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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 December 2016 (22.12.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/204863 A1

(51) International Patent Classification:
G06F 21/55 (2013.01) *G06F 15/78* (2006.01)
G06F 1/26 (2006.01)

(21) International Application Number:
PCT/US2016/029103

(22) International Filing Date:
25 April 2016 (25.04.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/740,639 16 June 2015 (16.06.2015) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: **POORNACHANDRAN, Rajesh**; 15317 NW Twoponds Dr, Portland, Oregon 97229 (US). **SMITH, Ned M.**; 375 SW Delta Drive, Beaverton, Oregon 97006 (US).

(74) Agent: **JORDAN, B. Delano**; Jordan IP Law LLC, c/o CPA Global, P.O. Box 52050, Minneapolis, Minnesota 55402 (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, 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, 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.

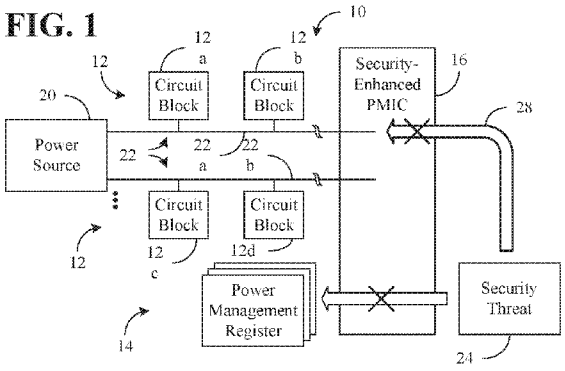
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR,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

(54) **Title**: ENHANCED SECURITY OF POWER MANAGEMENT COMMUNICATIONS AND PROTECTION FROM SIDE CHANNEL ATTACKS

FIG. 1



(57) **Abstract**: Systems, apparatuses and methods may provide for detecting a request to access a power management register and conducting, via a power management security interface, a runtime credential exchange with a source of the request. Additionally, the request may be denied if the runtime credential exchange is unsuccessful. In one example, a plug event is detected, via a dedicated side channel, with respect to a debug port. A noise analysis may be conducted of one or more power rails in response to the plug event, wherein policy based counter noise may be generated on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

WO 2016/204863 A1

ENHANCED SECURITY OF POWER MANAGEMENT COMMUNICATIONS AND PROTECTION FROM SIDE CHANNEL ATTACKS

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application claims the benefit of priority to U.S. Non-Provisional Patent Application No. 14/740,639 filed on June 16, 2015.

TECHNICAL FIELD

Embodiments generally relate to data security. More particularly, embodiments relate to enhanced security of power management communications and protection from side channel attacks.

BACKGROUND

System on chip (SoC) computing architectures may integrate a variety of circuit blocks (e.g., intellectual property/IP blocks) into a common silicon package, wherein different circuit blocks may perform different functions such as, for example, particular types of encryption processing. The circuit blocks, which may be connected to one or more power rails on the silicon package, may be independently placed in various power states in order to improve efficiency and/or reduce power consumption. Attackers of the computing system may attempt to use power management information to determine which circuit blocks are active at a given moment in time. For example, malware may attempt to access power management registers that store information regarding circuit block power states in an effort to identify the type of encryption processing being used. Additionally, physical access to the system through Joint Test Access Group (JTAG) debug ports may enable attackers to monitor the power rails and determine which circuit blocks (and underlying functions) are active.

BRIEF DESCRIPTION OF THE DRAWINGS

The various advantages of the embodiments will become apparent to one skilled in the art by reading the following specification and appended claims, and by referencing the following drawings, in which:

FIG. 1 is a block diagram of an example of a security threat solution according to an embodiment;

FIG. 2 is a flowchart of an example of a method of operating a security-enabled power management integrated circuit (PMIC) at boot time according to an embodiment;

FIG. 3 is a flowchart an example of a method of operating a PMIC at runtime according to an embodiment;

FIG. 4 is a block diagram of an example of a power management computing architecture according to an embodiment;

FIG. 5 is a block diagram of an example of a processor according to an embodiment; and

FIG. 6 is a block diagram of an example of a computing system according to an embodiment.

DESCRIPTION OF EMBODIMENTS

Turning now to FIG. 1, a security threat solution for a computing system such as a system on chip (SoC) 10 is shown. The SoC 10 may generally include a plurality of circuit blocks 12 (12a-12d), one or more power management registers 14 and a security-enhanced power management integrated circuit (PMIC) 16, wherein the circuit blocks 12 may be configured to perform various functions such as, for example, different types of encryption processing, communications, calculations, and so forth. For example, a first circuit block 12a may be configured to perform a first type of encryption/decryption on data processed by the SoC 10, a second circuit block 12b may be configured to perform a second type of encryption/decryption, a third circuit block 12c may be configured to perform a third type of encryption/decryption, etc. In the illustrated example, the circuit blocks 12 receive power from a power source 20 (e.g., battery, voltage regulator, adapter) via power rails 22 (22a, 22b).

As will be discussed in greater detail, the PMIC 16 may prevent various power management related attacks from a security threat 24 (e.g., attacker/intruder, malware). More particularly, the PMIC 16 may prevent/block one or more attempts 26 by the security threat 24 to access the power management registers 14 if, for example, the security threat 24 does not present appropriate credentials to access the power management registers 14. In one example, the PMIC 16 prevents/blocks the attempts 26 by using a power management security interface (e.g., architectural enclave, not shown) to establish a secure communication path between a host processor (not shown) and the power management registers 14. Additionally, the PMIC 16 may prevent/block

one or more attempts 28 by the security threat 24 to monitor the power rails 22 (e.g., wire-level probing) during operation of the SoC 10. In one example, the PMIC 16 prevents/blocks the attempts 28 by using a side channel (e.g., out-of-band channel, not shown) to monitor a debug port of the system. Accordingly, the illustrated solution enhances security by preventing the security threat 24 from maliciously determining which circuit blocks 12 are active at any given moment in time. As a result, attackers may be unable to identify, for example, the type of encryption, communication and/or calculations being used during operation of the SoC 10.

FIG. 2 shows a method 30 of operating a PMIC at boot time. The method 30 may generally be implemented in a PMIC such as, for example, the security-enabled PMIC 16 (FIG. 1), already discussed. More particularly, the method 30 may be implemented as a module or related component in a set of logic instructions stored in a non-transitory machine- or computer-readable storage medium such as random access memory (RAM), read only memory (ROM), programmable ROM (PROM), firmware, flash memory, etc., in configurable logic such as, for example, programmable logic arrays (PLAs), field programmable gate arrays (FPGAs), complex programmable logic devices (CPLDs), in fixed-functionality hardware logic using circuit technology such as, for example, application specific integrated circuit (ASIC), complementary metal oxide semiconductor (CMOS) or transistor-transistor logic (TTL) technology, or any combination thereof. For example, computer program code to carry out operations shown in the method 30 may be written in any combination of one or more programming languages, including an object oriented programming language such as JAVA, SMALLTALK, C++ or the like and conventional procedural programming languages, such as the "C" programming language or similar programming languages.

Illustrated processing block 32 provides for determining whether a boot event, for example, a power-on-self-test (POST) routine, reset request, and so forth, has been detected. If a boot event has been detected, processing block 34 may determine whether enhanced security is enabled. If so, illustrated processing block 36 conducts, in response to the boot event, a boot time credential exchange via a power management security interface (e.g, running on a host processor) and a trusted execution environment (TEE, e.g., a converged manageability security engine/CSME of INTEL Corporation, TRUSTZONE of ARM Holdings). The boot time credential exchange may involve the transfer of security credentials from the PMIC through the power management security interface to the trusted execution environment, which may in turn

determine the authenticity of the PMIC (e.g., through challenge/response attestations). If either a boot event is not detected at processing block 32 or it is determined that enhanced security is not enabled at processing block 34, the illustrated method terminates.

