



- (51) **International Patent Classification:**
G06F 21/50 (2013.01) *G06F 21/78* (2013.01)
- (21) **International Application Number:**
PCT/US2016/063764
- (22) **International Filing Date:**
25 November 2016 (25.11.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
14/998,362 24 December 2015 (24.12.2015) US
- (71) **Applicant:** INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).
- (72) **Inventors:** NARENDRA TRIVEDI, Alpa; 6434 NE Meadowgate Ct., Hillsboro, Oregon 97124 (US). SAHITA, Ravi; 7854 SW Kimber Place, Beaverton, Oregon 97007 (US). DURHAM, David; 20431 SW Tremont

Way, Beaverton, Oregon 97007 (US). GREWAL, Karanvir; 2631 NE 9th Drive, Hillsboro, Oregon 97124 (US). DEWAN, Prashant; 119 NE Atlantic Place, Hillsboro, Oregon 97124 (US). CHHABRA, Siddhartha; 1221 NE 51st. Avenue, Apt. 30, Hillsboro, Oregon 97124 (US).

(74) **Agent:** DYER, Richard A; Kacvinsky Daisak Bluni PLLC, 430 Davis Drive, Suite 150, Morrisville, North Carolina 27560 (US).

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

[Continued on next page]

(54) **Title:** SYSTEM AND METHOD FOR ENABLING SECURE MEMORY TRANSACTIONS USING ENCLAVES

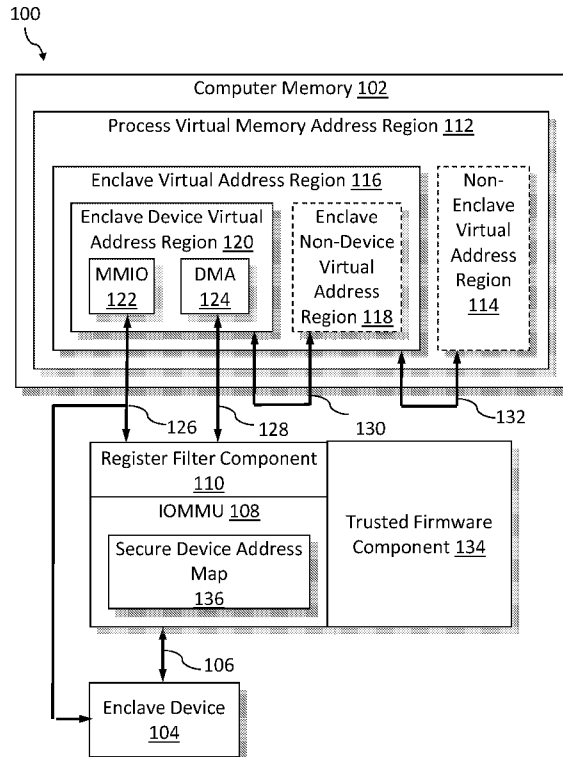


FIG. 1

(57) **Abstract:** Various embodiments are generally directed to an apparatus, method, and other techniques to provide direct-memory access, memory-mapped input-output, and/or other memory transactions between devices designated for use by an enclave and the enclave itself. A secure device address map may be configured to map addresses for the enclave device and the enclave, and a register filter component may grant access to the enclave device to the enclave.



(84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE,

SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *with international search report (Art. 21(3))*
- *with amended claims (Art. 19(1))*

SYSTEM AND METHOD FOR ENABLING SECURE MEMORY TRANSACTIONS USING ENCLAVES

CROSS-REFERENCE TO RELATED APPLICATIONS

5 This application claims the benefit of and priority to previously filed United States Patent Application Serial Number 14/998,362, filed December 24, 2015, the subject matter of which is incorporated herein by reference in its entirety.

BACKGROUND

10 A trusted execution environment, referred to herein as an “enclave,” protects the execution of trusted computer-program code from potentially malicious other code by executing the trusted code in a secure region of computer memory that is inaccessible to other code – a so-called “inverse sandbox” mechanism. In this approach, rather than attempting to identify and isolate all the malicious code on a system, trusted code may be sealed inside the enclave and protected from attack by the malicious code, irrespective of the privilege level of the latter. The enclave
15 even protects the trusted code from malicious software in a system’s operating system or other malicious software having heightened system privileges.

In order to preserve the security of the enclave, communication between the enclave and hardware devices on or connected to the system must be trusted. For example, a device may encrypt data before sending it to the enclave which may then decrypt it; any malicious code
20 would therefore be able to view only encrypted data. This encryption and decryption is, however, both time- and power-consuming and may be the cause of a significant performance and/or battery-life problem on some systems, especially mobile systems.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates an example embodiment of a computing environment;

25 **FIGS. 2A** and **2B** illustrate example embodiments of a computer memory;

FIG. 3 illustrates an example embodiment of an enclave;

FIG. 4 illustrates an example embodiment of an enclave device;

FIG. 5 illustrates an example embodiment of a computing system; and

FIG. 6 illustrates a flow diagram.

DETAILED DESCRIPTION

30 In various embodiments, an enclave memory region and/or access control model includes a region of memory allocated for access by an enclave device. This allocated region of memory may be specific to and exclusive to each device, but any number of devices may be assigned any number of allocated regions of enclave memory. The device may communicate securely with

the enclave by writing to and/or reading from the allocated region of memory; the rest of the enclave (e.g., the non-allocated region) may write to or read from the allocated region of memory. Thus the enclave device may communicate with the enclave without overhead associated with, for example, encrypting and decrypting communications. The communications
5 may include direct memory access communications, memory mapped input-output communications, or any other device to memory communications.

In various embodiments, a trusted firmware component identifies a device enclave mode bit assigned to the enclave device that is transmitted with a memory transaction associated with the device. The enclave device may be configured to store the device enclave mode bit in its
10 memory and to transmit the device enclave mode bit with each transaction with the enclave; no specialized hardware or software may be necessary to be installed and/or implemented on the enclave device, and any device may be therefore used as an enclave device. Once the trusted firmware component identifies the device enclave mode bit, it may load an associated secure device address map into a memory translation map disposed between the enclave device and the
15 enclave to map physical addresses, as seen by the enclave device, to enclave device virtual addresses. Any number of enclaves and/or enclave devices may thus be accommodated; when a memory transaction between an enclave and an enclave device is desired, the associated secure device address map is loaded. In some embodiments, the secure device address map is loaded into an input-output memory management unit.

In some embodiments, a register filter component grants access to the enclave device to the
20 enclave device virtual memory. The register filter component may include a base and a device mask; the enclave device is permitted read and/or write access to addresses in the enclave device virtual memory falling between only the base and device mask. Other code, such as operating-system code or application program code, is denied access to the addresses in the enclave device
25 virtual memory falling between the base and device mask.

FIG. 1 illustrates a computing environment 100 that includes a computer memory 102 and an enclave device 104. In various embodiments, the computer memory 102 is volatile memory such as static and/or dynamic random-access memory. An exemplary type of dynamic random-access memory is double-data-rate, fourth generation, synchronous dynamic random-access
30 memory (“DDR4 SDRAM”); any type of static and/or dynamic random-access memory is, however, within the scope of the embodiments described herein. In some embodiments, the computer memory 102 includes, instead of or in addition to volatile memory, non-volatile memory such as a read-only memory, flash memory, phase-change memory, or any other type of non-volatile memory.

The enclave device 104 may be any input/output device, controller, peripheral, transceiver, or any other type of device that communicates with the computer memory 102. Only one enclave device 104 is shown in the computing environment 100, but one of skill in the art will understand that any number of enclave devices 104 may be used. In some embodiments, the enclave device 104 is disposed on the same silicon substrate as the computer memory 102 in a system-on-a-chip configuration; in other embodiments, the enclave device 104 is disposed on a different silicon substrate as the computer memory 102 but in the same chip package as the computer memory 102; in other embodiments, the enclave device 104 is disposed in a different chip package as the computer memory 102 but disposed on the same circuit board and/or in the same housing as the computer memory 102; in still other embodiments, the enclave device 104 is disposed in a different package or housing than the computer memory 102 and communicates therewith via a network connection. Any disposition of the computer memory 102 and enclave device 104 is, however, within the scope of the embodiments described herein.

