



(51) International Patent Classification:

G06F 9/48 (2006.01) *G06F 21/53* (2013.01)*G06F 9/50* (2006.01) *G06F 21/70* (2013.01)*G06F 21/74* (2013.01) *G06F 21/71* (2013.01)*G06F 21/52* (2013.01)

(21) International Application Number:

PCT/EP2022/081213

(22) International Filing Date:

09 November 2022 (09.11.2022)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

17/457,446 03 December 2021 (03.12.2021) US

(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).(71) Applicant (for MG only): **IBM DEUTSCHLAND GMBH** [DE/DE]; IBM-Allee 1, 71139 Ehningen (DE).(72) Inventors: **LICHTENAU, Cedric**; c/o IBM Deutschland Research & Development GmbH, Schoenaicher Strasse 220, 71032 Boeblingen (DE). **LANG, Jakob**; c/o IBM Deutschland Research & Development GmbH, Schoenaich-er Strasse 220, 71032 Boeblingen (DE). **PASCH, Eberhard**; c/o IBM Deutschland Research & Development GmbH, Schoenaicher Strasse 220, 71032 Boeblingen (DE). **BORNTRAEGER, Christian**; c/o IBM Deutschland Research & Development GmbH, Schoenaicher Strasse 220, 71032 Boeblingen (DE).(74) Agent: **VETTER, Svenja**; c/o IBM Deutschland GmbH, Patentwesen und Urheberrecht, IBM-Allee 1, 71139 Ehningen (DE).

(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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,

(54) Title: OPERATING A SECURE CODE SEGMENT ON A PROCESSOR CORE OF A PROCESSING UNIT

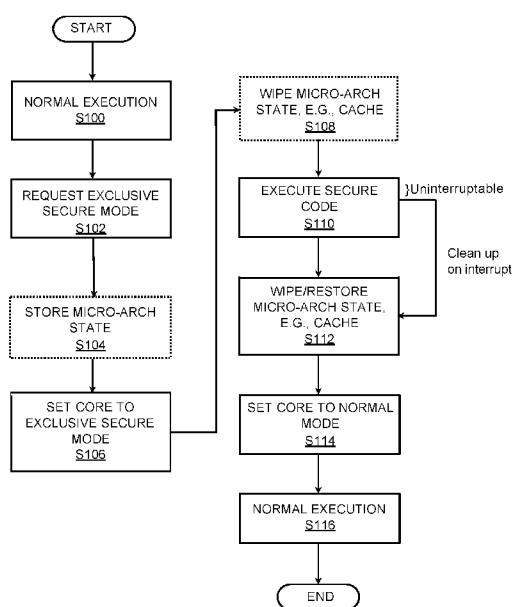


FIG. 1

(57) Abstract: A computer-implemented method, a computer system and a computer program product operate a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core. The method comprises requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core. The method also comprises setting the at least one processor core to exclusive secure execution for the secure code segment. The method further comprises executing the secure code segment on the at least one processor core uninterruptably. In addition, the method comprises wiping an architected state and a non-architected state of a physical processor core from the at least one processor core. Lastly, the method comprises setting the at least one processor core to the first execution mode for program code on the at least one processor core.

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, ME, 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))*
- *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

OPERATING A SECURE CODE SEGMENT ON A PROCESSOR CORE OF A PROCESSING UNIT

BACKGROUND

[0001] The present invention relates in general to data processing systems, in particular, to a computer implemented method, a computer system, a computer program product and a data processing system for operating a secure code segment on a processor core of a processing unit.

[0002] Multi-threaded processors may include a plurality of hardware threads that can execute concurrently. One example of a multi-threaded processor may be a simultaneous multithreaded (SMT) processor. In simultaneous multithreading, instructions from more than one thread can be executing in any given instruction pipeline stage at a time. Thus, simultaneous multithreading may allow multiple independent threads of execution to better utilize the resources provided by the processor architecture. Simultaneous multithreading may make each processor of a virtual machine appear as n-processors, called threads, to applications.

[0003] Processes or threads from different customers in the cloud or on premise may be virtualized on a virtual machine (VM) or on a container and may run on the same physical core. It is common to request for the thread to run in single-threaded mode.

[0004] One implementation of a pipeline for secure multithread execution may be an integrated circuit for executing instructions that includes: a processor pipeline configured to execute instructions from two or more threads in parallel using execution units of the processor pipeline; and a thread hazard circuitry configured to detect that an instruction of a first thread has been designated as a sensitive instruction, and, responsive to detection of the sensitive instruction, block instructions of threads other than the first thread from being executed using execution units of processor pipeline while the sensitive instruction is being executed by an execution unit of the processor pipeline.

[0005] In this implementation, just single instructions may be protected verifying that the architected state of a thread is isolated, whereas the non-architected state may not be considered.

[0006] Permanently reserving a core for all secure work can be done but this may effectively reduce the overall processor performance if not doing the whole-time secure

work. Further, it does not scale if a large portion of the work on the computer system requests a secure environment to execute.

SUMMARY

[0007] A computer-implemented method is proposed for operating a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core for running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core. The method at least comprises: (i) requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core; (ii) setting the at least one processor core to exclusive secure execution for the secure code segment; (iii) executing the secure code segment on the at least one processor core uninterruptably; (iv) wiping an architected state and a non-architected state of a physical processor core from the at least one processor core; (v) setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

[0008] According to the proposed method, a secure code segment may request an exclusive secure context for at least a minimum time and a minimum count of operating instructions. This comprises, under hardware control, at least securely setting the physical core to an exclusive secure mode, executing without interruption for a minimum time and/or number of operating instructions and finally wiping out at least an architected and non-architected state of the physical core from a cache storage or a register or a memory, when finishing the secure code segment or taking any interruption after the initial period of time and/or number of operating instructions.

