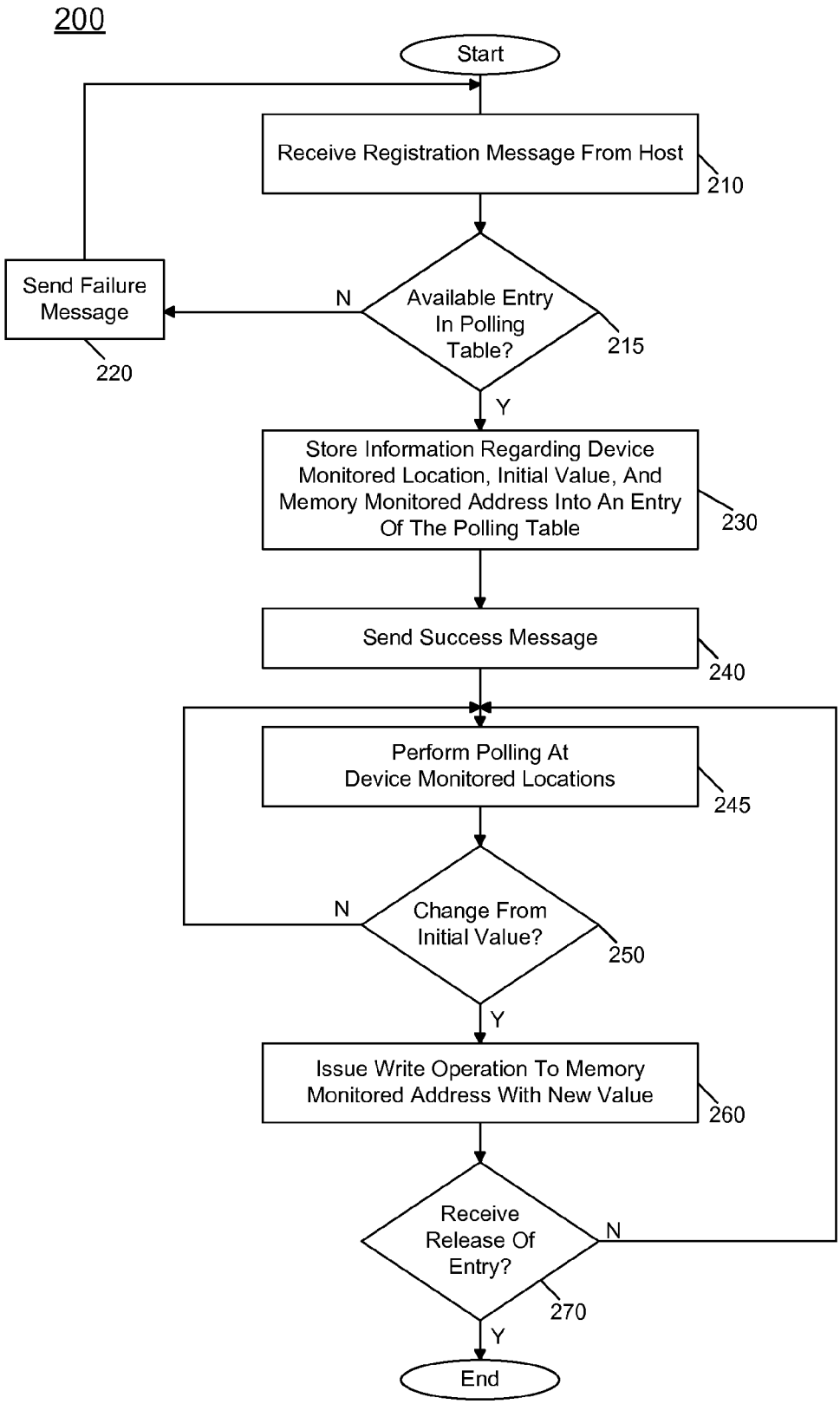
200
Start
Receive Registration Message From Host
210
Available Entry In Polling Table?
215
N
Send Failure Message
220
Y
Store Information Regarding Device Monitored Location, Initial Value, And Memory Monitored Address Into An Entry Of The Polling Table
230
Send Success Message
240
Perform Polling At Device Monitored Locations
245
Change From Initial Value?
250
N
Y
Issue Write Operation To Memory Monitored Address With New Value
260
Receive Release Of Entry?
270
N
Y
End