A link 106 may connect the computer memory 102 and the enclave device 104 and allow electronic communication of data therebetween (via, in some embodiments, an input-output memory management unit ("IOMMU") 108 and/or register filter component 110, as explained in greater detail below). The link 106 may include metal lines in a system-on-a-chip metal stack, traces on a circuit board, package connectors, network links (e.g., a WI-FI or ETHERNET connection), and/or any other type of electrical communication link. The link 106 may include a serial bus, parallel bus, or any other configuration of conductors or signals.

In some embodiments, the computer memory 102 includes a process virtual memory address region 112. As one of skill in the art will understand, a virtual-machine monitor or hypervisor may allocate the process virtual memory address region 112 to a process, program, operating system, or other collection of software instructions executing on a computer system such that the process, program, operating system, or other collection of software instructions may read or write data and/or instructions to the process virtual memory address region 112 as if it were a physical memory exclusive to the process, program, operating system, or other collection of software instructions. The process virtual memory address region 112 may correspond to a contiguous or non-contiguous range of addresses in the computer memory 102. Although only one process virtual memory address region 112 is shown in the environment 100, the computer memory 102 may include any number of additional process virtual memory address regions (associated with a corresponding process, program, operating system, or other collection of software instructions) consistent with the system constraints of the environment 100.

The process virtual memory address region 112 may include virtual addresses; these virtual addresses are shown as a non-enclave virtual address region 114 to distinguish them from an enclave virtual address region 116. The non-enclave virtual address region 114 may include any instructions and/or data associated with its process; the non-enclave virtual address region 114 may not be, however, secure or trusted, and may contain malicious code and/or be vulnerable to malicious code.

The process virtual memory address region 112 may include an enclave virtual address region 116 associated with an enclave, which may also be known as a trusted execution environment. One of skill in the art will understand that the enclave virtual address region 116 may be created, modified, removed, used, or have any other number of operations performed on it or performed using it in accordance with enclave software and/or hardware instructions, such as the SOFTWARE GUARD EXTENSIONS (“SGX”) instructions provided by Intel Corporation of Santa Clara, California. For example, SGX instructions EENTER and/or EERESUME may be used to enter the enclave virtual address region 116 from the non-enclave virtual address region 114, and SGX instructions EEXIT and/or AEX may be used to leave the enclave virtual address region 116 to the non-enclave virtual address region 114. Any trusted execution environment, is, however, within the scope of the embodiments described herein, and the embodiments are not limited to only SGX instructions.

The enclave virtual memory address region 116 may include an enclave non-device virtual address region 118 for storing instructions and/or data associated with code in the enclave. In addition, in accordance with embodiments described herein, the enclave virtual memory address region 116 may further include an enclave device virtual address region 120 for communicating with the device 104. As explained in greater detail below, the enclave device virtual address region 120 may include a memory mapped input output (“MMIO”) region 122 and/or a direct-memory access (“DMA”) region 124.

The enclave device 104 may communicate with the computer memory using an MMIO link 126 and/or using a DMA link 128 (via the IOMMU 108 and/or register filter component 110). The enclave non-device virtual address region 118 may communicate with the enclave device virtual address region using in-enclave memory operations 130, which may include raising interrupts. For example, the enclave device may transfer data to the enclave device virtual address region 120 using the MMIO link 126 and/or the DMA link 128; once transferred, code in the enclave non-device virtual address region 118 may access the transferred data using one or more in-enclave memory operations 130. The MMIO link 126 and/or the DMA link 128 may correspond to physical wires, routes, busses, or similar electrical conductors; the in-enclave

memory operations 130 may correspond to data moving in accordance with software instructions. In some embodiments, the enclave is entered or exited using enter or exit instructions 132.

As described in greater detail below, a trusted firmware component 134 may identify a device enclave mode bit in communications to or from the enclave device 104 (using, e.g., the link 106, MMIO link 126, or any other input or output link to or from the enclave device 104). If the device enclave mode bit is detected, the trusted firmware component may configure a secure device address map 136 in the IOMMU 108 to map addresses for that particular enclave device 104. The secure device address map 136 for the enclave device 104 may be stored in the enclave virtual address region 116, as may the device mode enable bit.

FIGS. 2A and 2B illustrate exemplary memories 200A, 200B, respectively, in accordance with embodiments described herein. Turning first to FIG. 2A, the memory 202A includes a first section of enclave memory 204A and a second section of hypervisor-accessible memory 206A (i.e., non-enclave memory). Any computer code, software, programs, or data may be stored in or execute from the hypervisor-accessible memory 206A, such as an untrusted virtual machine 208A, a trusted virtual machine 210A, or any other virtual machine, operating system, or other such construct.

In various embodiments, an enclave device 212A communicates with the enclave memory 204A. A secure device address map 214A maps physical addresses to and from enclave virtual addresses in a DMA region 216A in the enclave memory 204A. In some embodiments, the secure device address map 214A is part of an input-output memory management unit; in other embodiments, the secure device address map 214A resides in trusted firmware, processor microcode, and/or in hardware. One of skill in the art will understand, however, that any system or method of DMA mapping is within the scope of the present invention. The secure device address map 214A may include a look-up table that pairs physical addresses with enclave virtual addresses and that outputs a physical address given an enclave virtual address (and vice versa); the secure device address map 214A may instead or in addition calculate a physical address and/or enclave virtual address given one or the other by, for example, adding an offset to the given address. Any implementation of the secure device address map 214A is, however, within the scope of the embodiments described herein.

In some embodiments, a base 218, device mask 220, and/or mask 222 values may be used to grant access for the enclave device 212A to the DMA region 216A. These values 218A, 220A, 222A may be stored in a register, such as the register filter 108 of FIG. 1, or in any other register or memory. In some embodiments, a memory transaction associated with the enclave

device 212A, such as a DMA read or write, is permitted only if a memory address associated with the transaction falls within the range defined by the base 218A and device mask 220A, thus blocking the enclave device 212A from accessing other portions of the enclave memory 204A.

FIG. 2B illustrates the memory 200B in accordance with embodiments described herein.

5 Like the memory 200A of FIG. 2A, the memory 200B includes an enclave memory 204B, a hypervisor-accessible memory 206B (containing an untrusted virtual machine 208B, a trusted virtual machine 210B, and/or any other such constructs), a secure device address map 214B, and base 218B, device mask 220B, and mask 222B values. In this embodiment, however, there are two enclave devices: a first enclave device 224B and a second enclave device 226B. Within the
10 enclave memory 204B, a first enclave DMA region 228B associated with a first enclave is assigned to the first enclave device 224B, and a second enclave DMA region 230 associated with a second enclave is assigned to the second enclave device 226B. In other words, different enclaves may communicate independently with different enclave devices; this communication may be exclusive (i.e., the first enclave device 224B may not be permitted to communicate with
15 the second enclave DMA region 230), or multiple enclaves may share the same enclave device(s). As explained in greater detail below, the secure device address map 214B may be configured when an enclave is active to map any enclave devices assigned to that enclave to an assigned enclave DMA region.

FIG. 3 illustrates an exemplary enclave 300 in accordance with embodiments described
20 herein. As mentioned above, the enclave 300 may include enclave data 302 and/or enclave instructions 304 in accordance with enclave operations and functions. As also mentioned above, the enclave 300 may include an enclave device virtual address region 306 for communicating with enclave devices. In addition, in some embodiments, the enclave 300 includes an enclave control structure 308; the enclave control structure 308 may be the SGX enclave control
25 structure ("SECS") associated with SGX enclaves, but the embodiments disclosed herein are not limited to any particular implementation of the enclave control structure 308. The enclave control structure 308 may contain data and/or metadata associated with the setup, configuration, and/or status of the enclave 300. In some embodiments, the enclave control structure 308 is a memory page in the enclave 300; in other embodiments, the enclave control structure 308 is
30 stored in another memory, in firmware, or in any other similar storage.

In some embodiments, the enclave control structure 308 includes a device enclave mode bit 310. The device enclave mode bit 310 may be one or more binary digits that signify that an enclave device is in communication with the enclave 300; in some embodiments, the device enclave mode bit 310 identifies the enclave device to the enclave 300. In other embodiments,

any other bits may be used to identify the enclave device, such as a machine-access code. For example, if the device enclave mode bit 310 is zero, the enclave 300 is not in communication with an enclave device, and if the device enclave mode bit 310 is one, the enclave 300 is in communication with an enclave device. In other embodiments, if the device enclave mode bit 310 is non-zero and contains multiple bits, the bits may identify a particular device as the enclave device.