If it is determined at processing block 38 that the boot time credential exchange is successful (e.g., the credentials of the PMIC are verified), illustrated processing block 40 loads policy information from the trusted execution environment to local policy storage. The policy information may include, for example, action identifiers (e.g., indicating what actions to take in order to enforce a security policy), credential data (e.g., TEE certification), power thresholds (e.g., rail noise thresholds, voltage swing thresholds), and so forth. Processing block 40 may also provide for registering an interrupt service routine (ISR) with a dedicated side channel of a debug port such as, for example, a Joint Test Access Group (JTAG) debug port. Illustrated processing block 42 transfers control to the runtime security component. If it is determined at block 38 that the boot time credential exchange is unsuccessful (e.g., the credentials of the PMIC are not verified), illustrated processing block 39 enforces appropriate security policies such as, for example, generating an alert, logging the unsuccessful exchange, or other action.

FIG. 3 shows a method 44 of operating a PMIC at runtime. The method 44 may generally be implemented in a PMIC such as, for example, the security-enabled PMIC 16 (FIG. 1), already discussed. More particularly, the method 44 may be implemented as a module or related component in a set of logic instructions stored in a non-transitory machine- or computer-readable storage medium such as RAM, ROM, PROM, firmware, flash memory, etc., in configurable logic such as, for example, PLAs, FPGAs, CPLDs, in fixed-functionality hardware logic using circuit technology such as, for example, ASIC, CMOS or TTL technology, or any combination thereof.

Illustrated processing block 46 determines whether a request to access a power management register has been detected. The request may be received from an application (e.g., valid secure mode application instance/enclave, malware) running on, for example, a host processor, wherein the power management register may store power consumption information such as, for example, power state information (e.g., Advanced Configuration and Power Interface/ACPI state transition data), Device Power States (e.g. “D0ix”) or Platform Power State (“s0ix”). A runtime credential exchange may be conducted, via a power management security interface, with a source

of the request at processing block 48. The runtime credential exchange may involve the transfer of security credentials from the source/originator of the request through the power management security interface to the PMIC, which may in turn determine the authenticity of the originator (e.g., through challenge/response attestations).

If it is determined at processing block 50 that the runtime credential exchange is successful (e.g., the credentials of the source/originator are verified), illustrated processing block 52 may grant the request and provide access to the power management register in question. If, on the other hand, it is determined at processing block 50 that the runtime credential exchange is unsuccessful (e.g., the credentials of the source/originator are not verified), illustrated processing block 54 enforces appropriate security policies such as, for example, denying the request, generating an alert, or other action. Processing block 54 may involve retrieving policy information from local policy storage, wherein the policy information (e.g., previously loaded from the TEE at boot time) includes action identifiers, credential data, power thresholds, etc., or any combination thereof. Thus, the request might be denied based on the retrieved policy information.

Illustrated processing block 56 determines whether a plug event (e.g., JTAG hot plug/unplug event) has been detected, via a dedicated side channel (e.g., out-of-band interface), with respect to a debug port. If so, an interrupt associated with the plug event may be handled by an interrupt service routing (ISR, previously registered with the side channel) at processing block 58. Illustrated processing block 60 may conduct a noise analysis of one or more power rails in response to the plug event. The noise analysis may generally identify potential security attacks based on the electrical characteristics of the power rails. For example, processing block 60 may determine whether voltage fluctuations observed on a given power rail are within expected voltage thresholds. In this regard, the unauthorized monitoring of a voltage rail may involve an electrical connection that generates unexpected noise (e.g., outside the voltage/power thresholds) on the power rail.

If processing block 60 identifies a potential security attack (e.g., the observed voltage fluctuations are outside acceptable thresholds), illustrated processing block 62 generates policy based counter noise on at least one of the one or more power rails. The policy based counter noise may generally prevent security threats from obtaining a valid measurement from the power rails involved. If it is determined at processing block 60 that no potential security attack has been identified, illustrated processing block 54

enforces appropriate security policies (e.g., generating an alert), as already discussed. The attributes of the attempt (e.g., type of attempt, status, request source and/or attacker identity) may also be logged at processing block 64. The rightmost register portion and the leftmost power rail portion shown in the method 44 may be implemented in combination or independently of one another (e.g., the register portion may be implemented without the power rail portion or the power rail portion may be implemented without the register portion), depending on the circumstances.

Turning now to FIG. 4, a power management computing architecture 66 is shown in which a power management integrated circuit (PMIC) 68 provides enhanced security. The PMIC 68 may generally implement one or more aspects of the method 30 (FIG. 2) and/or the method 44 (FIG. 3), already discussed. In the illustrated example, a host processor 70 includes one or more secure application instances 72 (e.g., security guard extension/SGX enclaves), a mobile operating system (OS) 74 and a power management security interface 76. One or more circuit blocks 78 may also be incorporated into the host processor 70 and/or elsewhere in the power management computing architecture 66. The architecture 66 may also include a trusted execution environment (TEE) 80 that includes secure storage 82, a power management unit (PMU) 84 that includes one or more power management registers 86, and a debug port 88 (e.g., JTAG port). The PMU 84 may use the power management registers 86 in communications that manage the power state transitions of the circuit blocks 78.

The illustrated PMIC 68 includes a secure register interface 90 (90a-90f) that generally restricts access to power management registers such as, for example, the power management registers 86 and/or one or more local power management registers 92. The PMIC 68 may also include a charger and power selector 106 that uses the local power management registers 92 to support the power state transitions managed by the PMU 84. The secure register interface 90 may have a request handler 90a to detect a request to access to one or more of the power management registers 86, 92, wherein the power management registers 86, 92 may include power consumption information associated with one or more circuit blocks. The source of the request may be, for example, one or more of the secure application instances 72, the mobile OS 74, malware (not shown), and so forth.

An authenticator 90b may conduct, via the power management security interface 76, a runtime credential exchange with the source of the request. In addition, a policy enforcer 90c may deny the request if the runtime credential exchange is

unsuccessful. In this regard, a policy retriever 90d may retrieve policy information (e.g., action identifiers, credential data, power thresholds) from local policy storage 98, wherein the request may be denied based on the policy information. If the runtime credential exchange is successful, the policy enforcer 90c may alternatively grant the request (e.g., based on the policy information).

The illustrated PMIC 68 also includes a port monitor 100 to detect, via a dedicated side channel 102, a plug event (e.g., JTAG hot plug/unplug event) with respect to the debug port 88. The illustrated side channel 102 is dedicated to the port monitor 100 and may be considered a side channel to the extent that it does not carry primary debug signals. A noise monitor 94 may conduct a noise analysis of one or more power rails (not shown) in response to the plug event. Additionally, the illustrated PMIC 68 includes a noise generator 96 to generate policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

More particularly, the general technique of conducting a policy based counter attack may be to use real time noise cancellation techniques applied to power loading. A description of one example of such a counter attack approach is as follows:

1) Monitor power fluctuations (e.g., sinusoidal waves);

2) Apply a real-time Fourier transform to generate a sinusoidal wave that is the cosine of the sine wave such that it exactly opposes the power wave;

3) Input the cosine wave into a variable resistor that is powered by the same power source feeding the monitored power rails;

4) Drive the current over the power rails that simultaneously feed the circuit block and variable resistor so that the load over the power rail is constant.

5) Apply a backoff-retry logic to the system so that when circuit block power requirements are relatively low, the variable resistor compensation wave is lowered (e.g., over a period of time seconds or milliseconds) to prevent power consumption at full power for long periods.

The secure register interface 90 may also include a boot detector 90e to detect boot events, wherein the authenticator 90b may conduct, in response to the boot events, boot time credential exchanges via the power management security interface 76 and the trusted execution environment 80. A policy loader 90f may load policy information from the secure storage 82 of the trusted execution environment 80 to the local policy storage 98 if the boot time credential exchanges are successful. Thus, the

trusted execution environment 80 may act as an arbitrator between the PMIC 68 and potential attackers, while maintaining, updating, and managing access to the policy information without concern over vulnerability to malware attacks. Indeed, the trusted execution environment 80 may make dynamic policy changes such as threshold updates (e.g., a 5% to 7% threshold adjustment), depending on the circumstances.

