



- (51) **International Patent Classification:**
G06F 21/20 (2006.01) **G06F 11/08** (2006.01)
G06F 21/22 (2006.01)

(21) **International Application Number:**
PCT/US201 1/068281

(22) **International Filing Date:**
31 December 201 1 (31.12.201 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, MS: RNB-4-150, Santa Clara, California 95052 (US).

(72) **Inventors; and**

(75) **Inventors/Applicants (for US only):** **DATTA, Shamanna M.** [US/US]; 532 NE Lenox Street, Hillsboro, Oregon 97124 (US). **PARTHASARATHY, Rajesh S.** [IN/US]; 1209 NE Creeksedge Dr., Hillsboro, Oregon 97124 (US). **NATU, Mahesh S.** [US/US]; 202 Peppermint Tree Terrace, Unit 1, Sunnyvale, California 94086 (US). **BINNS, Frank** [GB/US]; 2420 NW Pinnacle Drive, Portland, Oregon 97229 (US). **KUMAR, Mohan J.** [US/US]; 18680 SW Marko Lane, Aloha, Oregon 97007 (US).

(74) **Agents:** **HUNTER, Spencer K.** et al; **BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP**, 1279 Oakmead Parkway, Sunnyvale, California 94085 (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).

Published:
with international search report (Art. 21(2))

Published:

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

- (54) Title:** PROCESSOR THAT DETECTS WHEN SYSTEM MANAGEMENT MODE ATTEMPTS TO REACH PROGRAM CODE OUTSIDE OF PROTECTED SPACE

- (57) Abstract:** A method is described that includes detecting that a memory access of system management mode program code is attempting to reach program code outside of a protected region of memory by comparing a target memory address of a memory access instruction of the system management program code again information that defines confines of the protection region. The method also includes raising an error signal in response to the detecting.

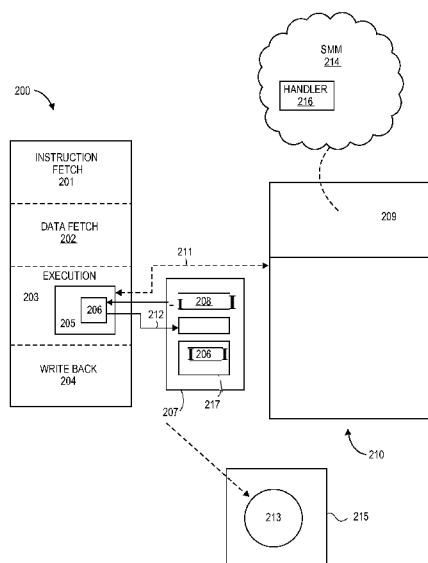


FIG. 2

PROCESSOR THAT DETECTS WHEN SYSTEM MANAGEMENT MODE ATTEMPTS TO
REACH PROGRAM CODE OUTSIDE OF PROTECTED SPACE

Field of Invention

5 The field of invention pertains to the computing sciences generally, and, more specifically, to a processor that detects when system management mode attempts to reach program code outside of protected space.

Background

10 System management mode (SMM) is an operating mode of a computing system in which normal execution (including operation of the operating system (OS)) is suspended, and special separate software (usually firmware or a hardware-assisted debugger, hereinafter referred to as "SMM code") is executed in a high-privilege mode. Here, the SMM code is special code that, ideally, fully comprehends the complete hardware details of the particular computing system that
15 it runs on.

 Examples of possible SMM functions include: i) centralized system configuration (such as the dedicated configuration of a specific computer); ii) handling of system events like memory or chipset errors; iii) security functions, such as flash device lock down or the forwarding of calls to a Trusted Platform Module (TPM); iv) system safety management functions, such as computer
20 shutdown upon detection of a high CPU temperature, turning fans on/off, etc.; and, v) power management functions such as deep sleep power management and management of voltage regulator modules.