[0009] Advantageously, interferences between different code segments in the processor core may be avoided. A number of failures during execution of the code segments may be reduced. Further, a performance of a computer system may favorably be enhanced by the proposed method of secure operation of a secure code segment in the processor core.

[0010] Due to an embodiment of the invention, additionally or alternatively, at least the architected and non-architected state of the physical processor core may be saved before setting the processor core to an exclusive secure mode. Thus, it is possible to restore, after finishing the execution of the secure code segment, the architected and non-architected state of the physical processor core in order to continue normal execution of program code.

[0011] Due to an embodiment of the invention, additionally or alternatively, at least the architected and non-architected state of the physical processor core may be restored before setting the processor core to the first execution mode. Thus, after finishing the execution of the secure code segment, the architected and non-architected state of the physical processor core may be restored in order to continue normal execution of program code.

[0012] Due to an embodiment of the invention, additionally or alternatively, the non-architected state of the physical processor core may be wiped, in particular from at least one cache storage of the physical processor core, before executing the secure code segment. By this way, no interference with already existing data may occur if the secure code segment is executed.

[0013] Due to an embodiment of the invention, additionally or alternatively, wiping the non-architected state of the physical processor core may be performed in a hardware-controlled manner by hardware of the processing unit. Favorably, the risk of software failures followed by insufficient wiping may be reduced.

[0014] Due to an embodiment of the invention, additionally or alternatively, the first execution mode of the processor core may be a simultaneous multithreading mode. Simultaneous multithreading allows multiple independent threads of execution to better utilize the resources provided by the processor architecture and is a normal execution mode of a usual processor core.

[0015] Due to an embodiment of the invention, additionally or alternatively, the uninterruptable execution of the secure code segment may be defined for a minimum period of time and/or a minimum number of instructions. Thus, an undisturbed execution of the secure code segment may be guaranteed for efficient processing.

[0016] Due to an embodiment of the invention, additionally or alternatively, the uninterruptable execution of the secure code segment may be controlled by a state machine. A state machine may prove as an efficient way of controlling the uninterruptable execution of the secure code segment.

[0017] Due to an embodiment of the invention, additionally or alternatively, an interrupt may be blocked during execution of the secure code segment. Favorably, the secure code segment may be executed without disturbance for a minimum amount of time and/or number of operating instructions.

[0018] Due to an embodiment of the invention, additionally or alternatively, wiping the architected and non-architected state from the at least one processor core after execution of the secure code segment may be initiated by an interrupt request. Thus, further normal execution of program code in the first execution state may be achieved without interference with remaining data from an execution of the secure code segment.

[0019] Due to an embodiment of the invention, additionally or alternatively, operating instructions may be implemented on a virtual machine or on a container. Processes/threads running on the same physical core from different customers in the cloud or on premise may favorably be virtualized on a virtual machine or on a container.

[0020] Further, a computer system is proposed for operating a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core. The computer system comprises at least one processor unit with at least one processor core, the processor core being configured to running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core. The computer system is configured to perform a method, at least comprising: (i) requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core; (ii) setting the at least one processor core to exclusive secure execution for the secure code segment; (iii) executing the secure code segment on the at least one processor core uninterruptably; (iv) wiping an architected state and a non-architected state of a physical processor core from the at least one processor core; (v) setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

[0021] Running on the proposed computer system, a secure code segment may request an exclusive secure context for at least a minimum time and/or a minimum count of operating instructions. This comprises, under hardware control, at least securely setting the physical core to an exclusive secure mode, executing without interruption for a minimum time and/or number of operating instructions and finally wiping out at least an architected and non-architected state from a cache storage or a register or a memory of the physical core when finishing the secure code segment or taking any interruption after the initial period of time and/or number of operating instructions.

[0022] Advantageously, interferences between different code segments in the processor core may be avoided. A number of failures during execution of the code segments may be

reduced. Further a performance of a computer system may favorably be enhanced by the proposed method of secure operation of a secure code segment in the processor core.

[0023] Due to an embodiment of the invention, additionally or alternatively, the computer system may be configured to save at least the architected and non-architected state of the physical processor core before setting the processor core to an exclusive secure mode. Thus, it is possible to restore, after finishing the execution of the secure code segment, the architected and non-architected state of the physical processor core in order to continue normal execution of program code.

[0024] Due to an embodiment of the invention, additionally or alternatively, the computer system may be configured to restore at least the architected and non-architected state of the physical processor core before the processor core is set to the first execution mode. Thus, after finishing the execution of the secure code segment the architected and non-architected state of the physical processor core may be restored in order to continue normal execution of program code.

[0025] Due to an embodiment of the invention, additionally or alternatively, the computer system may be configured to wipe the non-architected state of the physical processor core, in particular from at least one cache storage of the physical processor core, before the secure code segment is executed. By this way, no interference with already existing data may occur if the secure code segment is executed.

[0026] Due to an embodiment of the invention, additionally or alternatively, the computer system may be configured to wipe the non-architected state of the physical processor core in a hardware-controlled manner by hardware of the processing unit. Favorably, the risk of software failures followed by insufficient wiping may be reduced.

[0027] Due to an embodiment of the invention, additionally or alternatively, the first execution mode of the processor core may be a simultaneous multithreading mode. Simultaneous multithreading allows multiple independent threads of execution to better utilize the resources provided by the processor architecture and is a normal execution mode of a usual processor core.