In addition, an interrupt handler 104 may register an interrupt service routine (ISR) with the dedicated side channel 102 if the boot time credential exchange is successful. Thus, links 108 between the trusted execution environment 80, the power management security interface 76, the secure register interface 90 and the power management unit 84, as well as the dedicated side channel 102, may be considered secure paths that are resistant to power management attacks. Additionally, the trusted execution environment 80 may schedule crypto operations including key generation operations, where a security relevant operation may be performed and an analog signature corresponding to the operation is supplied to the PMIC 68 to inform the noise generator 96 regarding a noise subtraction pattern that "cancels" the anticipated noise produced by the crypto operation.

FIG. 5 illustrates a processor core 200 according to one embodiment. The processor core 200 may be the core for any type of processor, such as a micro-processor, an embedded processor, a digital signal processor (DSP), a network processor, or other device to execute code. Although only one processor core 200 is illustrated in FIG. 5, a processing element may alternatively include more than one of the processor core 200 illustrated in FIG. 5. The processor core 200 may be a single-threaded core or, for at least one embodiment, the processor core 200 may be multithreaded in that it may include more than one hardware thread context (or "logical processor") per core.

FIG. 5 also illustrates a memory 270 coupled to the processor core 200. The memory 270 may be any of a wide variety of memories (including various layers of memory hierarchy) as are known or otherwise available to those of skill in the art. The memory 270 may include one or more code 213 instruction(s) to be executed by the processor core 200, wherein the code 213 may implement the method 30 (FIG. 2) and/or the method 44 (FIG. 3), already discussed. The processor core 200 follows a program sequence of instructions indicated by the code 213. Each instruction may enter a front end portion 210 and be processed by one or more decoders 220. The decoder 220 may generate as its output a micro operation such as a fixed width micro operation in a predefined format, or may generate other instructions, microinstructions, or control

signals which reflect the original code instruction. The illustrated front end portion 210 also includes register renaming logic 225 and scheduling logic 230, which generally allocate resources and queue the operation corresponding to the convert instruction for execution.

The processor core 200 is shown including execution logic 250 having a set of execution units 255-1 through 255-N. Some embodiments may include a number of execution units dedicated to specific functions or sets of functions. Other embodiments may include only one execution unit or one execution unit that can perform a particular function. The illustrated execution logic 250 performs the operations specified by code instructions.

After completion of execution of the operations specified by the code instructions, back end logic 260 retires the instructions of the code 213. In one embodiment, the processor core 200 allows out of order execution but requires in order retirement of instructions. Retirement logic 265 may take a variety of forms as known to those of skill in the art (e.g., re-order buffers or the like). In this manner, the processor core 200 is transformed during execution of the code 213, at least in terms of the output generated by the decoder, the hardware registers and tables utilized by the register renaming logic 225, and any registers (not shown) modified by the execution logic 250.

Although not illustrated in FIG. 5, a processing element may include other elements on chip with the processor core 200. For example, a processing element may include memory control logic along with the processor core 200. The processing element may include I/O control logic and/or may include I/O control logic integrated with memory control logic. The processing element may also include one or more caches.

Referring now to FIG. 6, shown is a block diagram of a computing system 1000 embodiment in accordance with an embodiment. Shown in FIG. 6 is a multiprocessor system 1000 that includes a first processing element 1070 and a second processing element 1080. While two processing elements 1070 and 1080 are shown, it is to be understood that an embodiment of the system 1000 may also include only one such processing element.

The system 1000 is illustrated as a point-to-point interconnect system, wherein the first processing element 1070 and the second processing element 1080 are coupled via a point-to-point interconnect 1050. It should be understood that any or all

of the interconnects illustrated in FIG. 6 may be implemented as a multi-drop bus rather than point-to-point interconnect.

As shown in FIG. 6, each of processing elements 1070 and 1080 may be multicore processors, including first and second processor cores (i.e., processor cores 1074a and 1074b and processor cores 1084a and 1084b). Such cores 1074a, 1074b, 1084a, 1084b may be configured to execute instruction code in a manner similar to that discussed above in connection with FIG. 5.

Each processing element 1070, 1080 may include at least one shared cache 1896a, 1896b. The shared cache 1896a, 1896b may store data (e.g., instructions) that are utilized by one or more components of the processor, such as the cores 1074a, 1074b and 1084a, 1084b, respectively. For example, the shared cache 1896a, 1896b may locally cache data stored in a memory 1032, 1034 for faster access by components of the processor. In one or more embodiments, the shared cache 1896a, 1896b may include one or more mid-level caches, such as level 2 (L2), level 3 (L3), level 4 (L4), or other levels of cache, a last level cache (LLC), and/or combinations thereof.

While shown with only two processing elements 1070, 1080, it is to be understood that the scope of the embodiments are not so limited. In other embodiments, one or more additional processing elements may be present in a given processor. Alternatively, one or more of processing elements 1070, 1080 may be an element other than a processor, such as an accelerator or a field programmable gate array. For example, additional processing element(s) may include additional processors(s) that are the same as a first processor 1070, additional processor(s) that are heterogeneous or asymmetric to processor a first processor 1070, accelerators (such as, e.g., graphics accelerators or digital signal processing (DSP) units), field programmable gate arrays, or any other processing element. There can be a variety of differences between the processing elements 1070, 1080 in terms of a spectrum of metrics of merit including architectural, micro architectural, thermal, power consumption characteristics, and the like. These differences may effectively manifest themselves as asymmetry and heterogeneity amongst the processing elements 1070, 1080. For at least one embodiment, the various processing elements 1070, 1080 may reside in the same die package.

The first processing element 1070 may further include memory controller logic (MC) 1072 and point-to-point (P-P) interfaces 1076 and 1078. Similarly, the second processing element 1080 may include a MC 1082 and P-P interfaces 1086 and

1088. As shown in FIG. 6, MC's 1072 and 1082 couple the processors to respective memories, namely a memory 1032 and a memory 1034, which may be portions of main memory locally attached to the respective processors. While the MC 1072 and 1082 is illustrated as integrated into the processing elements 1070, 1080, for alternative embodiments the MC logic may be discrete logic outside the processing elements 1070, 1080 rather than integrated therein.

The first processing element 1070 and the second processing element 1080 may be coupled to an I/O subsystem 1090 via P-P interconnects 1076 1086, respectively. As shown in FIG. 6, the I/O subsystem 1090 includes P-P interfaces 1094 and 1098. Furthermore, I/O subsystem 1090 includes an interface 1092 to couple I/O subsystem 1090 with a high performance graphics engine 1038. In one embodiment, bus 1049 may be used to couple the graphics engine 1038 to the I/O subsystem 1090. Alternately, a point-to-point interconnect may couple these components.

In turn, I/O subsystem 1090 may be coupled to a first bus 1016 via an interface 1096. In one embodiment, the first bus 1016 may be a Peripheral Component Interconnect (PCI) bus, or a bus such as a PCI Express bus or another third generation I/O interconnect bus, although the scope of the embodiments are not so limited.

As shown in FIG. 6, various I/O devices 1014 (e.g., speakers, cameras, sensors) may be coupled to the first bus 1016, along with a bus bridge 1018 which may couple the first bus 1016 to a second bus 1020. In one embodiment, the second bus 1020 may be a low pin count (LPC) bus. Various devices may be coupled to the second bus 1020 including, for example, a keyboard/mouse 1012, communication device(s) 1026, and a data storage unit 1019 such as a disk drive or other mass storage device which may include code 1030, in one embodiment. The illustrated code 1030 may implement the method 30 (FIG. 2) and/or the method 44 (FIG. 3), already discussed, and may be similar to the code 213 (FIG. 5), already discussed. Further, an audio I/O 1024 may be coupled to second bus 1020 and a battery 1010 may supply power to the computing system 1000.

Note that other embodiments are contemplated. For example, instead of the point-to-point architecture of FIG. 6, a system may implement a multi-drop bus or

another such communication topology. Also, the elements of FIG. 6 may alternatively be partitioned using more or fewer integrated chips than shown in FIG. 6.

