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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 July 2013 (04.07.20 13)

W I P O | P C T

(10) International Publication Number
WO 2013/101178 Al

(51) **International Patent Classification:**
G06F 21/22 (2006.01) ***G06F 9/24*** (2006.01)
G06F 13/14 (2006.01)

(21) **International Application Number:**
PCT/US201 1/068078

(22) **International Filing Date:**
30 December 201 1 (30. 12.201 1)

(25) **Filing Language:** English

(26) **Publication Language:** English

(71) **Applicant** *(for all designated States except US):* **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, M/S: RNB-4-150, Santa Clara, California 95052 (US).

(72) **Inventor; and**
(75) **Inventor/Applicant** *(for US only):* **WISEMAN, Willard M.** [US/US]; 14199 SW 120th Place, Tigard, Oregon 97224 (US).

(74) **Agents: VINCENT, Lester J.** et al; Blakely, Sokoloff, Taylor & Zafman LLP, 1279 Oakmead Parkway, Sunnyvale, California 94085-4040 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(ivf)*

Published:

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

(54) Title: USING A TRUSTED PLATFORM MODULE FOR BOOT POLICY AND SECURE FIRMWARE

(57) Abstract: Embodiments of apparatuses and methods for using a trusted platform module for boot policy and secure firmware are disclosed. In one embodiment, a trusted platform module includes a non-volatile memory, a port, and a mapping structure. The port is to receive an input/output transaction from a serial bus. The transaction includes a system memory address in the address space of a processor. The mapping structure is to map the system memory address to a first location in non-volatile memory.

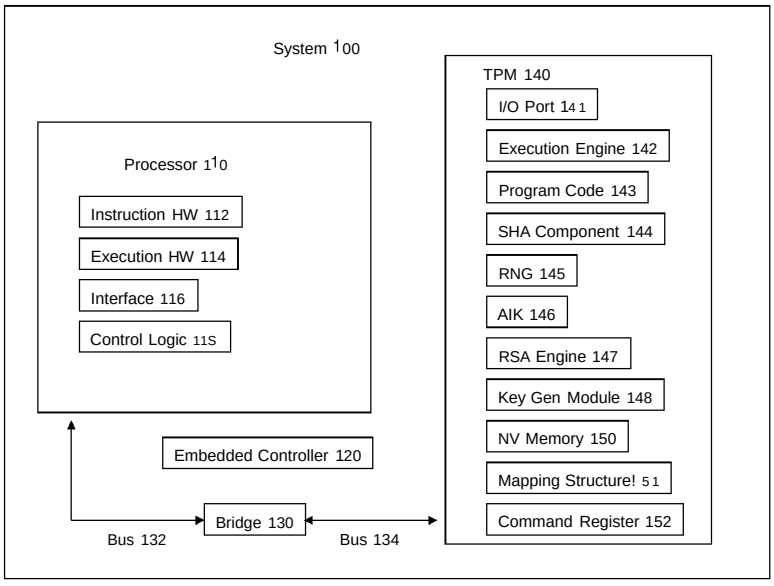

FIGURE 1

WO 2013/101178 A1

USING A TRUSTED PLATFORM MODULE FOR BOOT POLICY AND SECURE FIRMWARE

BACKGROUND

Field

The present disclosure pertains to the field of information processing, and more particularly, to the field of security in information processing systems.

Description of Related Art

A Trusted Platform Module ("TPM") is a hardware component that provides a specific set of security functions to an information processing system. These functions include secure key generation and storage and authenticated access to encrypted data. A TPM may be implemented according to specifications such as the TPM Main Specification Version 1.2, published by the Trusted Computed Group and available from the Internet at www.trustedcomputinggroup.org.

The sub-components of a TPM may include an execution engine and secure non-volatile ("NV") memory or storage. The secure NV memory may be used to store sensitive information, such as encryption keys, and the execution engine may be used to protect the sensitive information according to the security policies dictated by the TPM's control logic. An inherent feature of a TPM is the control logic that implements access controls which prevent unauthorized access to the NV memory. A TPM is typically accessed through a serial bus connection.