The enclave control structure 308 may further include a secure device address map 312. As mentioned above, the secure device address map 312 may be used to map physical addresses to enclave device virtual addresses. In some embodiments, as described in more detail below, a component such as the trusted firmware component 134 of FIG. 1 may load the map from the secure device address map 312 and store it in the IOMMU 108 for a particular enclave device and enclave.

FIG. 4 illustrates an enclave device 400. The enclave device 400 may include a memory 402 for storing that enclave device's device mode enable bit 300 as received from, for example, the trusted firmware component 134 of FIG. 1. The enclave device 400 may further include a network interface 406 for communicating with an enclave. The enclave device may include any other component or system, such as a processor, non-volatile storage, keyboard, display, touchscreen, bus controller, or any other such component; one of skill in the art will understand that the embodiments disclosed herein are not limited to any particular device 400.

FIG. 5 illustrates a computing device 500 in accordance with embodiments described herein. The computing device 500 may include a processor 502, a computer memory 504 (i.e., volatile storage), one or more enclave devices 506, an IOMMU 508, storage 510 (i.e., non-volatile storage), trusted firmware 512, a network interface 514, and any other components or features, which may communicate via a link 516. The processor 502 may be any type of processing circuit such as a central-processing unit, digital-signal processor, application-specific integrated circuit, or similar processor, and may include one or units, cores, or co-processors, some or all of which may operate in a low-power mode. The computer memory 504 may be random-access memory such as DDR memory, read-only memory, or any other type of memory. The enclave device(s) 506 may include a keyboard, mouse, touchscreen, microphone, LCD monitor, disk controller, bus controller, touchscreen, speaker, vibration generator or any other similar input and/or output device. The IOMMU 508 may be hardware, software, and/or firmware and may be implemented as a stand-alone device, as part of the processor 502, as part of the trusted firmware 512, or as any other implementation. The storage 510 may be any magnetic hard disk drive, solid-state hard disk drive, flash memory, or similar non-volatile

storage or removable media such as a CD, DVD, or USB flash drive. The trusted firmware 512 may be implemented as a stand-alone device, as part of the processor 502, as part of the IOMMU 508, or as any other implementation. The network interface 214 may include a cellular transceiver, a WI-FI transceiver, a BLUETOOTH transceiver, an ETHERNET transceiver, or
5 any other similar network interface. The link 216 may be any system bus, interconnect, network link, or similar link, and may include the DMA and/or MMIO links described above.

The computer memory 504 may be used to store instructions to be executed by the processor 502. These instructions may be part of an enclave 518, part of a trusted virtual machine 520, part of an untrusted virtual machine 522, or any other instructions. The
10 instructions may include an operating system (e.g., a Microsoft WINDOWS, Linux, ANDROID, APPLE IOS, or APPLE OS X operating system) that directs the execution of low-level, basic system functions (such as memory allocation, file management, and the operation of mass storage devices), as well as higher-level application routines. The instructions may include any
15 suitable programming language, including, without limitation high-level languages such as C, C++, Java, Perl, Python, or Ruby or low-level assembly languages. The computer memory 504 may further store input and/or output data associated with execution of the instructions as well as additional information used by the various application routines.

FIG. 6 illustrates a flow diagram 600 in accordance with embodiments disclosed herein. A device enclave mode bit, such as the device enclave mode bit 310 of FIG. 3, assigned to an
20 enclave device, such as the enclave device 104 of FIG. 1, is identified (602), using a trusted firmware component such as the trusted firmware component 134 of FIG. 1, and transmitted with a memory transaction associated with the enclave device 104 and an associated enclave, such as the enclave 300 of FIG. 3. An input/output memory management unit, such as the
25 input/output memory management unit 108 of FIG. 1, is configured (604), using the trusted software component 134, with a secure device address map, such as the secure device address map 136 of FIG. 1, associated with the device enclave mode bit 310. Access to the enclave device is granted (606), using a register filter component, to the enclave device virtual address region. Data is transmitted (608) from the enclave device to an enclave device virtual address region associated with the enclave or from the enclave device virtual address region associated
30 with the enclave to the enclave device.

The following paragraphs describe one system and method that implements the embodiments described herein with particular reference to SGX instructions. One of skill in the art will understand, however, that this implementation is not limiting and that the embodiments described herein may be implemented on any system using any instruction set.

One SGX instruction, EADD, commits virtual memory pages within a process virtual memory address region 112 to the enclave virtual address region 120. In some embodiments, the EADD instruction is extended to create the device MMIO region 122 and/or the device DMA region 124. In these embodiments, EADD (which may be given the argument ENCLS[EAX=01
5 h]) assigns the device MMIO region 122 from the enclave virtual address region 116, from the non-enclave virtual address region 114, or from any other OS-accessible region into the enclave virtual address region 116, associates the device MMIO region 122 with an enclave control structure 308, and/or stores security attributes, such as an associated device enclave mode bit 310, in the enclave control structure 308, in an enclave page cache map, or in any other location.
10 As part of the association, the security attributes may be measured and extended into a field that stores and maintains enclave metadata and other attributes, such as SECS.MRENCLAVE. In some embodiments, the EADD instruction may only be executed when a current privilege level is set to a heightened or restricted level, such as level 0. General process registers may be used to store information related to the committed virtual memory pages; for example, an RBX
15 register may be configured to contain the effective address of a PAGEINFO structure, while an RCX register may be configured to contain the effective address of an enclave page cache page.

In some embodiments, the enclave virtual address region 116 includes a plurality of different page types, such as a PT_SECS page type that is configured to store an enclave control structure 308. The number and type of page types may be extended in accordance with
20 embodiment disclosed herein. For example, a PT_DEV page type may be defined to store a device MMIO region 122. In some embodiments, the device MMIO region 122 may correspond to physical memory that corresponds to the enclave device configuration space. An enclave device 104 may be configured to accept configuration input-output accesses from only an associated enclave via use of the device enclave mode bit 310.

25 In some embodiments, the enclave virtual address region 116 instead or in addition includes a PT_DMA page type: for a device DMA region 124. Different types of DMA regions 124, such as read-only or read-write types, may be created using the existing permissions bits in the PAGEINFO.SECINFO field, such as read (R), write (W), and/or execute (X) permission bits.

The SGX instructions, in this example, further include EENTER/ERESUME and
30 EEXIT/AEX instructions to transition 132 between an enclave virtual address region 116 and a non-enclave virtual address region 114. These instructions may be used to transition into and out of an enclave execution context for an enclave that has device access (configured, for example, via a license/capability granted to the enclave or for specific types of architectural enclaves). In various embodiments, these instructions are extended to context switch two additional fields in

the enclave-control structure for the enclave: the device-enclave-mode bit 310 on the logical processor that is in enclave mode of operation and the secure device address map 312. In various embodiments, the processor 502 and/or microcode executing thereon is responsible for sending this context switch to the IOMMU 108 (and/or other system-on-a-chip fabric logic) to
5 load the secure device address map 312 into the IOMMU 108 if required. In some embodiments, an input-output translation-lookaside buffer is flushed during this operation.

The enclave device 104 may signal activities/events to the enclave 300 by using an enclave specific interrupt remapping table which may be used to assign specific interrupts to specific enclaves. When the enclave device 104 is configured to be in secure mode (i.e., it issues device
10 input/output cycles with the device enclave mode bit 310 set), interrupts sent to an enclave 300 may not be receivable by the enclave 300 (due to, for example, it being scheduled out). For such operations, the enclave device 104 may post an interrupt event to the enclave MMIO region 122, which may then be delivered to the enclave 300 when it is scheduled and an ERESUME is executed. The processor 502 may deliver the information of the device event to the enclave
15 virtual address region 116 via a posted interrupt descriptor, and a notification event flag may be set when the ERESUME successfully transitions control to the enclave 300. In response, the device enclave code may perform normal memory operations to the device MMIO region 122 to acknowledge the posted interrupt.

Example 1 is an apparatus for secure enclave device memory access comprising a
20 computer processor; an enclave disposed in a computer memory to execute trusted computer instructions using the computer processor and comprising an enclave device virtual address region; an input/output memory management unit to map a memory address from a physical address to an enclave device virtual address in the enclave device virtual address region; a register filter component to grant an enclave device access to the enclave device virtual address
25 region; a trusted firmware component to identify a device enclave mode bit assigned to the enclave device and transmitted with a memory transaction associated with the enclave device and to configure the input/output memory management unit with a secure device address map associated with the device enclave mode bit.