FIG. 2

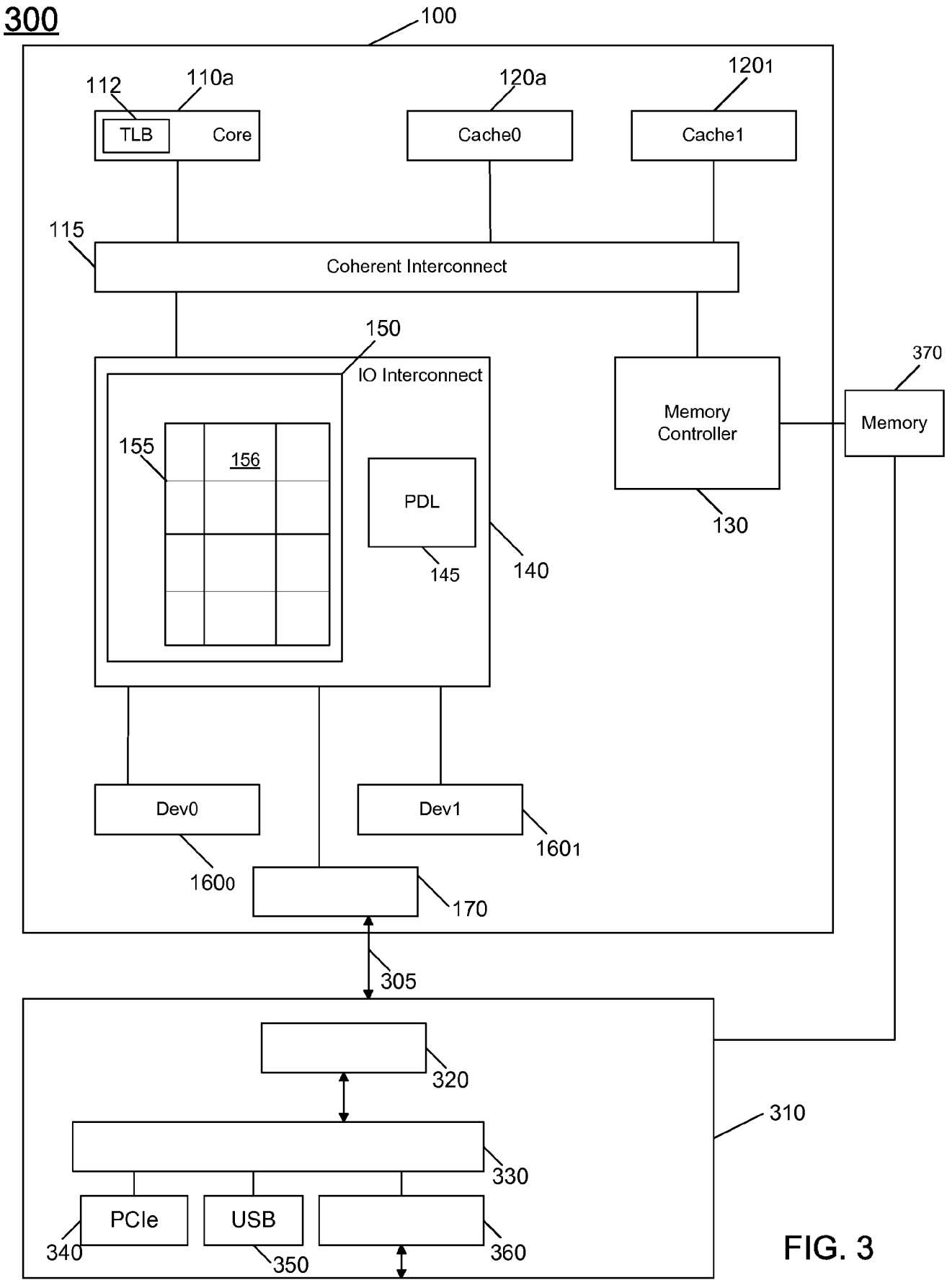
300
100
112
110a
TLB
Core
120a
Cache0
1201
Cache1
115
Coherent Interconnect
150
IO Interconnect
155
156
PDL
145
140
Memory Controller
130
370
Memory
Dev0
1600
Dev1
1601
170
305
320
310
330
PCIe
340
USB
350
360