[0028] Due to an embodiment of the invention, additionally or alternatively, the computer system may be configured to execute the secure code segment uninterruptably for a minimum period of time and/or a minimum number of instructions. Thus, an undisturbed execution of the secure code segment may be guaranteed for efficient processing.

[0029] Due to an embodiment of the invention, additionally or alternatively, the uninterruptable execution of the secure code segment may be controlled by a state machine. A state machine may prove as an efficient way of controlling the uninterruptable execution of the secure code segment.

[0030] Due to an embodiment of the invention, additionally or alternatively, the computer system may be configured to block an interrupt during execution of the secure code segment. Favorably, the secure code segment may be executed without disturbance for a minimum amount of time and/or number of operating instructions.

Due to an embodiment of the invention, additionally or alternatively, wiping the architected and non-architected state from the at least one processor core after execution of the secure code segment may be initiated by an interrupt request. Thus, further normal execution of program code in the first execution state may be achieved without interference with remaining data from an execution of the secure code segment.

[0031] Due to an embodiment of the invention, additionally or alternatively, operating instructions may be implemented on a virtual machine or on a container. Processes/threads running on the same physical core from different customers in the cloud or on premise may favorably be virtualized on a virtual machine or on a container.

[0032] Further, a favorable computer program product is proposed for operating a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core. The computer system comprises at least one processor unit with at least one processor core, the processor core being configured to running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core.

[0033] The computer program product comprises a computer readable storage medium having program instructions embodied therewith, the program instructions executable by the computer system to cause the computer system to perform a method comprising: (i) requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core; (ii) setting the at least one processor core to exclusive secure execution for the secure code segment; (iii) executing the secure code segment on the at least one processor core uninterruptably; (iv) wiping an architected state and a non-architected state of a physical processor core from the at least one processor core; (v) setting the at least

one processor core to the first execution mode for executing program code on the at least one processor core.

[0034] Further, a data processing system for execution of a data processing program is proposed, comprising computer readable program instructions for performing the method described above.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0035] The present invention together with the above-mentioned and other objects and advantages may best be understood from the following detailed description of the embodiments, but not restricted to the embodiments.

[0036] Figure 1 depicts a flow chart of a computer implemented method for operating a secure code segment on a processor core of a processing unit according to an embodiment of the invention.

[0037] Figure 2 depicts an example embodiment of a data processing system for executing a method according to the invention.

DETAILED DESCRIPTION

[0038] In the drawings, like elements are referred to with equal reference numerals. The drawings are merely schematic representations, not intended to portray specific parameters of the invention. Moreover, the drawings are intended to depict only typical embodiments of the invention and therefore should not be considered as limiting the scope of the invention.

[0039] The illustrative embodiments described herein provide a computer system for operating a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core. The computer system comprises at least one processor unit with at least one processor core, the processor core being configured to running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core.

[0040] The illustrative embodiments may further be used for a method, at least comprising: (i) requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core; (ii) setting the at least one processor core to

exclusive secure execution for the secure code segment; (iii) executing the secure code segment on the at least one processor core uninterruptably; (iv) wiping an architected and non-architected state from the at least one processor core; (v) setting the at least one processor core to the first execution mode for program code on the at least one processor core.

[0041] Figure 1 depicts a flow chart of a computer implemented method for operating a secure code segment on a processor core of a processing unit 216, as depicted in Figure 2, according to an embodiment of the invention.

[0042] The processing unit 216 is configured with at least one processor core for running a first execution mode for executing program code on the at least one processor core as a normal execution.

[0043] The first execution mode of the processor core may be a simultaneous multithreading mode. Simultaneous multithreading allows multiple independent threads of execution to better utilize the resources provided by the processor architecture and is a normal execution mode of a usual processor core.

[0044] Operating instructions may be implemented on a virtual machine or on a container. Processes/threads running on the same physical core from different customers in the cloud or on premise may favorably be virtualized on a virtual machine or on a container.

[0045] In step S100 a program code of operating instructions is executed on the at least one processor core in the first execution mode.

[0046] In step S102 an exclusive secure mode is requested for exclusive secure execution of a secure code segment of the program code on the at least one processor core.

[0047] Optionally, in step S104, at least the architected and non-architected state of the physical processor core may be saved before setting the processor core to the exclusive secure mode. Thus, it is possible to restore, after finishing the execution of the secure code segment, the architected and non-architected state of the physical processor core in order to continue normal execution of program code.

[0048] Next, in step S106, the at least one processor core is set to exclusive secure execution for the secure code segment.

[0049] Also optionally, in step S108, the non-architected state of the physical processor core may be wiped before executing the secure code segment. In particular, the non-

architected state may be wiped from the at least one cache storage or from a register or from a memory of the physical processor core. Wiping the non-architected state of the physical processor core may be performed in a hardware-controlled manner by hardware of the processing unit 216. Favorably, the risk of software failures followed by insufficient wiping may be reduced. By this way, no interference with already existing data may occur if the secure code segment is executed.

[0050] Then, in step S110, the secure code segment is executed on the at least one processor core. The secure code segment is executed uninterruptably for a minimum period of time and/or a minimum number of instructions. The uninterruptable execution of the secure code segment may be controlled by a state machine. A state machine may prove as an efficient way of controlling the uninterruptable execution of the secure code segment.

[0051] Any interrupt may be blocked during execution of the secure code segment. Thus, an undisturbed execution of the secure code segment may be guaranteed for efficient processing. Favorably, the secure code segment may be executed without disturbance for a minimum amount of time and/or number of operating instructions.