Brief Description of the Figures

The present invention is illustrated by way of example and not limitation in the accompanying figures.

Figure 1 illustrates a system and a TPM in which an embodiment of the present invention may be present and/or operate.

Figure 2 illustrates a method for using a TPM for boot policy and secure firmware according to an embodiment of the present invention.

Detailed Description

Embodiments of apparatuses, methods, and systems for using a TPM for boot policy and secure firmware are described below. In this description, numerous specific details, such as component and system configurations, may be set forth in order to provide a more thorough understanding of the present invention. It will be appreciated, however, by one skilled in the art, that the invention may be practiced without such specific details. Additionally, some well known structures, circuits, and the like have not been shown in detail, to avoid unnecessarily obscuring the present invention.

Embodiments of the present invention may be used for managing boot policy and for storing basic input/output system ("BIOS") firmware and firmware for platform components such as embedded controllers, taking advantage of a TPM's strict access control features. Embodiments provide for a using the NV memory of a TPM to store such firmware, and adding a mapping table to the TPM to provide for this NV memory space to be accessed as memory mapped input/out (MMIO) memory space, so that a processor or other agent can access the firmware using instructions and/or micro-instructions based on its standard instruction set architecture and microarchitecture, rather than requiring it to use the special access commands typically needed to access NV memory of a TPM.

Figure 1 illustrates system 100, an information processing system in which an embodiment of the present invention may be present and/or operate. System 100 may represent any type of information processing system, such as a server, a desktop computer, a portable computer, a set-top box, a hand-held device, or an embedded control system. System 100 includes processor 110, embedded controller 120, bridge 130, and TPM 140. Systems embodying the present invention may include number of each of these components and any other components or other elements. Any or all of the components or other elements in any system embodiment may be connected, coupled, or otherwise in communication with each other through any number of buses, point-to-point, or other wired or wireless connections.

Processor 110 may represent any type of processor, including a general purpose microprocessor, such as a processor in the Core® Processor Family, or other processor family from Intel Corporation, or another processor from another company, or any other processor for processing information according to an embodiment of the present invention. Processor 110 may include any number of execution cores and/or support any number of execution threads, and therefore may represent any number of physical or logical processors, and/or may represent a multi-processor component or unit.

Processor 110 may include instruction hardware 112, execution hardware 114, interface 116, and control logic 118, plus any other desired units or elements.

Instruction hardware 112 may represent any circuitry, structure, or other hardware, such as an instruction decoder, for fetching, receiving, decoding, and/or scheduling instructions. Any instruction format may be used within the scope of the present invention; for example, an instruction may include an opcode and one or more operands, where the opcode may be decoded into one or more micro-instructions or micro-operations for execution by execution hardware 114. Execution hardware 114 may include any circuitry, structure, or other hardware, such as an arithmetic unit, logic unit, floating point unit, shifter, etc., for processing data and executing instructions, micro-instructions, and/or micro-operations.

Interface unit 116 may represent any circuitry, structure, or other hardware, such as a bus unit or any other unit, port, or interface, to allow processor 110 to communicate with other components in system 100 through any type of bus, point to point, or other connection, directly or through any other component, such as a memory controller or a bus bridge.

Control logic 118 may represent microcode, programmable logic, hard-coded logic, or any other type of logic to control the operation of the units and other elements of processor 110 and the transfer of data within processor 110. Control logic 118 may cause processor 110 to perform or participate in the performance of method embodiments of the present invention, such as the method embodiments described below, for example, by causing processor 110 to execute instructions received by instruction hardware 112 and micro-instructions or micro-operations derived from instructions received by instruction hardware 112.