Example 2 includes the subject matter of example 1, where the register filter component
30 comprises a base register and a device mask register.

Example 3 includes the subject matter of example 2, where the base register is further configured for storing a base memory address of the enclave and the device mask register is configured for storing a size of the enclave device virtual address region.

Example 4 includes the subject matter of example 1, where the enclave further comprises an enclave non-device virtual address region in communication with the enclave device virtual address region.

5 Example 5 includes the subject matter of example 1, where the secure device address map is stored in an enclave control structure associated with the enclave.

Example 6 includes the subject matter of example 1, where the memory transaction comprises a direct-memory access transaction or a memory-mapped input-output transaction.

10 Example 7 includes the subject matter of example 1, where the trusted firmware component configures the input/output memory management unit with the secure device address map using a software-guard extension instruction.

Example 8 includes the subject matter of example 7, where the software-guard extension instruction comprises an EENTER, EERESUME, EEXIT, or AEX instruction.

Example 9 includes the subject matter of example 1, where the computer memory comprises volatile or non-volatile memory.

15 Example 10 includes the subject matter of example 1, further comprising a plurality of enclave devices.

Example 11 includes the subject matter of example 1, where the enclave device is disposed on a same silicon substrate as the computer memory.

20 Example 12 includes the subject matter of example 1, further comprising a link between the computer memory and the enclave device.

Example 13 includes the subject matter of example 12, where the link comprises a system-on-a-chip metal stack, a trace on a circuit board, a package connector, or a network link.

Example 14 includes the subject matter of example 1, where the secure device address map comprises a look-up table.

25 Example 15 is a computer-implemented method for secure enclave device memory access comprising identifying, using a trusted firmware component, a device enclave mode bit assigned to an enclave device and transmitted with a memory transaction associated with the enclave device and an associated enclave; configuring, using the trusted firmware component, an input/output memory management unit with a secure device address map associated with the
30 device enclave mode bit; granting access to the enclave device, using a register filter component, to the enclave device virtual address region; and transmitting data from the enclave device to an enclave device virtual address region associated with the enclave or from the enclave device virtual address region associated with the enclave to the enclave device.

Example 16 includes the subject matter of example 15, where the register filter component grants access to the enclave device using a base register and a device mask register.

Example 17 includes the subject matter of example 15, where the base register stores a base memory address of the enclave and the device mask register is configured for storing a size
5 of the enclave device virtual address region.

Example 18 includes the subject matter of example 15, where the enclave device virtual address region communicates with an enclave non-device virtual address region.

Example 19 includes the subject matter of example 15, further comprising storing the secure device address map in an enclave control structure associated with the enclave.

10 Example 20 includes the subject matter of example 15, where the memory transaction comprises a direct-memory access transaction or a memory-mapped input-output transaction.

Example 21 includes the subject matter of example 15, further comprising configuring, using the trusted firmware component, the input/output memory management unit with the secure device address map using a software-guard extension instruction.

15 Example 22 includes the subject matter of example 21, where the software-guard extension instruction comprises an EENTER, EERESUME, EEXIT, or AEX instruction.

Example 23 includes the subject matter of example 15, where the enclave residing in a computer memory comprises volatile or non-volatile memory.

20 Example 24 includes the subject matter of example 15, further comprising identifying a plurality of device enclave mode bits assigned to a plurality of enclave devices.

Example 25 includes the subject matter of example 15, where the enclave device is disposed on a same silicon substrate as the computer memory.

Example 26 includes the subject matter of example 15, further comprising linking between the computer memory and the enclave device.

25 Example 27 includes the subject matter of example 26, where the linking comprises a system-on-a-chip metal stack linking, a tracing on a circuit board, a package connecting, or a network linking.

Example 28 includes the subject matter of example 15, where the secure device address map comprises a look-up table.

30 Example is a tangible machine-readable storage medium comprising instructions that, when executed by a processing device, cause the processing device to: identify, using a trusted firmware component, a device enclave mode bit assigned to an enclave device and transmitted with a memory transaction associated with the enclave device and an associated enclave; configure, using the trusted software component, an input/output memory management unit with

a secure device address map associated with the device enclave mode bit; grant access to the enclave device, using a register filter component, to the enclave device virtual address region; and transmit data from the enclave device to an enclave device virtual address region associated with the enclave or from the enclave device virtual address region associated with the enclave to the enclave device.

Example 30 includes the subject matter of example 29, where the register filter component is configured to grant access to the enclave device using a base register and a device mask register.

Example 31 includes the subject matter of example 29, where the base register is configured to store a base memory address of the enclave and the device mask register is configured for storing a size of the enclave device virtual address region.

Example 32 includes the subject matter of example 29, where the enclave device virtual address region is configured to communicate with an enclave non-device virtual address region.

Example 33 includes the subject matter of example 29, where the instructions further cause the processing device to store the secure device address map in an enclave control structure associated with the enclave.

Example 34 includes the subject matter of example 29, where the memory transaction comprises a direct-memory access transaction or a memory-mapped input-output transaction.

Example 35 includes the subject matter of example 29, where the instructions further cause the processing device to configure, using the trusted firmware component, the input/output memory management unit with the secure device address map using a software-guard extension instruction.

Example 36 includes the subject matter of example 35, where the software-guard extension instruction comprises an EENTER, EERESUME, EEXIT, or AEX instruction.

Example 37 includes the subject matter of example 29, where the enclave residing in a computer memory comprises volatile or non-volatile memory.

Example 38 includes the subject matter of example 29, further comprising identifying a plurality of device enclave mode bits assigned to a plurality of enclave devices.

Example 39 includes the subject matter of example 29, where the enclave device is disposed on a same silicon substrate as the computer memory.

Example 40 includes the subject matter of example 29, further comprising linking between the computer memory and the enclave device.

Example 41 includes the subject matter of example 40, where the linking comprises a system-on-a-chip metal stack linking, a tracing on a circuit board, a package connecting, or a network linking.

Example 42 includes the subject matter of example 29, where the secure device address map comprises a look-up table.

As used in this application, the terms “system” and “component” are intended to refer to a hardware and/or software device in which digital processing is carried out, examples of which are provided by this depicted exemplary processing architecture. For example, a component can be, but is not limited to being, a process running on a processor component, the processor component itself, a storage device (e.g., a hard disk drive, multiple storage drives in an array, etc.) that may employ an optical and/or magnetic storage medium, a software object, an executable sequence of instructions, a thread of execution, a program, and/or an entire device (e.g., an entire computer). One or more components can reside within a process and/or thread of execution, and a component may be localized on one device and/or distributed between two or more devices. Further, components may be communicatively coupled to each other by various types of communications media to coordinate operations.

It should also be noted that embodiments may be provided as one or more computer-readable programs embodied on or in one or more articles of manufacture. The article of manufacture may be any suitable hardware apparatus, such as, for example, a floppy disk, a hard disk, a CD ROM, a CD-RW, a CD-R, a DVD ROM, a DVD-RW, a DVD-R, a flash memory card, a PROM, a RAM, a ROM, or a magnetic tape. In general, the computer-readable programs may be implemented in any programming language. Some examples of languages that may be used include C, C++, or JAVA. The software programs may be further translated into machine language or virtual machine instructions and stored in a program file in that form. The program file may then be stored on or in one or more of the articles of manufacture.

Some embodiments may be described using the expression “one embodiment” or “an embodiment” along with their derivatives. These terms mean that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment. Further, some embodiments may be described using the expression “coupled” and “connected” along with their derivatives. These terms are not necessarily intended as synonyms for each other. For example, some embodiments may be described using the terms “connected” and/or “coupled” to indicate that two or more elements are in direct physical or electrical contact with each other. The term

"coupled," however, may also mean that two or more elements are not in direct contact with each other, but yet still co-operate or interact with each other.

It is emphasized that the Abstract of the Disclosure is provided to allow a reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In addition, in the foregoing Detailed Description, it can be seen that various features are grouped together in a single embodiment for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus the following claims are hereby incorporated into the Detailed Description, with each claim standing on its own as a separate embodiment. In the appended claims, the terms "including" and "in which" are used as the plain-English equivalents of the respective terms "including" and "wherein," respectively. Moreover, the terms "first," "second," "third," and so forth, are used merely as labels, and are not intended to impose numerical requirements on their objects.