[0052] After finishing the execution of the secure code segment, in step S112, an architected and non-architected state may be wiped, in particular from at least one cache storage or from a register or from a memory of the at least one processor core. Wiping the architected and non-architected state from the at least one processor core after execution of the secure code segment may be initiated by an interrupt request. Thus, further normal execution of program code in the first execution state may be achieved without interference with remaining data from an execution of the secure code segment.

[0053] Optionally, in step S112, at least the architected and non-architected state of the physical processor core may be restored before setting the processor core to the first execution mode, if the architected and non-architected state was saved in step S104. Thus, after finishing the execution of the secure code segment the architected and non-architected state of the physical processor core may be restored in order to continue normal execution of program code.

[0054] Then, in step S114, the at least one processor core is set to the first execution mode for program code on the at least one processor core, in order to continue normal execution in step S116.

[0055] Referring now to Figure 2, a schematic of an example of a data processing system 210 is shown. Data processing system 210 is only one example of a suitable data processing system and is not intended to suggest any limitation as to the scope of use or functionality of embodiments of the invention described herein. Regardless, data processing system 210 is capable of being implemented and/or performing any of the functionality set forth herein above.

[0056] In data processing system 210 there is a computer system/server 212, which is operational with numerous other general-purpose or special-purpose computing system environments or configurations. Examples of well-known computing systems, environments, and/or configurations that may be suitable for use with computer system/server 212 include, but are not limited to, personal computer systems, server computer systems, thin clients, thick clients, handheld or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputer systems, mainframe computer systems, and distributed cloud computing environments that include any of the above systems or devices, and the like.

[0057] Computer system/server 212 may be described in the general context of computer system executable instructions, such as program modules, being executed by a computer system. Generally, program modules may include routines, programs, objects, components, logic, data structures, and so on that perform particular tasks or implement particular abstract data types. Computer system/server 212 may be practiced in distributed cloud computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed cloud computing environment, program modules may be located in both local and remote computer system storage media including memory storage devices.

[0058] As shown in Fig. 2, computer system/server 212 in data processing system 210 is shown in the form of a general-purpose computing device. The components of computer system/server 212 may include, but are not limited to, one or more processors or processing units 216, a system memory 228, and a bus 218 that couples various system components including system memory 228 to processing unit 216.

[0059] Bus 218 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro

Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus.

[0060] Computer system/server 212 typically includes a variety of computer system readable media. Such media may be any available media that is accessible by computer system/server 212, and it includes both volatile and non-volatile media, removable and non-removable media.

[0061] System memory 228 can include computer system readable media in the form of volatile memory, such as random-access memory (RAM) 230 and/or cache memory 232. Computer system/server 212 may further include other removable/non-removable, volatile/non-volatile computer system storage media. By way of example only, storage system 234 can be provided for reading from and writing to a non-removable, non-volatile magnetic media (not shown and typically called a "hard drive"). Although not shown, a magnetic disk drive for reading from and writing to a removable, non-volatile magnetic disk (e.g., a "floppy disk"), and an optical disk drive for reading from or writing to a removable, non-volatile optical disk such as a CD-ROM, DVD-ROM or other optical media can be provided. In such instances, each can be connected to bus 218 by one or more data media interfaces. As will be further depicted and described below, memory 228 may include at least one program product having a set (e.g., at least one) of program modules that are configured to carry out the functions of embodiments of the invention.

[0062] Program/utility 240, having a set (at least one) of program modules 242, may be stored in memory 228 by way of example, and not limitation, as well as an operating system, one or more application programs, other program modules, and program data. Each of the operating system, one or more application programs, other program modules, and program data or some combination thereof, may include an implementation of a networking environment. Program modules 242 generally carry out the functions and/or methodologies of embodiments of the invention as described herein.

[0063] Computer system/server 212 may also communicate with one or more external devices 214 such as a keyboard, a pointing device, a display 224, etc.; one or more devices that enable a user to interact with computer system/server 212; and/or any devices (e.g., network card, modem, etc.) that enable computer system/server 212 to communicate with one or more other computing devices. Such communication can occur via Input/Output (I/O) interfaces 222. Still yet, computer system/server 212 can communicate with one or more networks such as a local area network (LAN), a general wide area network (WAN), and/or a

public network (e.g., the Internet) via network adapter 220. As depicted, network adapter 220 communicates with the other components of computer system/server 212 via bus 218. It should be understood that although not shown, other hardware and/or software components could be used in conjunction with computer system/server 212. Examples, include, but are not limited to: microcode, device drivers, redundant processing units, external disk drive arrays, RAID systems, tape drives, and data archival storage systems, etc.

[0064] The present invention may be a system, a method, and/or a computer program product. The computer program product may include a computer readable storage medium (or media) having computer readable program instructions thereon for causing a processor to carry out aspects of the present invention.

[0065] The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-exhaustive list of more specific examples of the computer readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a static random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing. A computer readable storage medium, as used herein, is not to be construed as being transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire.

Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network adapter card or network interface in each computing/processing device receives computer readable program

instructions from the network and forwards the computer readable program instructions for storage in a computer readable storage medium within the respective computing/processing device.

Computer readable program instructions for carrying out operations of the present invention may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language such as Smalltalk, C++ or the like, and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present invention.

Aspects of the present invention are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions.