 Note that the awareness and reach of the SMM is far reaching for a particular system. That is, typically, the SMM is permitted to have unrestricted access into any aspect of the
25 computer that it runs on. A significant security risk, therefore, is that the SMM might be "highjacked" or otherwise compromised by some form of malware or unwanted code. If the security of the SMM were to be breached, the malware could potentially disrupt or infect normal system operation of any/all components within the system.

 Typically, the SMM code is kept in a "highly privileged" region of memory. Part of the
30 definition of the highly privileged and secure aspect of the SMM code is that no other code is permitted to access it within its special region of memory, nor is the SMM supposed to execute code that is stored outside the highly privileged region of memory. Thus, if malware is to attack the SMM, in all likelihood it will be a consequence of the SMM running code or at least making a call to code that is outside of the highly privileged region of memory. Upon this event,

malware stored outside the protected region of memory can be incorporated into the operation of the SMM thereby compromising its security.

Unfortunately it is becoming more and more difficult to keep the SMM code within the confines of the protected region of memory. The difficulty stems not only from the increasing sophistication of the SMM code (as a function of increasingly complex hardware platforms and associated features), but also, the reliance on, within the SMM, of OEM code provided by OEM manufacturers of the different components that the system is composed of.

Fig. 1 shows a typical transition from normal operating mode into SMM mode. Upon the detection of an event 101 that is supposed to trigger the SMM code (such as a configuration event, a power management, etc.), a processor will save its state 102. After the processor state is saved, SMM begins execution out the protected region of memory 103.

Brief Description of the Drawings

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

Fig. 1 shows a prior art SMM start up sequence;

Fig. 2 shows a system that can detect when SMM attempts to reach program code that is outside of protected space;

Fig. 3 shows an SMM start sequence that enables a function that detects when SMM attempts to reach program code that is outside of protected space;

Fig. 4 shows an embodiment of a computing system.

Description

A solution to the problem of not being able to sufficiently control when or how SMM code might attempt to reach code outside the protected region of memory, is to build into the hardware of the processor a mechanism for detecting and reporting an attempt by the SMM code to reach program code that is outside the protection memory space.

Fig. 2 shows an embodiment of a generic processor pipeline 200. The processor includes an instruction fetch stage 201, a data fetch stage 202, an execution stage 203 and a write back stage 204. As is known in the art, the execution stage 203 includes various functional units that execute the different instructions supported by the pipeline 200 and, thereby, the processor. One or more of these execution units is typically a memory access unit 205, or simply, memory unit, whose purpose is to fetch data or instructions from memory.

In a typical case, an instruction is presented to the memory unit 205 having an operand that specifies an address in memory 210 where an item of data or an instruction is located (or a

"page" of memory 210 where the item of data or instruction is located). The memory unit 205 typically includes a hardware table-walk or page-walk unit 206 that coordinates the read of the information from memory 210 (e.g., by issuing or overseeing a read transaction 211 directed to system memory 210 for the desired information).

5 In one embodiment of the improved approach described herein, control register space 207 keeps information 208 that defines the confines of the highly protected region of memory 209 within system memory 210. In an embodiment, information 208 is contained within two registers in control register space 207 (SMRR_Range and SMRR_Mask) that specify the confines in protected space 209.

10 Information 208 is made available to the hardware table walk unit 206 so that the hardware table walk unit 206 can comprehend the confines of the protected space 209. In keeping with the theory that malware might corrupt the SMM code 214 if the SMM code 214 attempts to reach code that is outside the protected memory space 209, according to one embodiment of the approach described herein, when the SMM code 214 is running, the hardware
15 table walk unit 206 includes or is otherwise associated with logic circuitry that compares the address operand of a memory address instruction that seeks to fetch program code against the address range or other definition of the confines of the protected space 209 contained in information 208 and received from the control register space 207.

If the address targeted by the instruction is determined to be outside the confines of the
20 protected space 209, the hardware table walk unit 206 generates a signal 212 (in one embodiment, an unrecoverable machine signal check (MCHECK)) to the hardware that eventually triggers appropriate handler program code 216 (e.g., an MCHECK handler) within the SMM 214. In one embodiment, the signal 212 is implemented as a special
Access_Violation_Detected bit that is set in control register space 207 upon detection of the
25 offending code address.