Additional Notes and Examples:

Example 1 may include a system on chip (SoC) computing architecture comprising a power source to supply power to the SoC, a plurality of circuit blocks and a power management integrated circuit including a power management register associated with one or more of the plurality of circuit blocks, a request handler to detect a request to access the power management register, an authenticator to conduct, via a power management security interface, a runtime credential exchange with a source of the request, and a policy enforcer to deny the request if the runtime credential exchange is unsuccessful.

Example 2 may include the system on chip of Example 1, further including one or more power rails coupled to the power source and the plurality of circuit blocks, and a debug port, wherein the power management integrated circuit includes a port monitor to detect, via a dedicated side channel, a plug event with respect to the debug port, a noise monitor to conduct a noise analysis of the one or more power rails in response to the plug event, and a noise generator to generate policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

Example 3 may include the system on chip of Example 1, further including a trusted execution environment, wherein the power management integrated circuit includes a boot detector to detect a boot event, wherein the authenticator is to conduct, in response to the boot event, a boot time credential exchange via the power management security interface and the trusted execution environment, a policy loader to load policy information from the trusted execution environment to local policy storage if the boot time credential exchange is successful, and an interrupt handler to register an interrupt search routine with a dedicated side channel of a debug port if the boot time credential exchange is successful.

Example 4 may include the system on chip of Example 1, wherein the power management integrated circuit further includes a local policy storage, and a policy retriever to retrieve policy information from the local policy storage, wherein the request is to be denied based on the policy information and the policy information includes one or more of action identifiers, credential data or power thresholds.

Example 5 may include the system on chip of Example 1, wherein the policy enforcer is to grant the request if the runtime credential exchange is successful.

Example 6 may include the system on chip of any one of Examples 1 to 5, wherein the power management register is to include power consumption information.

Example 7 may include a power management apparatus comprising a request handler to detect a request to access a power management register, an authenticator to conduct, via a power management security interface, a runtime credential exchange with a source of the request, and a policy enforcer to deny the request if the runtime credential exchange is unsuccessful.

Example 8 may include the apparatus of Example 7, further including a port monitor to detect, via a dedicated side channel, a plug event with respect to a debug port, a noise monitor to conduct a noise analysis of one or more power rails in response to the plug event, a noise generator to generate policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

Example 9 may include the apparatus of Example 7, further including a boot detector to detect a boot event, wherein the authenticator is to conduct, in response to the boot event, a boot time credential exchange via the power management security interface and a trusted execution environment, a policy loader to load policy information from the trusted execution environment to local policy storage if the boot time credential exchange is successful, and an interrupt handler to register an interrupt service routine with a dedicated side channel of a debug port if the boot time credential exchange is successful.

Example 10 may include the apparatus of Example 7, further including a policy retriever to retrieve policy information from local policy storage, wherein the request is to be denied based on the policy information and the policy information includes one or more of action identifiers, credential data or power thresholds.

Example 11 may include the apparatus of Example 7, wherein the policy enforcer is to grant the request if the runtime credential exchange is successful.

Example 12 may include the apparatus of any one of Examples 7 to 11, wherein the power management register is to include power consumption information.

Example 13 may include a method of operating a power management integrated circuit, comprising detecting a request to access a power management register, conducting, via a power management security interface, a runtime credential

exchange with a source of the request, and denying the request if the runtime credential exchange is unsuccessful.

Example 14 may include the method of Example 13, further including detecting, via a dedicated side channel, a plug event with respect to a debug port, conducting a noise analysis of one or more power rails in response to the plug event, and generating policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

Example 15 may include the method of Example 13, further including detecting a boot event, conducting, in response to the boot event, a boot time credential exchange via the power management security interface and a trusted execution environment, loading policy information from the trusted execution environment to local policy storage if the boot time credential exchange is successful, and registering an interrupt service routine with a dedicated side channel of a debug port if the boot time credential exchange is successful.

Example 16 may include the method of Example 13, further including retrieving policy information from local policy storage, wherein the request is denied based on the policy information and the policy information includes one or more of action identifiers, credential data or power thresholds.

Example 17 may include the method of Example 13, further including granting the request of the runtime credential exchange is successful.

Example 18 may include the method of any one of Examples 13 to 17, wherein the power management register includes power consumption information.

Example 19 may include at least one computer readable storage medium comprising a set of instructions, which when executed by a computing system, cause the computing system to detect a request to access a power management register, conduct, via a power management security interface, a runtime credential exchange with a source of the request, and deny the request if the runtime credential exchange is unsuccessful.

Example 20 may include the at least one computer readable storage medium of Example 19, wherein the instructions, when executed, cause a computing system to detect, via a dedicated side channel, a plug event with respect to a debug port, conduct a noise analysis of one or more power rails in response to the plug event, and generate policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

Example 21 may include the at least one computer readable storage medium of Example 19, wherein the instructions, when executed, cause a computing system to detect a boot event, conduct, in response to the boot event, a boot time credential exchange via the power management security interface and a trusted execution environment, load policy information from the trusted execution environment to local policy storage if the boot time credential exchange is successful, and register an interrupt service routine with a dedicated side channel of a debug port if the boot time credential exchange is successful.

Example 22 may include the at least one computer readable storage medium of Example 19, wherein the instructions, when executed, cause a computing system to retrieve policy information from local policy storage, and wherein the request is to be denied based on the policy information and the policy information includes one or more of action identifiers, credential data or power thresholds.

Example 23 may include the at least one computer readable storage medium of Example 19, wherein the instructions, when executed, cause a computing system to grant the request if the runtime credential exchange is successful.

Example 24 may include the at least one computer readable storage medium of any one of Examples 19 to 23, wherein the power management register is to include power consumption information.

Example 25 may include a security-enhanced power management apparatus comprising means for detecting a request to access a power management register, means for conducting, via a power management security interface, a runtime credential exchange with a source of the request, and means for denying the request if the runtime credential exchange is unsuccessful.

Example 26 may include the apparatus of Example 25, further including: means for detecting, via a dedicated side channel, a plug event with respect to a debug port, means for conducting a noise analysis of one or more power rails in response to the plug event, and means for generating policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

Example 27 may include the apparatus of Example 25, further including means for detecting a boot event, means for conducting, in response to the boot event, a boot time credential exchange via the power management security interface and a trusted execution environment, means for loading policy information from the trusted

execution environment to local policy storage if the boot time credential exchange is successful, and means for registering an interrupt service routine with a dedicated side channel of a debug port if the boot time credential exchange is successful.

Example 28 may include the apparatus of Example 25, further including means for retrieving policy information from local policy storage, wherein the request is to be denied based on the policy information and the policy information includes one or more of action identifiers, credential data or power thresholds.

Example 29 may include the apparatus of Example 25, further including means for granting the request of the runtime credential exchange is successful.

Example 30 may include the apparatus of any one of Examples 25 to 29, wherein the power management register is to include power consumption information.

Thus, techniques described herein may expose power management registers via a secure power management security interface so that only trusted authorized agents are able to manage and monitor power in a computing system. Moreover, JTAG hot plug/unplug events may be detected via a secure out-of-band dedicated channel. Power management integrated circuits may be provisioned by a trusted execution environment with expected power thresholds, wherein the power management integrated circuits may use the provisioned power thresholds to enforce policy based actions. In one example, power management integrated circuits include a noise monitor/detector and a noise generator to detect any side channel attacks and introduce runtime scramble/fuzz noise appropriately.

Embodiments are applicable for use with all types of semiconductor integrated circuit ("IC") chips. Examples of these IC chips include but are not limited to processors, controllers, chipset components, programmable logic arrays (PLAs), memory chips, network chips, systems on chip (SoCs), SSD/NAND controller ASICs, and the like. In addition, in some of the drawings, signal conductor lines are represented with lines. Some may be different, to indicate more constituent signal paths, have a number label, to indicate a number of constituent signal paths, and/or have arrows at one or more ends, to indicate primary information flow direction. This, however, should not be construed in a limiting manner. Rather, such added detail may be used in connection with one or more exemplary embodiments to facilitate easier understanding of a circuit. Any represented signal lines, whether or not having additional information, may actually comprise one or more signals that may travel in multiple directions and may be implemented with any suitable type of signal scheme,

e.g., digital or analog lines implemented with differential pairs, optical fiber lines, and/or single-ended lines.