[0066] These computer readable program instructions may be provided to a processor of a general-purpose computer, special-purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks. These computer readable program instructions may also be stored in a computer

readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a particular manner, such that the computer readable storage medium having instructions stored therein comprises an article of manufacture including instructions which implement aspects of the function/act specified in the flowchart and/or block diagram block or blocks.

[0067] The computer readable program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other device to cause a series of operational steps to be performed on the computer, other programmable apparatus or other device to produce a computer implemented process, such that the instructions which execute on the computer, other programmable apparatus, or other device implement the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0068] The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). In some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special-purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special-purpose hardware and computer instructions.

[0069] The descriptions of the various embodiments of the present invention have been presented for purposes of illustration but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments disclosed herein.

[0070] Further exemplary embodiments of the present disclosure are set out in the following numbered clauses:

[0071] Numbered clause 1: A computer implemented method for operating a secure code segment on a processor core of a processing unit (216), wherein the processing unit (216) is configured with at least one processor core for running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core, the method at least comprising:

- (i) requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core;
- (ii) setting the at least one processor core to exclusive secure execution for the secure code segment;
- (iii) executing the secure code segment on the at least one processor core uninterruptably;
- (iv) wiping an architected state and a non-architected state of a physical processor core from the at least one processor core;
- (v) setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

[0072] Numbered clause 2: The method according to numbered clause 1, further saving at least the architected state and the non-architected state of the physical processor core before setting the processor core to an exclusive secure mode.

[0073] Numbered clause 3: The method according to numbered clause 1 or 2, further restoring at least the architected state and the non-architected state of the physical processor core before setting the processor core to the first execution mode.

[0074] Numbered clause 4: The method according to any one of numbered clauses 1 to 3, further wiping the non-architected state of the physical processor core from at least one cache storage of the physical processor core before executing the secure code segment.

[0075] Numbered clause 5: The method according to numbered clause 4, wherein wiping the non-architected state of the physical processor core is performed in a hardware-controlled manner by hardware of the processing unit (216).

[0076] Numbered clause 6: The method according to any one of numbered clauses 1 to 5, wherein the first execution mode for executing program code on the processor core is a simultaneous multithreading mode.

[0077] Numbered clause 7: The method according to any one of numbered clauses 1 to 6, wherein executing the secure code segment on the at least one processor core uninterruptably is defined for a minimum period of time and/or a minimum number of instructions.

[0078] Numbered clause 8: The method according to any one of numbered clauses 1 to 7, wherein executing the secure code segment on the at least one processor core uninterruptably is controlled by a state machine.

Numbered clause 9: The method according to any one of numbered clauses 1 to 8, wherein an interrupt is blocked during execution of the secure code segment.

[0079] Numbered clause 10: The method according to any one of numbered clauses 1 to 9, wherein wiping the architected state and the non-architected state of the physical processor core from the at least one processor core after execution of the secure code segment is initiated by an interrupt request.

[0080] Numbered clause 11: The method according to any one of numbered clauses 1 to 10, wherein operating instructions are implemented on a virtual machine or on a container.

[0081] Numbered clause 12: A computer system (212) for operating a secure code segment on a processor core of a processing unit (216), wherein the processing unit (216) is configured with at least one processor core, the computer system (212) comprising at least one processor unit (216) with at least one processor core, the processor core being configured to running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core, wherein the computer system (212) is configured to perform a method according to any one of numbered clauses 1 to 11, at least comprising:

- (i) requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core;
- (ii) setting the at least one processor core to exclusive secure execution for the secure code segment;
- (iii) executing the secure code segment on the at least one processor core uninterruptably;
- (iv) wiping an architected state and a non-architected state of a physical processor core from the at least one processor core;
- (v) setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

[0082] Numbered clause 13: The computer system according to numbered clause 12, being configured to save at least the architected state and the non-architected state of the physical processor core before setting the processor core to an exclusive secure mode.

[0083] Numbered clause 14: The computer system according to numbered clause 12 or 13, being configured to restore at least the architected state and the non-architected state of the physical processor core before the processor core is set to the first execution mode.

[0084] Numbered clause 15: The computer system according to any one of numbered clauses 12 to 14, being configured to wipe the non-architected state of the physical processor core from at least one cache storage of the physical processor core before the secure code segment is executed.

[0085] Numbered clause 16: The computer system according to any one of numbered clauses 12 to 15, being configured to wipe the non-architected state of the physical processor core in a hardware-controlled manner by hardware of the processing unit.

[0086] Numbered clause 17: The computer system according to any one of numbered clauses 12 to 16, wherein the first execution mode for executing program code on the processor core is a simultaneous multithreading mode.

[0087] Numbered clause 18: The computer system according to any one of numbered clauses 12 to 17, being configured to execute the secure code segment on the at least one processor core uninterruptably for a minimum period of time and/or a minimum number of instructions.

[0088] Numbered clause 19: The computer system according to any one of numbered clauses 12 to 18, wherein executing the secure code segment on the at least one processor core uninterruptably is controlled by a state machine.

[0089] Numbered clause 20: The computer system according to any one of numbered clauses 12 to 19, being configured to block an interrupt during execution of the secure code segment.

[0090] Numbered clause 21: The computer system according to any one of numbered clauses 12 to 20, wherein wiping the architected state and the non-architected state of the physical processor core from the at least one processor core after execution of the secure code segment is initiated by an interrupt request.

[0091] Numbered clause 22: The computer system according to any one of numbered clauses 12 to 21, wherein operating instructions are implemented on a virtual machine or on a container.