As part of the error reporting process performed by logic circuitry in the hardware in response to the detection of an out of range address, code violation information 213 (e.g., that identifies the address that was attempted to be reached outside the protected space, the nature of the instruction that attempted the offending memory access, state information of the pipeline
30 including the state of the program counter, etc.) is dumped into a storage resource 215 (e.g., register space, part of memory 210, disk) and an address 216 where the code violation information 213 is stored in storage resource 215 is written into machine check register banks 217. The handler 216 looks to the machine check register banks 217, fetches the address 216 and retrieves the code violation information 213. Here, the handler 216 may store the
35 information or aspects thereof to some other storage resource including as a possibility remote

storage (e.g., storage connected to the computing system through a network), report the incident to a user, etc..

According to one embodiment, the initial error signal 212 is only generated when program code is attempted to be accessed by an offending SMM memory access instruction (and not data). This permits SMM 214 to reach any item of data with out triggering an alarm.

Also, according to another or combined embodiment, the signal 212 will not be generated if an offending memory access instruction stems from a predicted branch. Here, as is known in the art, some processor pipelines have the intelligence to "guess" along which path a program's execution flow will be directed. As part of the guess, the memory execution units may "pre-fetch" program code instructions that are presumed to be along the path that the program code will flow along. If the processor guesses incorrectly, the fetched program code is flushed. If the processor guesses correctly, the program code is executed.

In order to prevent a false alarm in the case where the processor pre-fetches offending instruction(s) outside the protected space 209 as a consequence of prediction but the processor guesses wrong such that the offending code is ultimately flushed, the hardware does not generate a signal 212 merely when offending code is pre-fetched as a consequence of prediction. Rather, in an embodiment, the hardware will generate the signal 212 in response to both: i) having flagged that the offending code was pre-fetched as a consequence of prediction; and, ii) the "guess" made by the hardware is ultimately deemed correct resulting in the offending code being in the path of execution. Upon both of these conditions being reached the signal 212 will be raised and error handling proceeds as described above.

In another embodiment, the logic that detects an offending address and generates the error signal 212 is enabled by the setting of a bit in hardware upon the SMM being newly entered. For example, referring to Fig. 3, upon a condition being detected 301 that triggers entry of the SMM, after the processor state is saved 302 (e.g., such that the processor is no longer currently executing any threads), the firmware BIOS enables the memory access trap function 303 and the SMM proceeds to commence operation 304. Depending on implementation the SMM may be considered to be part of the firmware BIOS. By making the enable bit only settable by the BIOS firmware, malware is not permitted to reset the bit so as to disable the function. In a further embodiment, when the SMM terminates and the processor state is restored, as part the SMM termination/processor bring-back, the enablement bit is cleared such that the function is disabled.

A processing core having the functionality described above can be implemented into various computing systems as well. Fig. 4 shows an embodiment of a computing system (e.g., a computer). The exemplary computing system of Fig. 4 includes: 1) one or more processing

cores 401 that may be designed to trap for SMM code that attempts to reach program code outside of protected space; 2) a memory control hub (MCH) 402; 3) a system memory 403 (of which different types exist such as DDR RAM, EDO RAM, etc.); 4) a cache 404; 5) an I/O control hub (ICH) 405; 6) a graphics processor 406; 7) a display/screen 407 (of which different types exist such as Cathode Ray Tube (CRT), flat panel, Thin Film Transistor (TFT), Liquid Crystal Display (LCD), DPL, etc.) one or more I/O devices 408.

The one or more processing cores 401 execute instructions in order to perform whatever software routines the computing system implements. The instructions frequently involve some sort of operation performed upon data. Both data and instructions are stored in system memory 403 and cache 404. Cache 404 is typically designed to have shorter latency times than system memory 403. For example, cache 404 might be integrated onto the same silicon chip(s) as the processor(s) and/or constructed with faster SRAM cells whilst system memory 403 might be constructed with slower DRAM cells. By tending to store more frequently used instructions and data in the cache 404 as opposed to the system memory 403, the overall performance efficiency of the computing system improves.