Example sizes/models/values/ranges may have been given, although embodiments are not limited to the same. As manufacturing techniques (e.g., photolithography) mature over time, it is expected that devices of smaller size could be manufactured. In addition, well known power/ground connections to IC chips and other components may or may not be shown within the figures, for simplicity of illustration and discussion, and so as not to obscure certain aspects of the embodiments. Further, arrangements may be shown in block diagram form in order to avoid obscuring embodiments, and also in view of the fact that specifics with respect to implementation of such block diagram arrangements are highly dependent upon the computing system within which the embodiment is to be implemented, i.e., such specifics should be well within purview of one skilled in the art. Where specific details (e.g., circuits) are set forth in order to describe example embodiments, it should be apparent to one skilled in the art that embodiments can be practiced without, or with variation of, these specific details. The description is thus to be regarded as illustrative instead of limiting.

The term "coupled" may be used herein to refer to any type of relationship, direct or indirect, between the components in question, and may apply to electrical, mechanical, fluid, optical, electromagnetic, electromechanical or other connections. In addition, the terms "first", "second", etc. may be used herein only to facilitate discussion, and carry no particular temporal or chronological significance unless otherwise indicated.

As used in this application and in the claims, a list of items joined by the term "one or more of" may mean any combination of the listed terms. For example, the phrases "one or more of A, B or C" may mean A; B; C; A and B; A and C; B and C; or A, B and C.

Those skilled in the art will appreciate from the foregoing description that the broad techniques of the embodiments can be implemented in a variety of forms. Therefore, while the embodiments have been described in connection with particular examples thereof, the true scope of the embodiments should not be so limited since other modifications will become apparent to the skilled practitioner upon a study of the drawings, specification, and following claims.

CLAIMS

We claim:

1. A system on chip comprising:

a power source to supply power to the system;

a plurality of circuit blocks; and

a power management integrated circuit including:

a power management register associated with one or more of the plurality of circuit blocks,

a request handler to detect a request to access the power management register,

an authenticator to conduct, via a power management security interface, a runtime credential exchange with a source of the request, and

a policy enforcer to deny the request if the runtime credential exchange is unsuccessful.

2. The system on chip of claim 1, further including:

one or more power rails coupled to the power source and the plurality of circuit blocks; and

a debug port,

wherein the power management integrated circuit includes:

a port monitor to detect, via a dedicated side channel, a plug event with respect to the debug port,

a noise monitor to conduct a noise analysis of the one or more power rails in response to the plug event, and

a noise generator to generate policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

3. The system on chip of claim 1, further including a trusted execution environment, wherein the power management integrated circuit includes:

a boot detector to detect a boot event, wherein the authenticator is to conduct, in response to the boot event, a boot time credential exchange via the power management security interface and the trusted execution environment,

a policy loader to load policy information from the trusted execution environment to local policy storage if the boot time credential exchange is successful, and

an interrupt handler to register an interrupt search routine with a dedicated side channel of a debug port if the boot time credential exchange is successful.

4. The system on chip of claim 1, wherein the power management integrated circuit further includes:

a local policy storage, and

a policy retriever to retrieve policy information from the local policy storage, wherein the request is to be denied based on the policy information and the policy information includes one or more of action identifiers, credential data or power thresholds.

5. The system on chip of claim 1, wherein the policy enforcer is to grant the request if the runtime credential exchange is successful.

6. The system on chip of any one of claims 1 to 5, wherein the power management register is to include power consumption information.

7. A security-enhanced power management apparatus comprising:

a request handler to detect a request to access a power management register;

an authenticator to conduct, via a power management security interface, a runtime credential exchange with a source of the request; and

a policy enforcer to deny the request if the runtime credential exchange is unsuccessful.

8. The apparatus of claim 7, further including:

a port monitor to detect, via a dedicated side channel, a plug event with respect to a debug port;

a noise monitor to conduct a noise analysis of one or more power rails in response to the plug event;

a noise generator to generate policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

9. The apparatus of claim 7, further including:

a boot detector to detect a boot event, wherein the authenticator is to conduct, in response to the boot event, a boot time credential exchange via the power management security interface and a trusted execution environment;

a policy loader to load policy information from the trusted execution environment to local policy storage if the boot time credential exchange is successful; and

an interrupt handler to register an interrupt service routine with a dedicated side channel of a debug port if the boot time credential exchange is successful.

10. The apparatus of claim 7, further including a policy retriever to retrieve policy information from local policy storage, wherein the request is to be denied based on the policy information and the policy information includes one or more of action identifiers, credential data or power thresholds.

11. The apparatus of claim 7, wherein the policy enforcer is to grant the request if the runtime credential exchange is successful.

12. The apparatus of any one of claims 7 to 11, wherein the power management register is to include power consumption information.

13. A method of operating a security-enhanced power management apparatus, comprising:

detecting a request to access a power management register;

conducting, via a power management security interface, a runtime credential exchange with a source of the request; and

denying the request if the runtime credential exchange is unsuccessful.

14. The method of claim 13, further including:

detecting, via a dedicated side channel, a plug event with respect to a debug port;

conducting a noise analysis of one or more power rails in response to the plug event; and

generating policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

15. The method of claim 13, further including:

detecting a boot event;

conducting, in response to the boot event, a boot time credential exchange via the power management security interface and a trusted execution environment;

loading policy information from the trusted execution environment to local policy storage if the boot time credential exchange is successful; and

registering an interrupt service routine with a dedicated side channel of a debug port if the boot time credential exchange is successful.

16. The method of claim 13, further including retrieving policy information from local policy storage, wherein the request is denied based on the policy information and the policy information includes one or more of action identifiers, credential data or power thresholds.

17. The method of claim 13, further including granting the request of the runtime credential exchange is successful.

18. The method of any one of claims 13 to 17, wherein the power management register includes power consumption information.

19. At least one computer readable storage medium comprising a set of instructions, which when executed by a computing system, cause the computing system to:

detect a request to access a power management register;

conduct, via a power management security interface, a runtime credential exchange with a source of the request; and

deny the request if the runtime credential exchange is unsuccessful.

20. The at least one computer readable storage medium of claim 19, wherein the instructions, when executed, cause a computing system to:

detect, via a dedicated side channel, a plug event with respect to a debug port;

conduct a noise analysis of one or more power rails in response to the plug event; and

generate policy based counter noise on at least one of the one or more power rails at runtime if the noise analysis identifies a potential security attack.

21. The at least one computer readable storage medium of claim 19, wherein the instructions, when executed, cause a computing system to:

detect a boot event;

conduct, in response to the boot event, a boot time credential exchange via the power management security interface and a trusted execution environment;

load policy information from the trusted execution environment to local policy storage if the boot time credential exchange is successful; and

register an interrupt service routine with a dedicated side channel of a debug port if the boot time credential exchange is successful.

22. The at least one computer readable storage medium of claim 19, wherein the instructions, when executed, cause a computing system to retrieve policy information from local policy storage, and wherein the request is to be denied based on the policy information and the policy information includes one or more of action identifiers, credential data or power thresholds.

23. The at least one computer readable storage medium of claim 19, wherein the instructions, when executed, cause a computing system to grant the request if the runtime credential exchange is successful.

24. The at least one computer readable storage medium of any one of claims 19 to 23, wherein the power management register is to include power consumption information.

25. A security-enhanced power management apparatus comprising means for performing the method of any one of claims 13 to 17.