Embedded controller 120 may represent any embedded controller, microcontroller, or other component or device, the operation of which may include the use of firmware. Bridge 130 may represent any component or device including circuitry or logic to provide for the communication between components, devices, or other elements of system 100. For example, bridge 130 may include a bridge between a bus through which processor 110 may access memory and a bus through which I/O devices may be accessed. Therefore, bridge 130 may provide for forwarding MMIO transactions from processor 110 through processor bus 132 to TPM 140 through serial bus 134.

TPM 140 includes I/O port 141 for receiving TPM commands and communicating the results over a serial bus, such as a serial peripheral interface ("SPI") or low pin count ("LPC") bus. Execution engine 142 executes program code 143 in response to the TPM commands and utilizes a variety of different TPM components to perform the necessary operations including a secure hash algorithm 1 (SHA-1) component 144 for performing SHA-1 hash operations; a random number generator 145 for generating random numbers; an attestation identify key (AIK) component 146 for securely establishing that a remote entity is communicating with the TPM; an RSA engine 147 for implementing RSA encryption and digital signatures; and a key generation module 148 for generating keys.

TPM 140 also includes NV memory 150, mapping structure 151, and command register 152. Mapping structure 151 may be include any circuitry, structure, or other hardware or firmware to map storage locations in NV memory 150 to addresses within the system memory space of system 100. Therefore, NV memory 150 may be accessed with memory transactions from processor 110 or embedded controller 120, forwarded to serial bus 134 through bridge 130 as MMIO transactions. The addresses may be received from serial bus 134 at I/O port 141, and mapped to a location in NV memory 150 by mapping structure 151. Therefore, NV memory 150

may be used to store firmware that may be accessed and executed by processor 110 or embedded controller 120 as if the firmware was stored in a flash memory or other NV memory elsewhere in system 100. However, the access control mechanisms of TPM 140 provide a higher level of security than that of a flash memory or other NV memory elsewhere in system 100.

Therefore, NV memory 150 may be used to securely store boot policy and BIOS firmware that may be executed by processor 100 in response to memory transactions initiated by control logic 118. Similarly, NV memory 150 may be used to securely store firmware to be executed by embedded controller 120 or other system components.

Command register 152 may be used to further control the mapping of the address received at I/O port 141 to a location in NV memory 150. For example, command register 152 may include bits or fields to be used to enable mapping by mapping structure 151 and/or to direct an access to recovery BIOS instead of standard BIOS.

Figure 2 illustrate method 200 for using a TPM for boot policy and secure firmware according to an embodiment of the present invention. The description of Figure 2 may refer to elements of Figure 1, but method 200 and other method embodiments of the present invention are not intended to be limited by these references.

In box 210, TPM 140 is configured with standard BIOS and recovery BIOS stored in NV memory 150. In box 212, mapping structure 151 is configured to map the system memory addresses for BIOS to the corresponding locations in NV memory 150 where standard BIOS and recovery BIOS have been stored. In box 214, command register 152 is configured to direct BIOS accesses to standard BIOS instead of recovery BIOS.

In box 220, system 100 is powered on. In box 222, control logic 118 directs instruction hardware 112 to fetch the first line of boot code. In box 224, interface 116 unit issues a memory transaction on processor bus 132 to fetch the first line of boot code. In box 226, bridge 130 converts the transaction to a MMIO transaction on serial bus 134. In box 228, I/O port 141 receives the transaction.

In box 230, mapping structure 151 maps the I/O address to a location in NV memory 150 at which the first line of boot code in standard BIOS has been stored. In box 232, the first line of boot code is returned to instruction hardware 111 through I/O port 141, serial bus 134, bridge 130, processor bus 132, and interface 116 unit. In box 234, execution hardware 112 executes the first line of standard boot code. In box 236, processor 100 continues executing standard boot code from TPM 140.

In box 240, processor 110 issues a write transaction to command register 152 to enable booting from recovery BIOS instead of standard BIOS. In box 242, the transaction is received

by I/O port 141. In box 244, command register 152 is programmed to direct an access to the first line of boot code to the NV memory location in which recovery BIOS has been stored.