[0092] Numbered clause 23: A computer program product for operating a secure code segment on a processor core of a processing unit (216), wherein the processing unit (216) is configured with at least one processor core for running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core, the computer program product comprising a computer readable storage medium having program instructions embodied therewith, the program instructions executable by the computer system (212) to cause the computer system (212) to perform a method according to any one of numbered clauses 1 to 11 comprising:

- (i) requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core;
- (ii) setting the at least one processor core to exclusive secure execution for the secure code segment;
- (iii) executing the secure code segment on the at least one processor core uninterruptably;
- (iv) wiping an architected state and a non-architected state of a physical processor core from the at least one processor core;
- (v) setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

[0093] Numbered clause 24: A data processing system (210) for execution of a data processing program (240) comprising computer readable program instructions for performing a method according to any one of numbered clauses 1 to 11.

[0094] Numbered clause 25: An integrated circuit for executing instructions for operating a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core, the computer system comprising at least one processor unit with at least one processor core, the processor core being configured to running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core, wherein the integrated circuit is configured to perform a method according to any one of numbered clauses 1 to 11, at least comprising:

- (i) requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core;
- (ii) setting the at least one processor core to exclusive secure execution for the secure code segment;
- (iii) executing the secure code segment on the at least one processor core uninterruptably;
- (iv) wiping an architected state and a non-architected state of a physical processor core from the at least one processor core; and
- (v) setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

CLAIMS

1. A computer-implemented method for operating a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core for running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core, the method at least comprising:

requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core;

setting the at least one processor core to exclusive secure execution for the secure code segment;

executing the secure code segment on the at least one processor core uninterruptably;

wiping an architected state and a non-architected state of a physical processor core from the at least one processor core; and

setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

2. The computer-implemented method of claim 1, further comprising:

saving at least the architected state and the non-architected state of the physical processor core before setting the processor core to an exclusive secure mode.

3. The computer-implemented method of claim 1 or 2, further comprising:

restoring at least the architected state and the non-architected state of the physical processor core before setting the processor core to the first execution mode.

4. The computer-implemented method of any one of the claims 1 to 3, further comprising:

wiping the non-architected state of the physical processor core from at least one cache storage of the physical processor core before executing the secure code segment.

5. The computer-implemented method of claim 4, wherein wiping the non-architected state of the physical processor core is performed in a hardware-controlled manner by hardware of the processing unit.
6. The computer-implemented method of any one of the claims 1 to 5, wherein the first execution mode for executing program code on the processor core is a simultaneous multithreading mode.
7. The computer-implemented method of any one of the claims 1 to 6, wherein executing the secure code segment on the at least one processor core uninterruptably is defined for a minimum period of time and a minimum number of instructions.
8. The computer-implemented method of any one of the claims 1 to 7, wherein executing the secure code segment on the at least one processor core uninterruptably is controlled by a state machine.
9. The computer-implemented method of any one of the claims 1 to 8, wherein an interrupt is blocked during execution of the secure code segment.
10. The computer-implemented method of any one of the claims 1 to 9, wherein wiping the architected state and the non-architected state of the physical processor core from the at least one processor core after execution of the secure code segment is initiated by an interrupt request.
11. The computer-implemented method of any one of the claims 1 to 10, wherein operating instructions are implemented on a virtual machine or on a container.
12. A computer system for operating a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core,

the computer system comprising at least one processor unit with at least one processor core, the processor core being configured to running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core, wherein the computer system is configured to perform a method, at least comprising:

requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core;

setting the at least one processor core to exclusive secure execution for the secure code segment;

executing the secure code segment on the at least one processor core uninterruptably;

wiping an architected state and a non-architected state of a physical processor core from the at least one processor core; and

setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

13. The computer system of claim 12, being configured to save at least the architected state and the non-architected state of the physical processor core before setting the processor core to an exclusive secure mode.

14. The computer system of claim 12 or 13, being configured to restore at least the architected state and the non-architected state of the physical processor core before the processor core is set to the first execution mode.

15. The computer system of any one of the claims 12 to 14, being configured to wipe the non-architected state of the physical processor core from at least one cache storage of the physical processor core before the secure code segment is executed.

16. The computer system of claim 15, being configured to wipe the non-architected state of the physical processor core in a hardware-controlled manner by hardware of the processing unit.

17. The computer system of any one of the claims 12 to 16, wherein the first execution mode for executing program code on the processor core is a simultaneous multithreading mode.

18. The computer system of any one of the claims 12 to 17, being configured to execute the secure code segment on the at least one processor core uninterruptably for a minimum period of time and a minimum number of instructions.

19. The computer system of any one of the claims 12 to 18, wherein the executing of the secure code segment on the at least one processor core uninterruptably is controlled by a state machine.

20. The computer system of any one of the claims 12 to 19, being configured to block an interrupt during execution of the secure code segment.

21. The computer system of any one of the claims 12 to 20, wherein wiping the architected state and the non-architected state of the physical processor core from the at least one processor core after execution of the secure code segment is initiated by an interrupt request.

22. The computer system of any one of the claims 12 to 21, wherein operating instructions are implemented on a virtual machine or on a container.

23. A computer program product for operating a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core for running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core,

the computer program product comprising a computer readable storage medium having program instructions embodied therewith, the program instructions executable by a computer system to cause the computer system to perform a method comprising:

requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core;

setting the at least one processor core to exclusive secure execution for the secure code segment;

executing the secure code segment on the at least one processor core uninterruptably;

wiping an architected state and a non-architected state of a physical processor core from the at least one processor core; and

setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

24. A data processing system for execution of a data processing program comprising computer readable program instructions for performing the computer-implemented method of claim 1.