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2010/027003

A. CLASSIFICATION OF SUBJECT MATTER

G06F 13/14(2006.01)i, G06F 17/30(2006.01)i, G06F 12/02(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 13/14; G06F 15/16; G06F 12/00; G06F 9/06; H04L 9/00; H03M 1/68; G06F 11/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: core, interconnect, device, poll, delegate, registration, message, coherent, table, entry, register, and address

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 7509540 B1 (DAVID M. LOVY et al.) 24 March 2009 See col. 8, line 33 - col. 17, line 6; figures 4-10.	1-20
A	US 2006-0277594 A1 (ARLINDO CHIAVEGATTO JR. et al.) 07 December 2006 See paragraphs [0033] - [0057]; figures 1-4.	1-20
A	US 2005-0223178 A1 (DAVID J. GARCIA et al.) 06 October 2005 See paragraphs [0012] - [0043]; figures 1-5B.	1-20
A	KR 10-2006-0112184 A (MICROSOFT CORP.) 31 October 2006 See page 3, line 31 - page 11, line 23; figures 1-5.	1-20
A	KR 10-2006-0126065 A (SAMSUNG ELECTRONICS CO., LTD.) 07 December 2006 See page 3, line 20 - page 4, line 28; figures 2-5.	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 SEPTEMBER 2010 (30.09.2010)	**30 SEPTEMBER 2010 (30.09.2010)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

KIM Jong Kee

Telephone No. 82-42-481-8301

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2010/027003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7509540 B1	24.03.2009	US 2009-0216881 A1	27.08.2009
		US 7197561 B1	27.03.2007
		US 7296194 B1	13.11.2007
US 2006-0277594 A1	07.12.2006	None	
US 2005-0223178 A1	06.10.2005	CN 100442248 C	10.12.2008
		CN 1776647 A	24.05.2006
		CN 1776647 C0	10.12.2008
KR 10-2006-0112184 A	31.10.2006	AU 2004-279195 A1	23.06.2005
		AU 2004-279195 A8	23.06.2005
		AU 2004-279195 B2	01.04.2010
		CA 2501729 A1	23.04.2005
		CN 101410795 A	15.04.2009
		CN 1836208 A0	20.09.2006
		EP 1673694 A2	28.06.2006
		EP 1678653 A2	12.07.2006
		JP 2007-509404 A	12.04.2007
		JP 2007-519075 A	12.07.2007
		KR 10-2007-0012178 A	25.01.2007
		US 2005-0091227 A1	28.04.2005
		US 2005-0091635 A1	28.04.2005
		US 2005-0091640 A1	28.04.2005
		US 2005-0091647 A1	28.04.2005
		US 2005-0114485 A1	26.05.2005
		US 2005-0114494 A1	26.05.2005
		US 7103874 B2	05.09.2006
		US 7506307 B2	17.03.2009
		US 7539974 B2	26.05.2009
		US 7676560 B2	09.03.2010
		US 7712085 B2	04.05.2010
		US 7765540 B2	27.07.2010
		WO 2005-045559 A2	19.05.2005
		WO 2005-045559 A3	19.05.2005
		WO 2005-045739 A2	19.05.2005
		WO 2005-045739 A3	19.05.2005
KR 10-2006-0126065 A	07.12.2006	None	