What has been described above includes examples of the disclosed architecture. It is, of course, not possible to describe every conceivable combination of components and/or methodologies, but one of ordinary skill in the art may recognize that many further combinations and permutations are possible. Accordingly, the novel architecture is intended to embrace all such alterations, modifications, and variations that fall within the spirit and scope of the appended claims.

What is claimed is:

CLAIMS

1. An apparatus for secure enclave device memory access comprising:
a computer processor;
5 an enclave disposed in a computer memory to execute trusted computer instructions using the computer processor and comprising an enclave device virtual address region;
an input/output memory management unit to map a memory address from a physical address to an enclave device virtual address in the enclave device virtual address region;
a register filter component to grant an enclave device access to the enclave device virtual
10 address region;
a trusted firmware component to identify a device enclave mode bit assigned to the enclave device and transmitted with a memory transaction associated with the enclave device and to configure the input/output memory management unit with a secure device address map associated with the device enclave mode bit.
- 15 2. The apparatus of claim 0, the register filter component comprising a base register and a device mask register.
3. The apparatus of claim 0, the base register further configured for storing a base memory address of the enclave and the device mask register is configured for storing a size of the enclave device virtual address region.
- 20 4. The apparatus of claim 0, the enclave further comprising an enclave non-device virtual address region in communication with the enclave device virtual address region.
5. The apparatus of claim 0, the secure device address map being stored in an enclave control structure associated with the enclave.
6. The apparatus of claim 0, the memory transaction comprising a direct-memory access
25 transaction or a memory-mapped input-output transaction.
7. The apparatus of claim 0, the trusted firmware component configuring the input/output memory management unit with the secure device address map using a software-guard extension instruction.
8. The apparatus of claim 0, the software-guard extension instruction comprising an
30 EENTER, EERESUME, EEXIT, or AEX instruction.
9. A computer-implemented method for secure enclave device memory access comprising:
identifying, using a trusted firmware component, a device enclave mode bit assigned to an enclave device and transmitted with a memory transaction associated with the enclave device and an associated enclave;

configuring, using the trusted software component, an input/output memory management unit with a secure device address map associated with the device enclave mode bit;

granting access to the enclave device, using a register filter component, to the enclave device virtual address region; and

5 transmitting data from the enclave device to an enclave device virtual address region associated with the enclave or from the enclave device virtual address region associated with the enclave to the enclave device.

10. The computer-implemented method of claim 0, the register filter component granting access to the enclave device using a base register and a device mask register.

10 11. The computer-implemented method of claim 0, the base register storing a base memory address of the enclave and the device mask register is configured for storing a size of the enclave device virtual address region.

12. The computer-implemented method of claim 0, the enclave device virtual address region communicating with an enclave non-device virtual address region.

15 13. The computer-implemented method of claim 0, further comprising storing the secure device address map in an enclave control structure associated with the enclave.

14. The computer-implemented method of claim 0, the memory transaction comprising a direct-memory access transaction or a memory-mapped input-output transaction.

15. The computer-implemented method of claim 0, further comprising configuring, using the
20 trusted firmware component, the input/output memory management unit with the secure device address map using a software-guard extension instruction.

16. The computer-implemented method of claim 0, the software-guard extension instruction comprising an EENTER, EERESUME, EEXIT, or AEX instruction.

17. At least one machine readable medium comprising a plurality of instructions that in
25 response to being executed by system at a storage device cause the system to carry out a method according to any one of claims 0 to 0.

18. An apparatus comprising means for performing the methods of any one of claims 0 to 0.

AMENDED CLAIMS

received by the International Bureau on 16 May 2017 (16.05.2017)

What is claimed is:

1. An apparatus for secure enclave device memory access comprising:
a computer processor;
an enclave disposed in a computer memory to execute trusted computer instructions using the computer processor and comprising an enclave device virtual address region;
an input/output memory management unit to map a memory address from a physical address to an enclave device virtual address in the enclave device virtual address region;
a register filter component to grant an enclave device access to the enclave device virtual address region;
a trusted firmware component to identify a device enclave mode bit assigned to the enclave device and transmitted with a memory transaction associated with the enclave device and to configure the input/output memory management unit with a secure device address map associated with the device enclave mode bit.
2. The apparatus of claim 1, the register filter component comprising a base register and a device mask register.
3. The apparatus of claim 2, the base register further configured for storing a base memory address of the enclave and the device mask register is configured for storing a size of the enclave device virtual address region.
4. The apparatus of claim 1, the enclave further comprising an enclave non-device virtual address region in communication with the enclave device virtual address region.
5. The apparatus of claim 1, the secure device address map being stored in an enclave control structure associated with the enclave.
6. The apparatus of claim 1, the memory transaction comprising a direct-memory access transaction or a memory-mapped input-output transaction.
7. The apparatus of claim 1, the trusted firmware component configuring the input/output memory management unit with the secure device address map using a software-guard extension instruction.
8. The apparatus of claim 7, the software-guard extension instruction comprising an EENTER, EERESUME, EEXIT, or AEX instruction.
9. A computer-implemented method for secure enclave device memory access comprising:

identifying, using a trusted firmware component, a device enclave mode bit assigned to an enclave device and transmitted with a memory transaction associated with the enclave device and an associated enclave;

configuring, using the trusted software component, an input/output memory management unit with a secure device address map associated with the device enclave mode bit;

granting access to the enclave device, using a register filter component, to the enclave device virtual address region; and

transmitting data from the enclave device to an enclave device virtual address region associated with the enclave or from the enclave device virtual address region associated with the enclave to the enclave device.

10. The computer-implemented method of claim 9, the register filter component granting access to the enclave device using a base register and a device mask register.

11. The computer-implemented method of claim 10, the base register storing a base memory address of the enclave and the device mask register is configured for storing a size of the enclave device virtual address region.

12. The computer-implemented method of claim 9, the enclave device virtual address region communicating with an enclave non-device virtual address region.

13. The computer-implemented method of claim 9, further comprising storing the secure device address map in an enclave control structure associated with the enclave.

14. The computer-implemented method of claim 9, the memory transaction comprising a direct-memory access transaction or a memory-mapped input-output transaction.

15. The computer-implemented method of claim 9, further comprising configuring, using the trusted firmware component, the input/output memory management unit with the secure device address map using a software-guard extension instruction.

16. The computer-implemented method of claim 15, the software-guard extension instruction comprising an EENTER, EERESUME, EEXIT, or AEX instruction.

17. At least one machine readable medium comprising a plurality of instructions that in response to being executed by system at a storage device cause the system to carry out a method according to any one of claims 9 to 16.