System memory 403 is deliberately made available to other components within the computing system. For example, the data received from various interfaces to the computing system (e.g., keyboard and mouse, printer port, LAN port, modem port, etc.) or retrieved from an internal storage element of the computing system (e.g., hard disk drive) are often temporarily queued into system memory 403 prior to their being operated upon by the one or more processor(s) 401 in the implementation of a software program. Similarly, data that a software program determines should be sent from the computing system to an outside entity through one of the computing system interfaces, or stored into an internal storage element, is often temporarily queued in system memory 403 prior to its being transmitted or stored.

The ICH 405 is responsible for ensuring that such data is properly passed between the system memory 403 and its appropriate corresponding computing system interface (and internal storage device if the computing system is so designed). The MCH 402 is responsible for managing the various contending requests for system memory 403 access amongst the processor(s) 401, interfaces and internal storage elements that may proximately arise in time with respect to one another.

One or more I/O devices 408 are also implemented in a typical computing system. I/O devices generally are responsible for transferring data to and/or from the computing system (e.g., a networking adapter); or, for large scale non-volatile storage within the computing system (e.g., hard disk drive). ICH 405 has bi-directional point-to-point links between itself and the observed I/O devices 408.

Processes taught by the discussion above may be performed with program code such as machine-executable instructions that cause a machine that executes these instructions to perform certain functions. In this context, a "machine" may be a machine that converts intermediate form (or "abstract") instructions into processor specific instructions (e.g., an abstract execution
5 environment such as a "virtual machine" (e.g., a Java Virtual Machine), an interpreter, a Common Language Runtime, a high-level language virtual machine, etc.)), and/or, electronic circuitry disposed on a semiconductor chip (e.g., "logic circuitry" implemented with transistors) designed to execute instructions such as a general-purpose processor and/or a special-purpose processor. Processes taught by the discussion above may also be performed by (in the
10 alternative to a machine or in combination with a machine) electronic circuitry designed to perform the processes (or a portion thereof) without the execution of program code.

It is believed that processes taught by the discussion above may also be described in source level program code in various object-orientated or non-object-orientated computer programming languages (e.g., Java, C#, VB, Python, C, C++, J#, APL, Cobol, Fortran, Pascal,
15 Perl, etc.) supported by various software development frameworks (e.g., Microsoft Corporation's .NET, Mono, Java, Oracle Corporation's Fusion, etc.). The source level program code may be converted into an intermediate form of program code (such as Java byte code, Microsoft Intermediate Language, etc.) that is understandable to an abstract execution environment (e.g., a Java Virtual Machine, a Common Language Runtime, a high-level language virtual machine, an
20 interpreter, etc.) or may be compiled directly into object code.

According to various approaches the abstract execution environment may convert the intermediate form program code into processor specific code by, 1) compiling the intermediate form program code (e.g., at run-time (e.g., a JIT compiler)), 2) interpreting the intermediate form program code, or 3) a combination of compiling the intermediate form program code at run-time
25 and interpreting the intermediate form program code. Abstract execution environments may run on various operating systems (such as UNIX, LINUX, Microsoft operating systems including the Windows family, Apple Computers operating systems including MacOS X, Sun/Solaris, OS/2, Novell, etc.).

An article of manufacture may be used to store program code. An article of manufacture
30 that stores program code may be embodied as, but is not limited to, one or more memories (e.g., one or more flash memories, random access memories (static, dynamic or other)), optical disks, CD-ROMs, DVD ROMs, EPROMs, EEPROMs, magnetic or optical cards or other type of machine-readable media suitable for storing electronic instructions. Program code may also be downloaded from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by

way of data signals embodied in a propagation medium (e.g., via a communication link (e.g., a network connection)).