FIG. 1

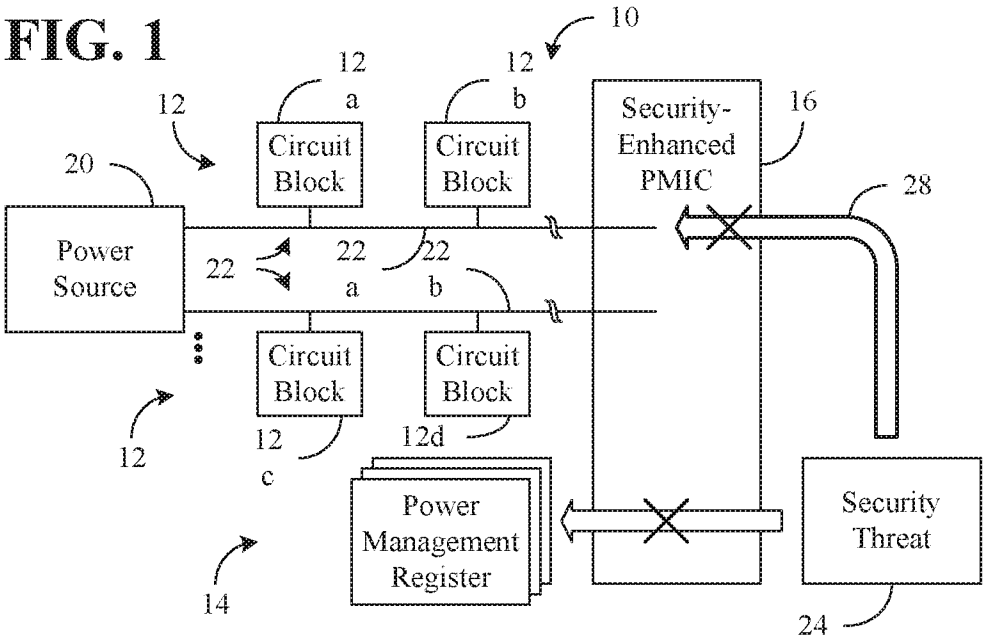
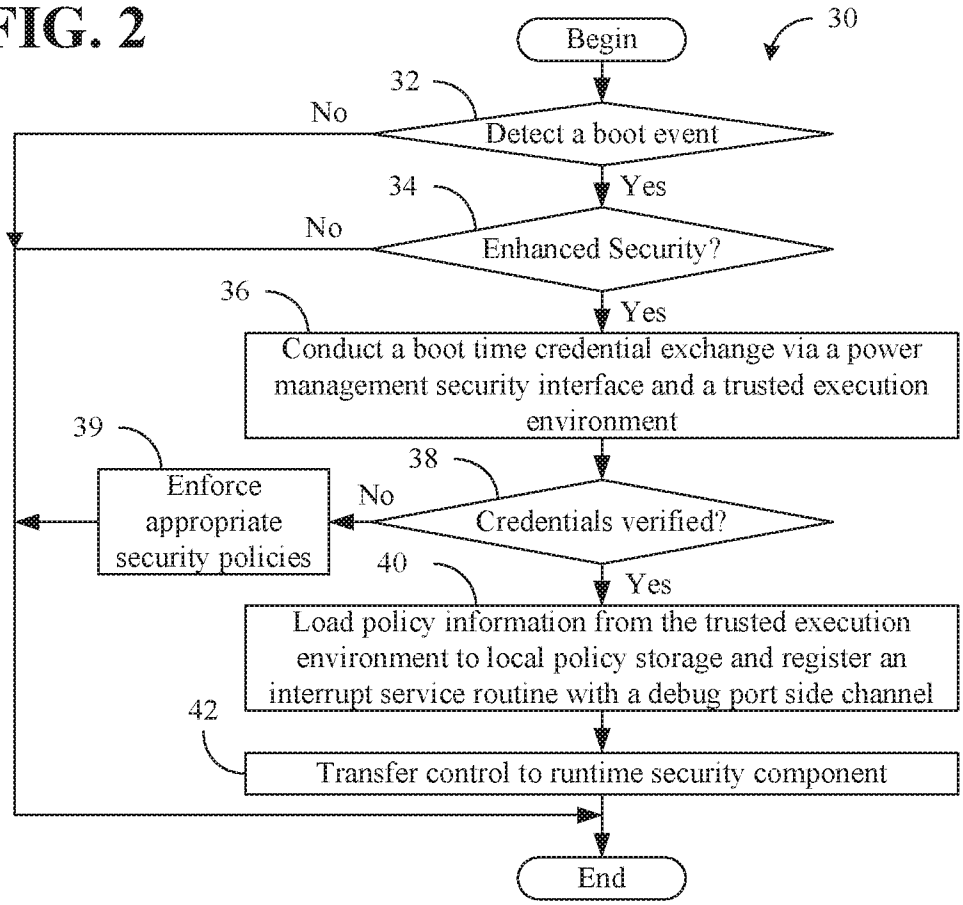

10
12
12 a
12 b
Circuit Block
Circuit Block
Security-Enhanced PMIC
16
20
28
Power Source
22
22 a
22 b
Circuit Block
Circuit Block
12
12 c
12d
Power Management Register
Security Threat
14
24

FIG. 2


30
Begin
32
No
Detect a boot event
Yes
34
No
Enhanced Security?
Yes
36
Conduct a boot time credential exchange via a power management security interface and a trusted execution environment
39
38
Enforce appropriate security policies
No
Credentials verified?
40
Yes
Load policy information from the trusted execution environment to local policy storage and register an interrupt service routine with a debug port side channel
42
Transfer control to runtime security component
End

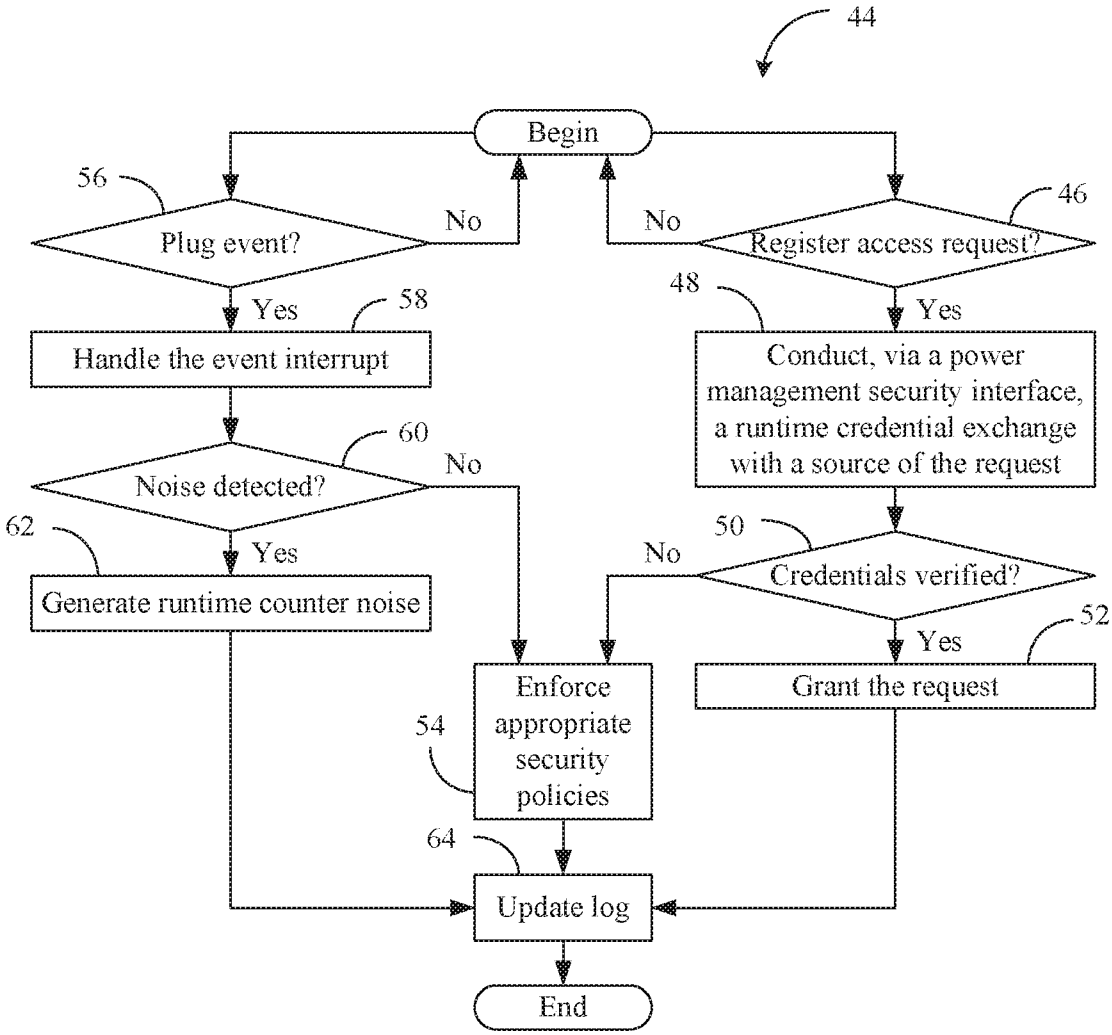

44
Begin
56
Plug event?
No
Yes
58
Handle the event interrupt
60
Noise detected?
No
62
Yes
Generate runtime counter noise
No
46
Register access request?
48
Yes
Conduct, via a power management security interface, a runtime credential exchange with a source of the request
50
Credentials verified?
No
52
Yes
Grant the request
Enforce appropriate security policies
54
64
Update log
End