18. An apparatus comprising means for performing the methods of any one of claims 9 to 16.

1 / 7

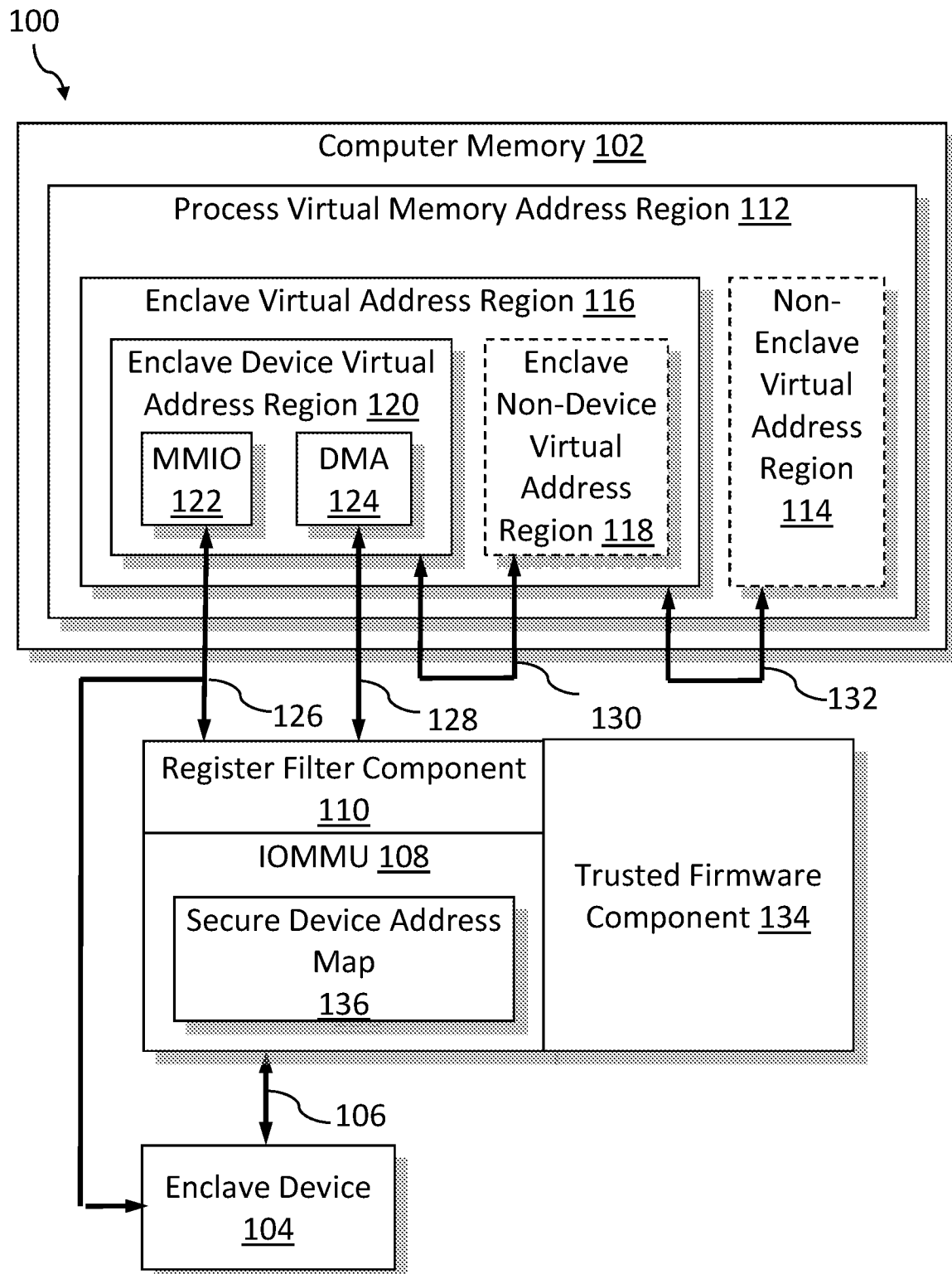


FIG. 1

2 / 7

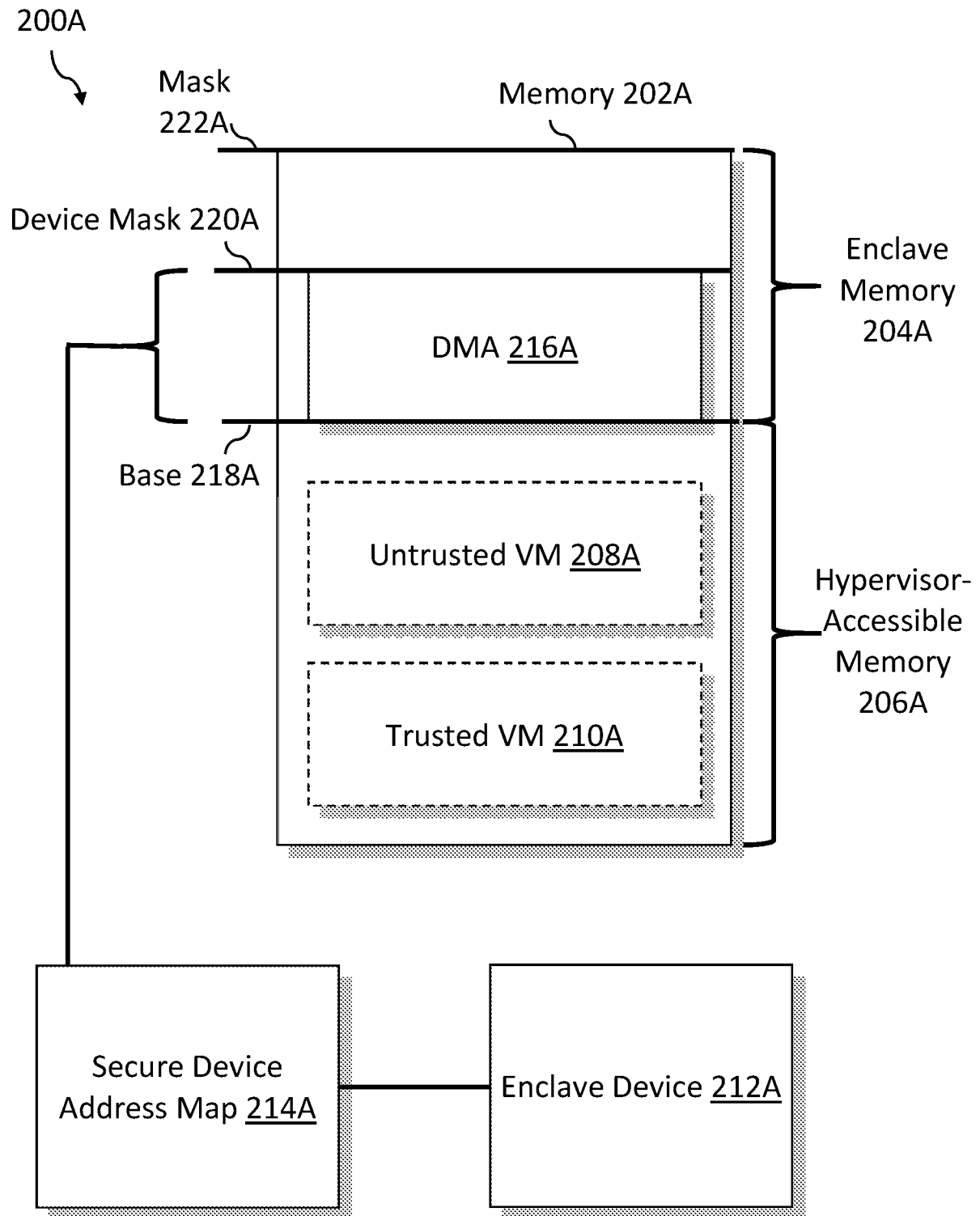


FIG. 2A

3 / 7

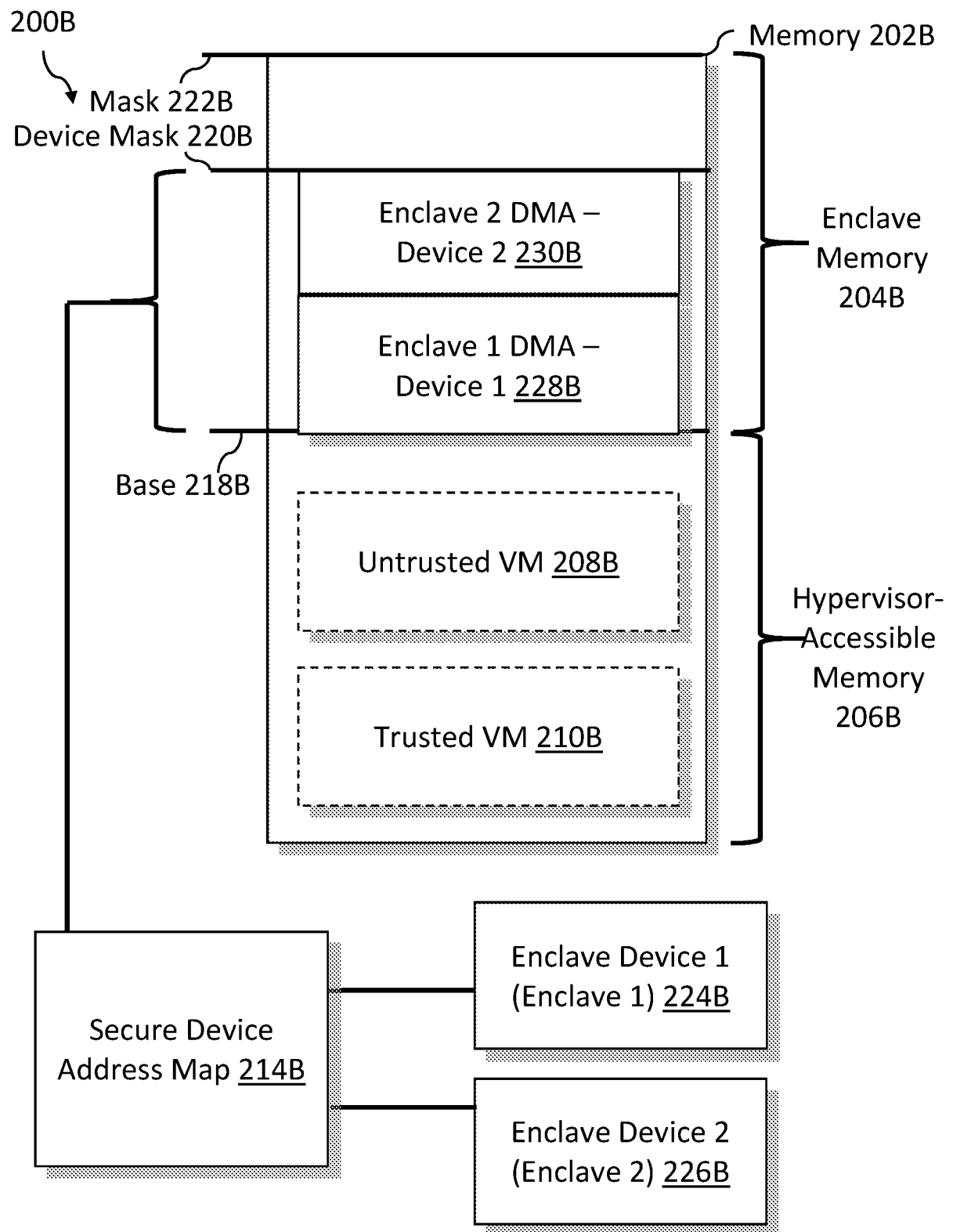


FIG. 2B

4 / 7

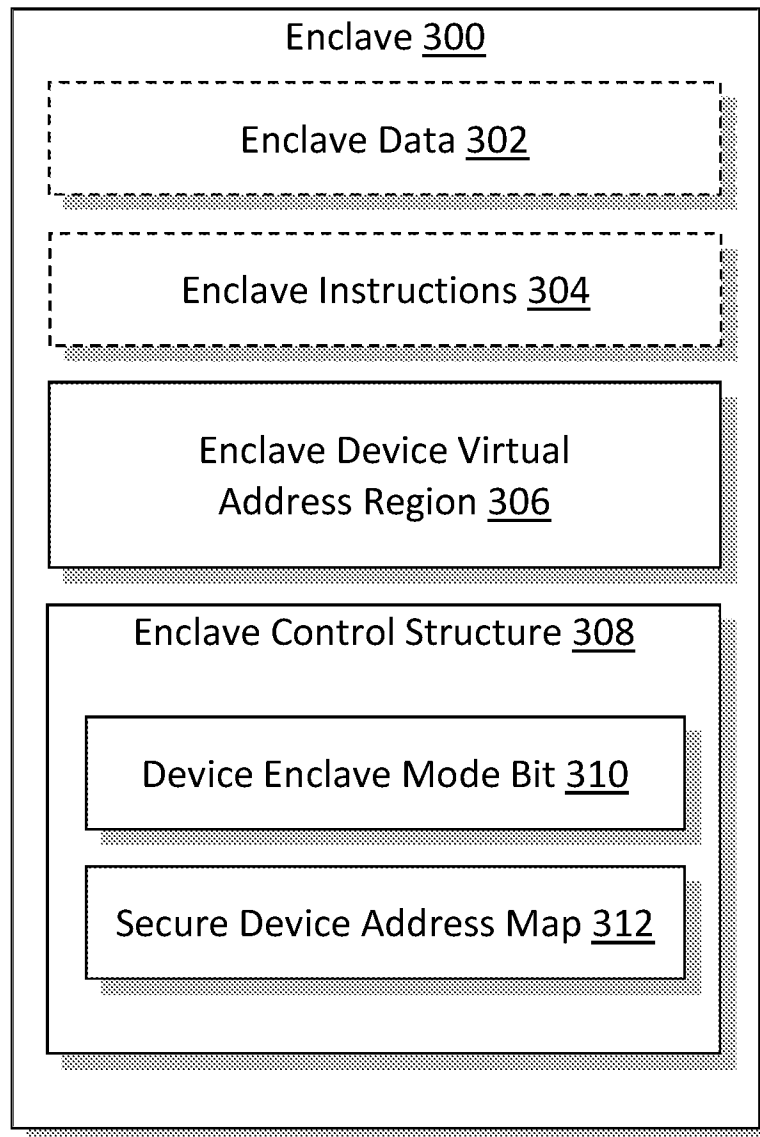


FIG. 3

5 / 7

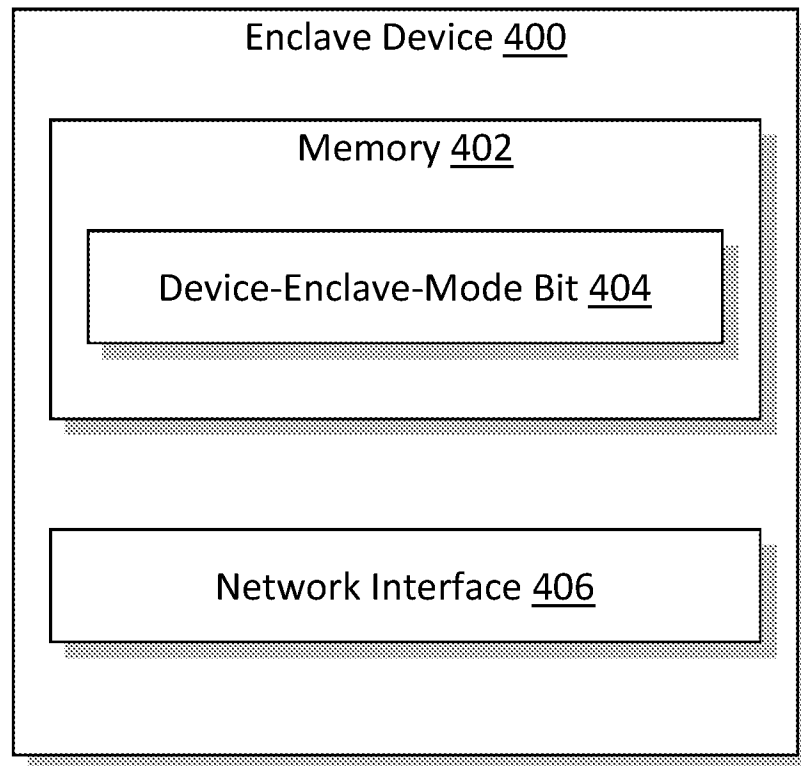


FIG. 4

6 / 7

500

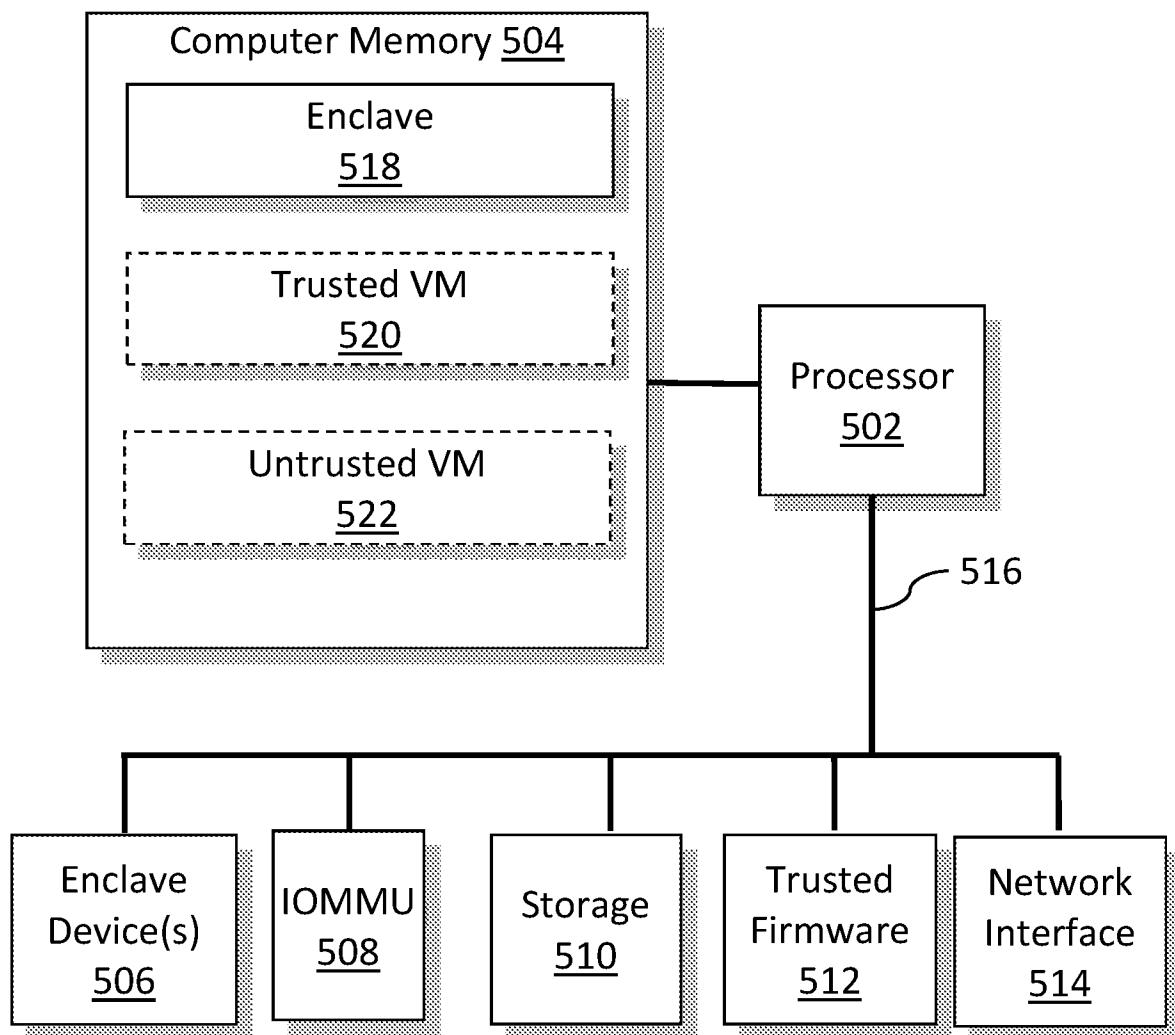


FIG. 5

7 / 7

600

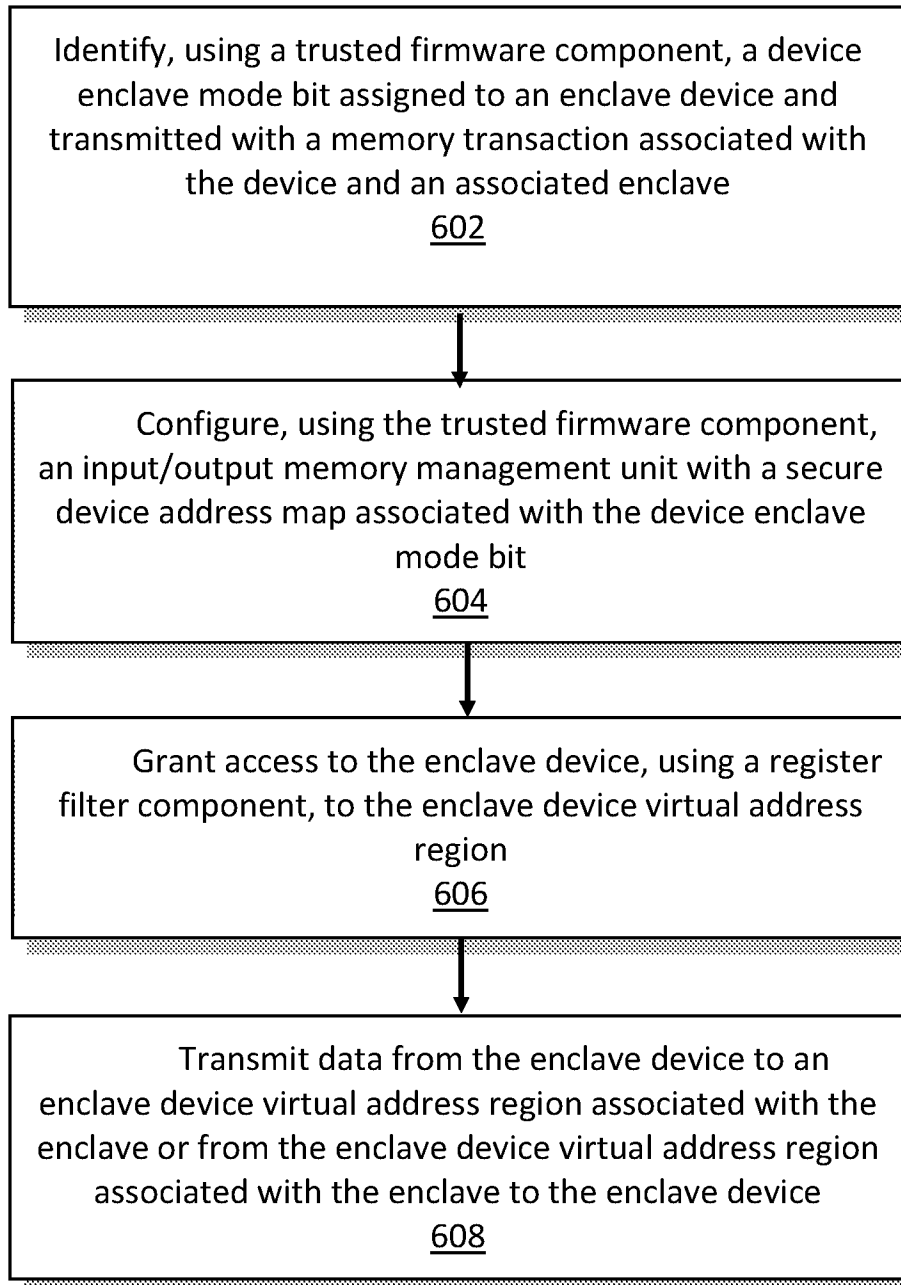


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/063764**A. CLASSIFICATION OF SUBJECT MATTER****G06F 21/50(2013.01)i, G06F 21/78(2013.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHEDMinimum documentation searched (classification system followed by classification symbols)