In box 250, system 100 experiences an event which requires a reboot. In box 252, control logic 118 directs instruction hardware 112 to fetch the first line of boot code. In box 254, interface unit 116 issues a memory transaction on processor bus 132 to fetch the first line of boot code. In box 256, bridge 130 converts the transaction to a MMIO transaction on serial bus 134. In box 258, I/O port 141 receives the transaction.

In box 260, mapping structure 151 maps the I/O address to a location in NV memory 150 at which the first line of boot code in recovery BIOS has been stored. In box 262, the first line of recovery boot code is returned to instruction hardware 111 through I/O port 141, serial bus 134, bridge 130, processor bus 132, and interface unit 116. In box 264, execution hardware 112 executes the first line of recovery boot code. In box 266, processor 100 continues executing recovery boot code from TPM 140.

Within the scope of the present invention, the method illustrated in Figure 2 may be performed in a different order, with illustrated boxes omitted, with additional boxes added, or with a combination of reordered, omitted, or additional boxes.

Thus, apparatuses, methods, and systems for using a TPM for boot policy and secure firmware have been disclosed. While certain embodiments have been described, and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative and not restrictive of the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art upon studying this disclosure. In an area of technology such as this, where growth is fast and further advancements are not easily foreseen, the disclosed embodiments may be readily modifiable in arrangement and detail as facilitated by enabling technological advancements without departing from the principles of the present disclosure or the scope of the accompanying claims.

Claim

What is claimed is:

1. A trusted platform module comprising:

non-volatile memory;

a port to receive an input/output transaction from a serial bus, the transaction including a system memory address in the address space of a processor; and

a mapping structure to map the system memory address to a first location in non-volatile memory.

2. The trusted platform module of claim 1, wherein the non-volatile memory is to store basic input/output system code at the first location.

3. The trusted platform module of claim 2, further comprising a command register to direct the input/output transaction to a second location in non-volatile memory instead of the first location.

4. The trusted platform module of claim 3, wherein the non-volatile memory is to store standard basic input/output system code in the first location.

5. The trusted platform module of claim 4, wherein the non-volatile memory is to store recovery basic input/output system code in the second location.

6. A method comprising:

receiving, by a trusted platform module, a first input/output transaction from a serial bus, the first input/output transaction including a first system memory address in the address space of a processor; and

mapping, by the trusted platform module, the first system memory address to a first location in non-volatile memory of the trusted platform module.

7. The method of claim 6, further comprising storing basic input/output system code at the first location.

8. The method of claim 7, further comprising configuring a command register in the trusted platform module to direct the first input/output transaction to a second location in the non-volatile memory instead of the first location.

9. The method of claim 8, further comprising storing standard basic input/output system code at the first location.

10. The method of claim 9, further comprising storing recovery basic input/output system code at the second location.

11. The method of claim 6, further comprising configuring a mapping structure in the trusted platform module to map the first system memory address to the first location in the non-volatile memory.

12. The method of claim 11, further comprising:

issuing, by the processor, a memory transaction to boot a system; and

converting, by a bridge, the memory transaction to the first input/output transaction.

13. The method of claim 8, further comprising issuing, by a processor, a write transaction to configure the command register.

14. The method of claim 7, further comprising storing embedded controller firmware at a third location in the non-volatile memory.

15. The method of claim 14, further comprising:

issuing, by an embedded controller, a memory transaction including a second system memory address in the address space of the processor;

converting, by a bridge, the memory transaction to a second input/output transaction; and

mapping, by the trusted platform module, the second system memory address to the third location.

16. A system comprising:

a processor including

instruction hardware to fetch a first instruction from a first system memory address in the address space of the processor, and

an interface unit to issue a first memory transaction including the first system memory address; and

a trusted platform module including:

non-volatile memory,

a port to receive a first input/output transaction from a serial bus, the transaction including the first system memory address, and

a mapping structure to map the first system memory address to a first location in non-volatile memory.