FIG. 3

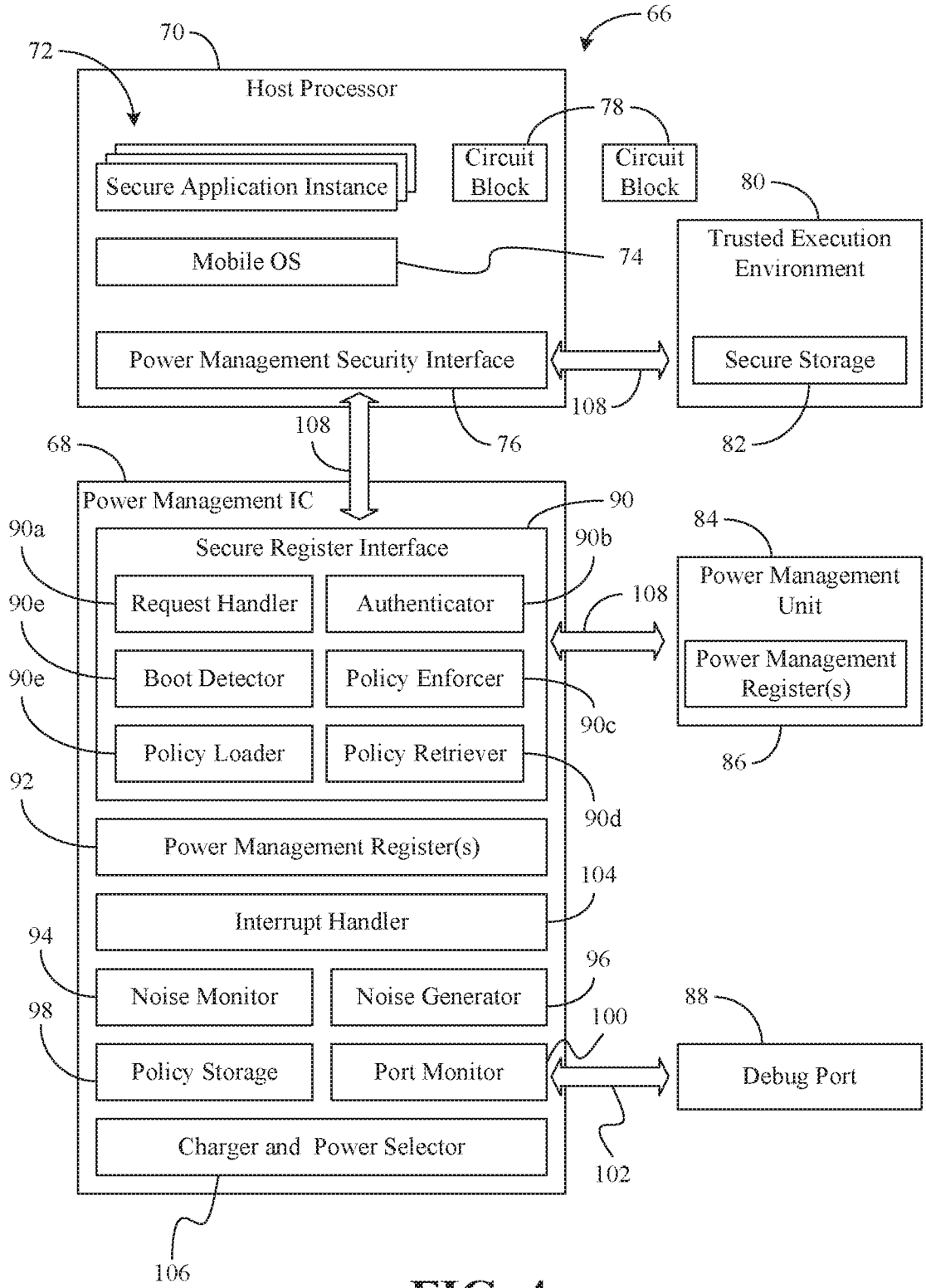

66
70
72
Host Processor
78
Secure Application Instance
Circuit Block
Circuit Block
80
Trusted Execution Environment
Mobile OS
74
Power Management Security Interface
Secure Storage
108
108
76
82
68
Power Management IC
90
84
90a
Secure Register Interface
90b
Power Management Unit
108
90e
Request Handler
Authenticator
90e
Boot Detector
Policy Enforcer
90c
Power Management Register(s)
90e
Policy Loader
Policy Retriever
86
92
90d
Power Management Register(s)
104
Interrupt Handler
94
96
Noise Monitor
Noise Generator
88
98
100
Policy Storage
Port Monitor
Debug Port
Charger and Power Selector
102
106