25. An integrated circuit for executing instructions for operating a secure code segment on a processor core of a processing unit, wherein the processing unit is configured with at least one processor core, the integrated circuit comprising at least one processor unit with at least one processor core, the processor core being configured to running a first execution mode for executing program code on the at least one processor core and executing a program code of operating instructions on the at least one processor core, wherein the integrated circuit is configured to perform a method, at least comprising:

requesting exclusive secure execution of a secure code segment of the program code on the at least one processor core;

setting the at least one processor core to exclusive secure execution for the secure code segment;

executing the secure code segment on the at least one processor core uninterruptably;

wiping an architected state and a non-architected state of a physical processor core from the at least one processor core; and

setting the at least one processor core to the first execution mode for executing program code on the at least one processor core.

1/2

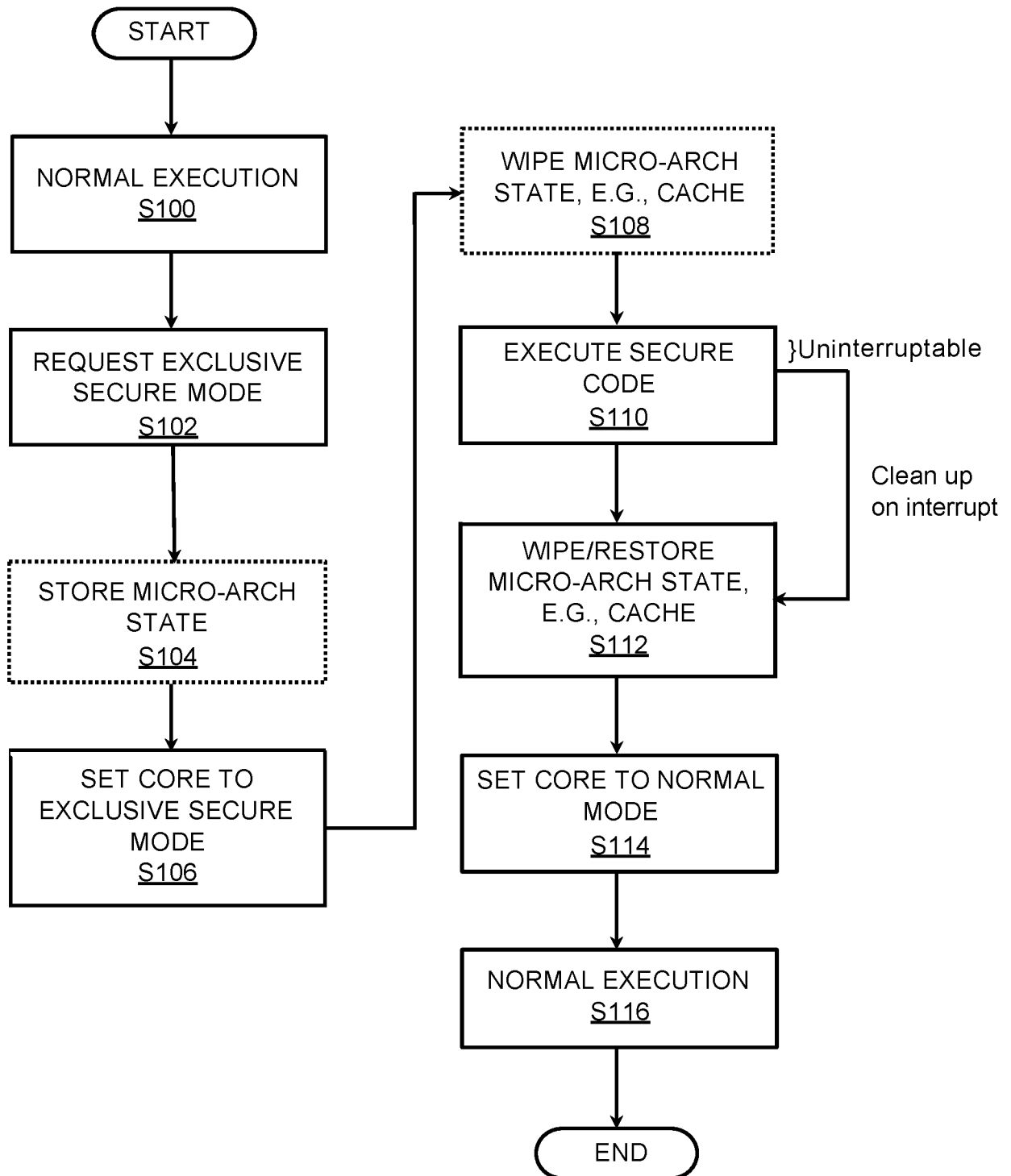


FIG. 1

2/2

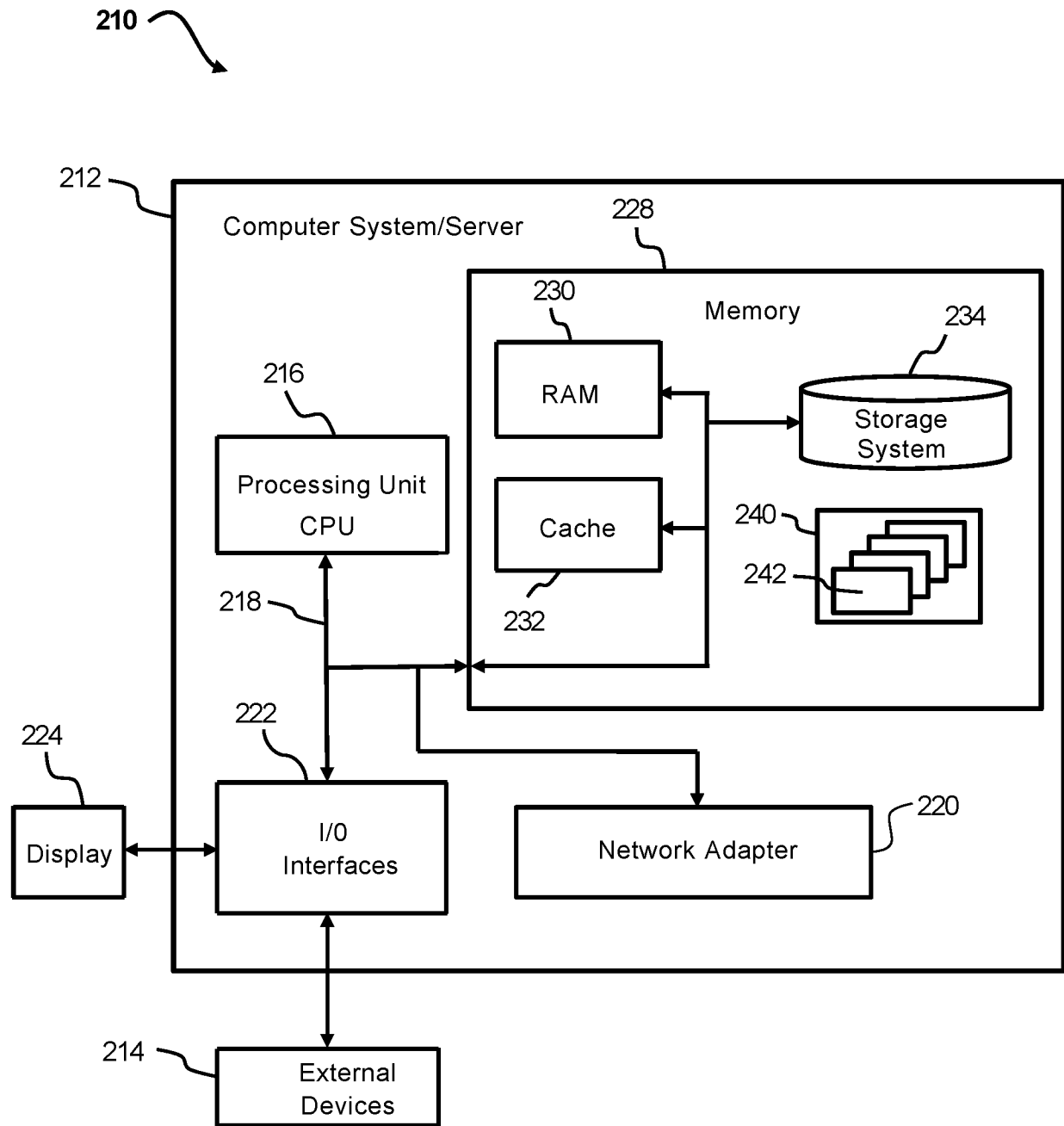


FIG. 2

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2022/081213

A. CLASSIFICATION OF SUBJECT MATTER		
INV.	G06F9/48	G06F9/50
		G06F21/74
ADD.	G06F21/52	G06F21/53
		G06F21/70
		G06F21/71
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		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