17. The system of claim 16, further comprising a bridge to convert the first memory transaction to the first input/output transaction.

18. The system of claim 16, wherein the first instruction is the boot the system.

19. The system of claim 16, further comprising an embedded controller to issue a second memory transaction including a second system memory address in the address space of the processor, wherein the mapping structure is also to map the second system memory address to a second location in non-volatile memory.

20. The system of claim 19, wherein the first location is to store basic input/output system code and the second location is to store embedded controller firmware.

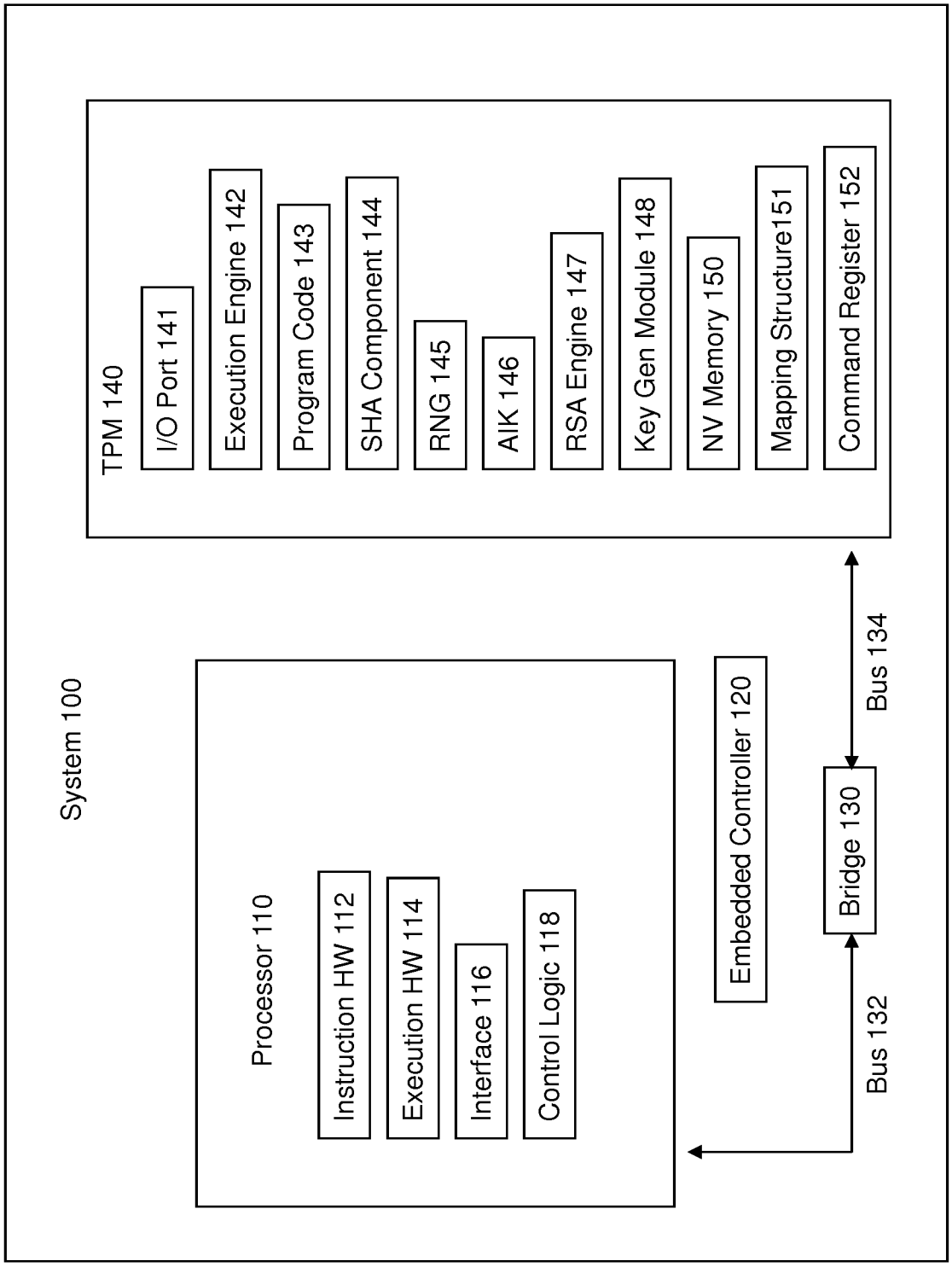
System 100
TPM 140
I/O Port 141
Execution Engine 142
Program Code 143
SHA Component 144
RNG 145
AIK 146
RSA Engine 147
Key Gen Module 148
NV Memory 150
Mapping Structure151
Command Register 152
Processor 110
Instruction HW 112
Execution HW 114
Interface 116
Control Logic 118
Embedded Controller 120
Bridge 130
Bus 132
Bus 134