FIG. 4

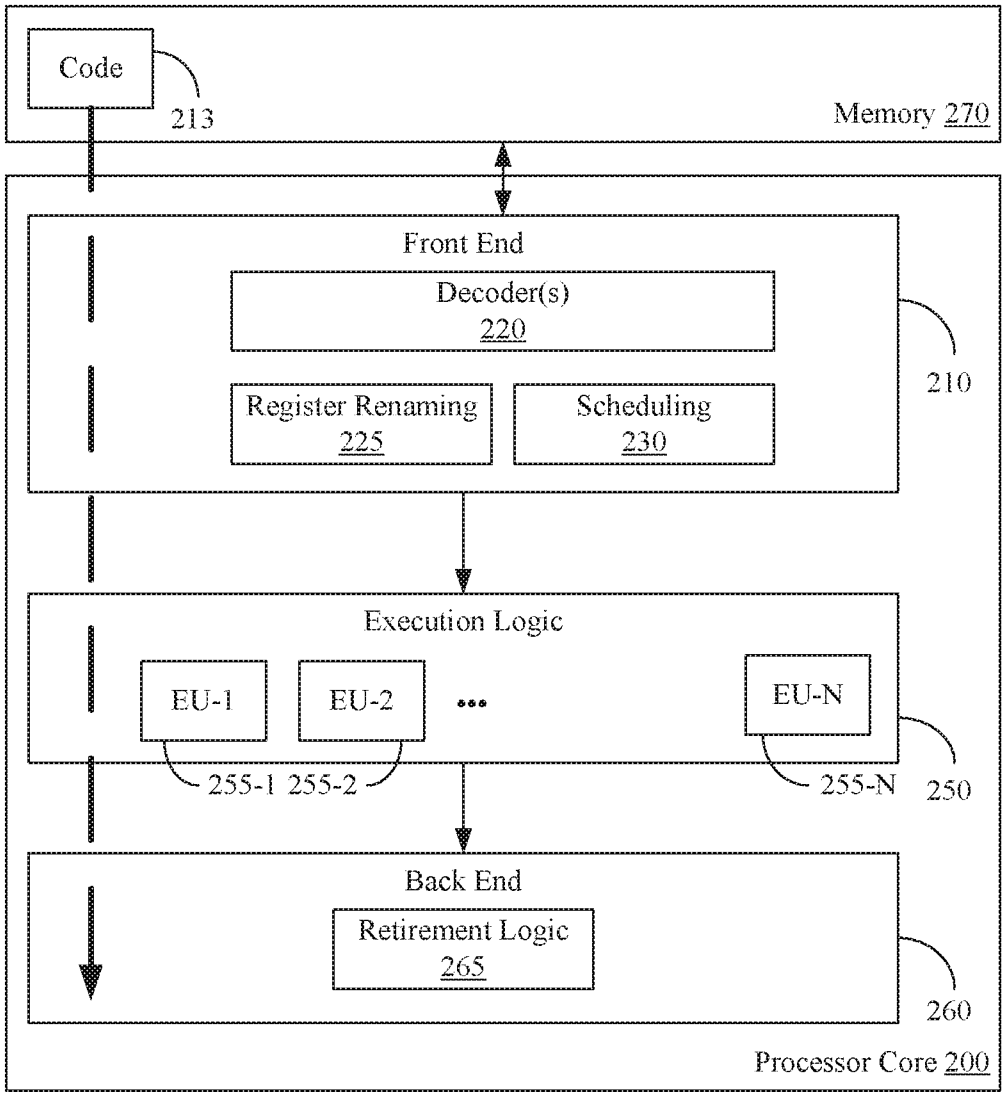
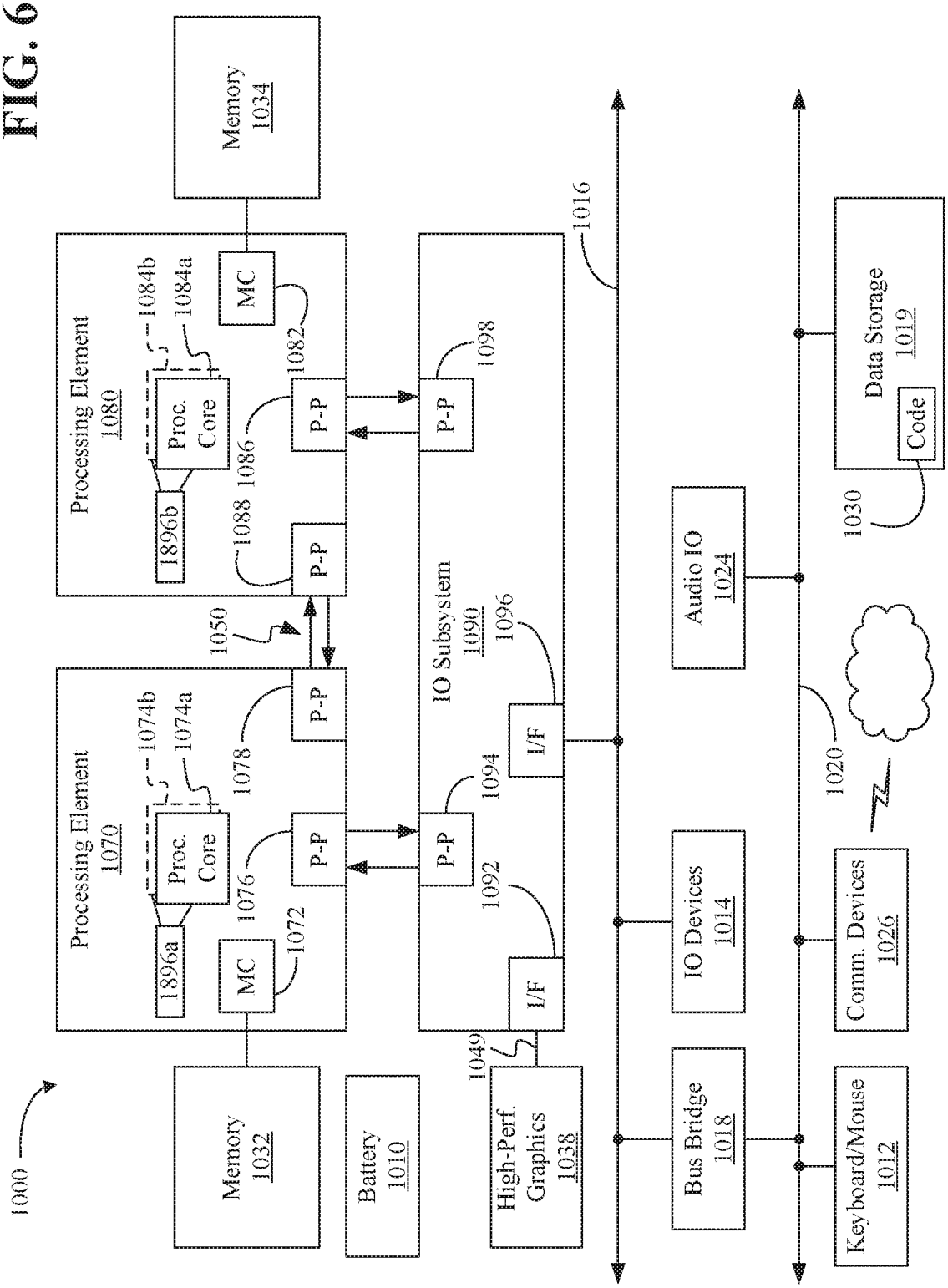

Code
213
Memory 270
Front End
Decoder(s)
220
Register Renaming
225
Scheduling
230
210
Execution Logic
EU-1
EU-2
...
EU-N
255-1
255-2
255-N
250
Back End
Retirement Logic
265
260
Processor Core 200

FIG. 5

FIG. 6


1000
Memory 1032
Battery 1010
High-Perf. Graphics 1038
Bus Bridge 1018
IO Devices 1014
Keyboard/Mouse 1012
Comm. Devices 1026
Audio IO 1024
Data Storage 1019
Code
1030
1020
1016
1049
I/F
1092
1096
IO Subsystem 1090
P-P
1094
1098
1050
Processing Element 1070
1074b
1074a
Proc. Core
1076
1078
MC
1072
1896a
Processing Element 1080
1084b
1084a
1086
1088
1082
1896b
Memory 1034

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/029103

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/55(2013.01)i, G06F 1/26(2006.01)i, G06F 15/78(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 21/55; G01R 31/3185; H04B 5/00; H04L 9/32; G01R 31/317; G06F 21/00; G06F 1/26; G06F 15/78

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: Power Management Integrated Circuit, security, System on Chip, side channel

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2014-0089682 A1 (APPLE INC.) 27 March 2014 See abstract; paragraphs [0030]-[0031], [0047]-[0049], [0052]-[0053], [0062], [0069], [0072], [0076]-[0077], [0083]; and figures 1-3, 5-6.	1,3-5,7,9-11,13,15-17,19,21-23,25
A		2,6,8,12,14,18,20,24
A	WO 2015-032571 A1 (TELEFONAKTIEBOLAGET L M ERICSSON (PUBL)) 12 March 2015 See page 13, lines 7-31; claim 1; and figures 5-6.	1-25
A	US 2014-0013406 A1 (CHRISTOPHE TREMLET) 09 January 2014 See claim 1; and figures 1A, 3, 6.	1-25
A	US 2009-0094702 A1 (CHING-CHAO YANG et al.) 09 April 2009 See claim 1; and figure 4.	1-25
A	US 2010-0241864 A1 (JIM KELLEY et al.) 23 September 2010 See abstract; and claim 1	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
27 July 2016 (27.07.2016)	**27 July 2016 (27.07.2016)**

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
CHIN, Sang Bum
Telephone No. +82-42-481-8398

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/029103

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0089682 A1	27/03/2014	US 8832465 B2	09/09/2014
WO 2015-032571 A1	12/03/2015	EP 2843429 A1	04/03/2015
US 2014-0013406 A1	09/01/2014	CN 103544599 A	29/01/2014
		DE 102013106295 A1	09/01/2014
US 2009-0094702 A1	09/04/2009	CN 101404799 A	08/04/2009
		DE 102007061583 A1	09/04/2009
		TW 200917801 A	16/04/2009
US 2010-0241864 A1	23/09/2010	WO 2010-059960 A1	27/05/2010