EPO-Internal, WPI Data		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 3 462 362 A1 (ST MICROELECTRONICS INC [US]) 3 April 2019 (2019-04-03) paragraph [0015] paragraph [0018] paragraph [0020] paragraph [0028] paragraph [0030] - paragraph [0032] paragraph [0077] paragraph [0085] - paragraph [0087] paragraph [0113] claim 1 paragraph [0109]	1-25
A	US 2014/230077 A1 (MUFF ADAM J [US] ET AL) 14 August 2014 (2014-08-14) the whole document	1-25
----- -/--		
☒ 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		Date of mailing of the international search report
31 January 2023		08/02/2023
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Alecu, Mihail

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2022/081213

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 112 800 431 A (ALIPAY HANGZHOU TECH CO LTD) 14 May 2021 (2021-05-14) the whole document & EP 3 961 446 A1 (ALIPAY HANGZHOU INF TECH CO LTD [CN]) 2 March 2022 (2022-03-02) the whole document -----	1-25
A	US 2019/034357 A1 (NUNEZ MENCINAS ANGEL [DE] ET AL) 31 January 2019 (2019-01-31) paragraph [0020] -----	1-25

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2022/081213

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 3462362 A1	03-04-2019	CN 109582354 A	05-04-2019
		EP 3462362 A1	03-04-2019
		US 2019102538 A1	04-04-2019
		US 2020380116 A1	03-12-2020

US 2014230077 A1	14-08-2014	US 2014229690 A1	14-08-2014
		US 2014230077 A1	14-08-2014

CN 112800431 A	14-05-2021	CN 111753311 A	09-10-2020
		CN 112800431 A	14-05-2021
		EP 3961446 A1	02-03-2022
		US 2022066809 A1	03-03-2022

US 2019034357 A1	31-01-2019	CN 110998545 A	10-04-2020
		DE 112018002947 T5	02-04-2020
		GB 2578410 A	06-05-2020
		JP 6955619 B2	27-10-2021
		JP 2020528608 A	24-09-2020
		US 2019034356 A1	31-01-2019
		US 2019034357 A1	31-01-2019
		US 2020110712 A1	09-04-2020
		WO 2019021153 A1	31-01-2019