FIGURE 1

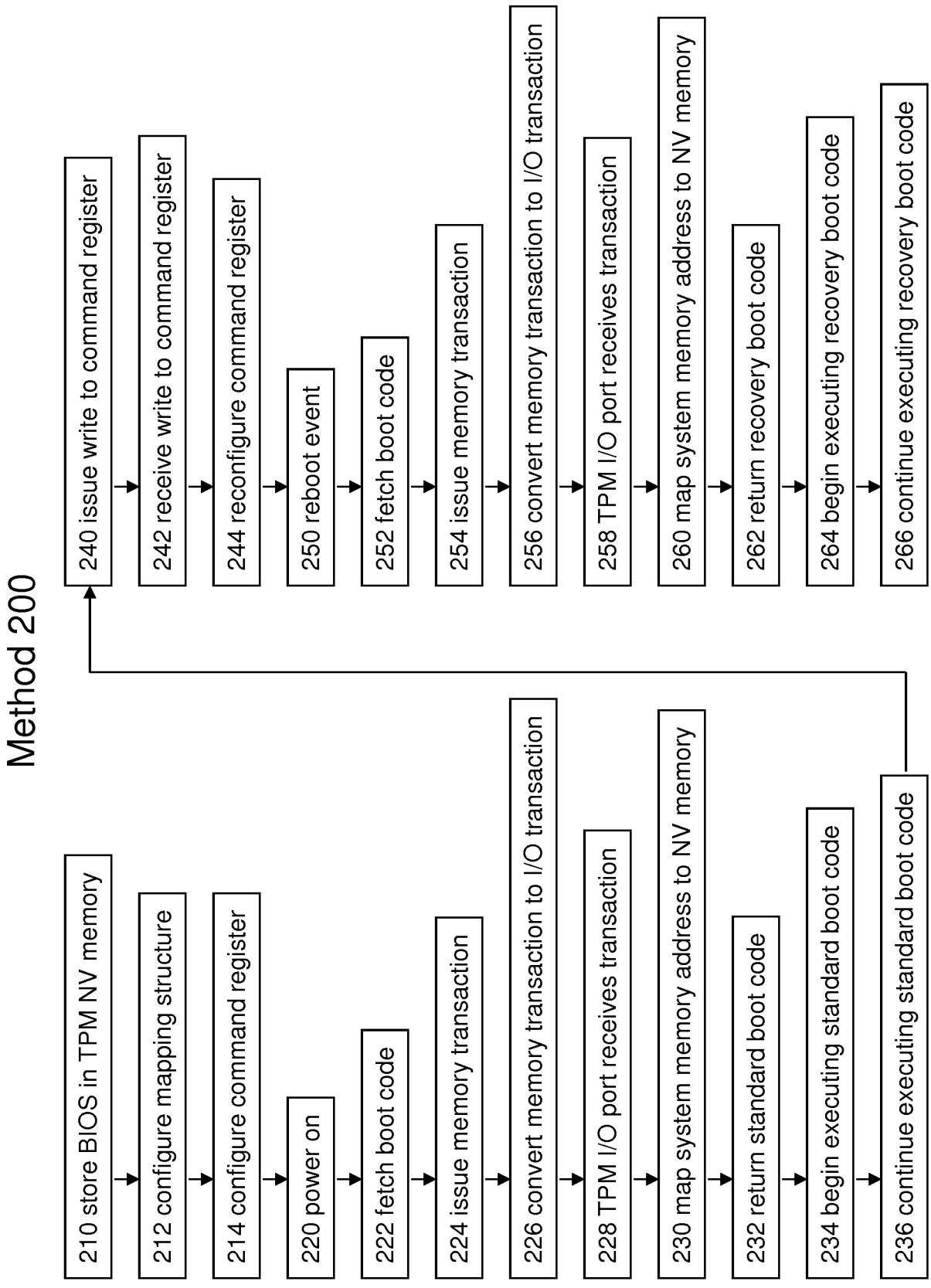
Method 200
210 store BIOS in TPM NV memory
212 configure mapping structure
214 configure command register
220 power on
222 fetch boot code
224 issue memory transaction
226 convert memory transaction to I/O transaction
228 TPM I/O port receives transaction
230 map system memory address to NV memory
232 return standard boot code
234 begin executing standard boot code
236 continue executing standard boot code
240 issue write to command register
242 receive write to command register
244 reconfigure command register
250 reboot event
252 fetch boot code
254 issue memory transaction
256 convert memory transaction to I/O transaction
258 TPM I/O port receives transaction
260 map system memory address to NV memory
262 return recovery boot code
264 begin executing recovery boot code
266 continue executing recovery boot code