G06F 21/50; G06F 21/71; G06F 12/10; G06F 12/14; H04L 9/32; G06F 21/60; G06F 21/78Documentation 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 modelsElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: enclave device virtual address region, map, physical address, enclave device virtual address, grant, access, device enclave mode bit, secure device address map**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2014-0089617 A1 (APPLE INC.) 27 March 2014 See paragraphs [0031]-[0081], [0086]-[0097]; and figs. 1-5, 7-9.	1-18
A	US 2015-0186680 A1 (INTEL CORPORATION) 02 July 2015 See paragraphs [0033]-[0051]; and figs. 2, 3.	1-18
A	US 2014-0006711 A1 (BIN XING) 02 January 2014 See paragraphs [0018]-[0028]; and fig. 1.	1-18
A	US 2013-0198853 A1 (FRANCIS X. MCKEEN et al.) 01 August 2013 See paragraphs [0047]-[0059]; and fig. 5.	1-18
A	US 2015-0188710 A1 (SIMON JOHNSON et al.) 02 July 2015 See paragraphs [0020]-[0031]; and fig. 2.	1-18



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

15 March 2017 (15.03.2017)

Date of mailing of the international search report

16 March 2017 (16.03.2017)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

KANG, Hee Gok

Telephone No. +82-42-481-8264



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/063764

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0089617 A1	27/03/2014	US 8775757 B2	08/07/2014
US 2015-0186680 A1	02/07/2015	US 2012-163589 A1	28/06/2012
		US 2013-232345 A1	05/09/2013
		US 8832452 B2	09/09/2014
		US 9059855 B2	16/06/2015
		WO 2012-087562 A2	28/06/2012
		WO 2012-087562 A3	16/08/2012
US 2014-0006711 A1	02/01/2014	US 2016-054945 A1	25/02/2016
		US 9053042 B2	09/06/2015
US 2013-0198853 A1	01/08/2013	US 2013-159726 A1	20/06/2013
		US 9087200 B2	21/07/2015
US 2015-0188710 A1	02/07/2015	EP 2889794 A2	01/07/2015
		EP 2889794 A3	05/08/2015
		KR 10-1653193 B1	01/09/2016
		KR 10-2015-0079405 A	08/07/2015