FIGURE 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US201 1/068078

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/22(2006.01)i, G06F 13/14(2006.01)1, G06F 9/24(2006.01)1

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 21/22; G06F 12/14; H04L 29/06; G06F 21/00; H04L 009/00

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

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

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

eKOMPASS(KIPO internal) & Keywords: boot, secure, firmware, trust, platform, module, non-volatile, memory, address, map, basic imput output system, command, register

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2009-0158419 A1 (B0YCE KEVIN GERARD) 18 June 2009 See abstract; paragraphs [55] - [72]; figures 3, 4.	1,2,6,7,11,16
A	See the whole document.	3-5,8-10,12-15,17-20
Y	US 2010-0281273 A1 (LEE RUBY B. et al.) 04 November 2010 See abstract; paragraphs [10] - [60]; figures 3-5.	1,2,6,7,11,16
A	See the whole document.	3-5,8-10,12-15,17-20
A	US 2005-0149729 A1 (ZIMMER VINCENT J. et al.) 07 July 2005 See abstract; paragraphs [42] - [51]; figure 4.	1-20
A	US 2011-0191580 A1 (IFT0DE LIVIU et al.) 04 August 2011 See abstract; paragraphs [86] - [100]; figure 3.	1-20
A	US 7702907 B2 (VAHA-SIPILA ANTTI et al.) 20 April 2010 See abstract; column 6, line 5 - column 7, line 58; figures 6-9.	1-20

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

☒ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
31 MAY 2012 (31.05.2012)	**28 JUNE 2012 (28.06.2012)**
Name and mailing address of the ISA/KR Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea Facsimile No. 82-42-472-7140	Authorized officer Shin Sang Gil Telephone No. 82-42-481-8480

Form PCT/ISA/210 (second sheet) **(My** 2009)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2011/068078

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009-0158419 A1	18.06.2009	CA 2625274 A1	13.06.2009
US 2010-0281273 A1	04. 11.2010	None	
US 2005-0149729 A1	07.07.2005	None	
US 2011-0191580 A1	04. 08.2011	US 7930733 B1	19. 04. 2011
US 7702907 B2	20.04. 2010	US 2006-00752 16 A1	06. 04. 2006
		WO 2006-038082 A1	13.04. 2006