In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and
5 changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

1. A method, comprising:

5 detecting that a memory access of system management mode program code is attempting to reach program code outside of a protected region of memory by comparing a target memory address of a memory access instruction of said system management program code again information that defines confines of said protection region; and,
 raising an error signal in response to said detecting.

10 2. The method of claim 1 wherein said information is stored in control register space.

 3. The method of claim 2 wherein said error signal includes setting a value in said control register space.

15 4. The method of claim 1 wherein said method further comprises

 not raising an error signal even though a second memory access instruction for program code targets memory space outside of said protected region because said second memory access instruction is being fetched as a consequence of speculation.

20 5. The method of claim 4 further comprising raising an error signal because said speculation is deemed to have been correct.

 6. The method of claim 1 further comprising storing second information pertaining to said
25 memory access instruction and storing an address where said second information is stored.

 7. The method of claim 6 wherein said address is stored in control register space.

 8. A semiconductor chip, comprising:

30 an instruction execution pipeline having logic circuitry to detect that a memory access of system management mode program code is attempting to reach program code outside of a protected region of memory by comparing a target memory address of a memory access instruction of said system management program code again information that defines confines of said protection region.

9. The semiconductor chip of claim 8 wherein said logic circuitry is coupled to control register space that stores said information.

10. The semiconductor chip of claim 9 wherein said information defines a range of said protected region.

11. The semiconductor chip of claim 8 wherein said logic circuitry is within a memory unit of said pipeline.

12. The semiconductor chip of claim 10 wherein said logic circuitry is part of hardware table walk logic circuitry within said memory unit.

13. The semiconductor chip of claim 8 wherein said logic writes an error signal to control register space in response to detection that said memory access is attempting to reach said program code.

14. A machine readable medium containing program code that when processed by a processor causes the following method to be performed:

reading information descriptive of an error condition that was automatically flagged in processor hardware because system mode program code was detected attempting to reach program code outside of a protected region of memory.

15. The machine readable medium of claim 14 wherein said information includes an address of said program code that was attempted to be reached.

16. The machine readable medium of claim 14 wherein said method further comprises fetching an address where said information is stored as a consequence of said error condition having been flagged.

17. The machine readable medium of claim 16 further comprising fetching said address from control register space of said processor hardware.

18. The machine readable medium of claim 14 further comprising presenting an indication of said error condition to a user.

19. The machine readable medium of claim 14 further comprising storing said information to a second storage location.

20. A computing system, comprising:

- 5 a processor having logic circuitry to detect that a memory access of system management mode program code is attempting to reach program code outside of a protected region of memory by comparing a target memory address of a memory access instruction of said system management program code against information that defines confines of said protection region; and,
- 10 handler program code within said protected region of memory, said handler program code to read information descriptive of an error condition flagged by said logic circuitry.

1/4

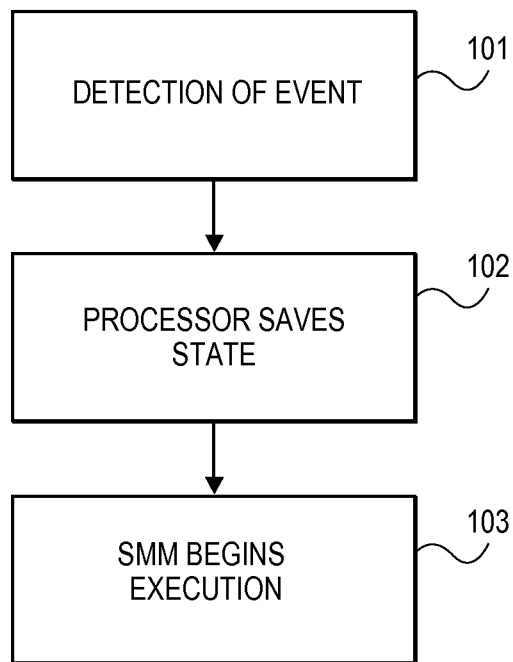
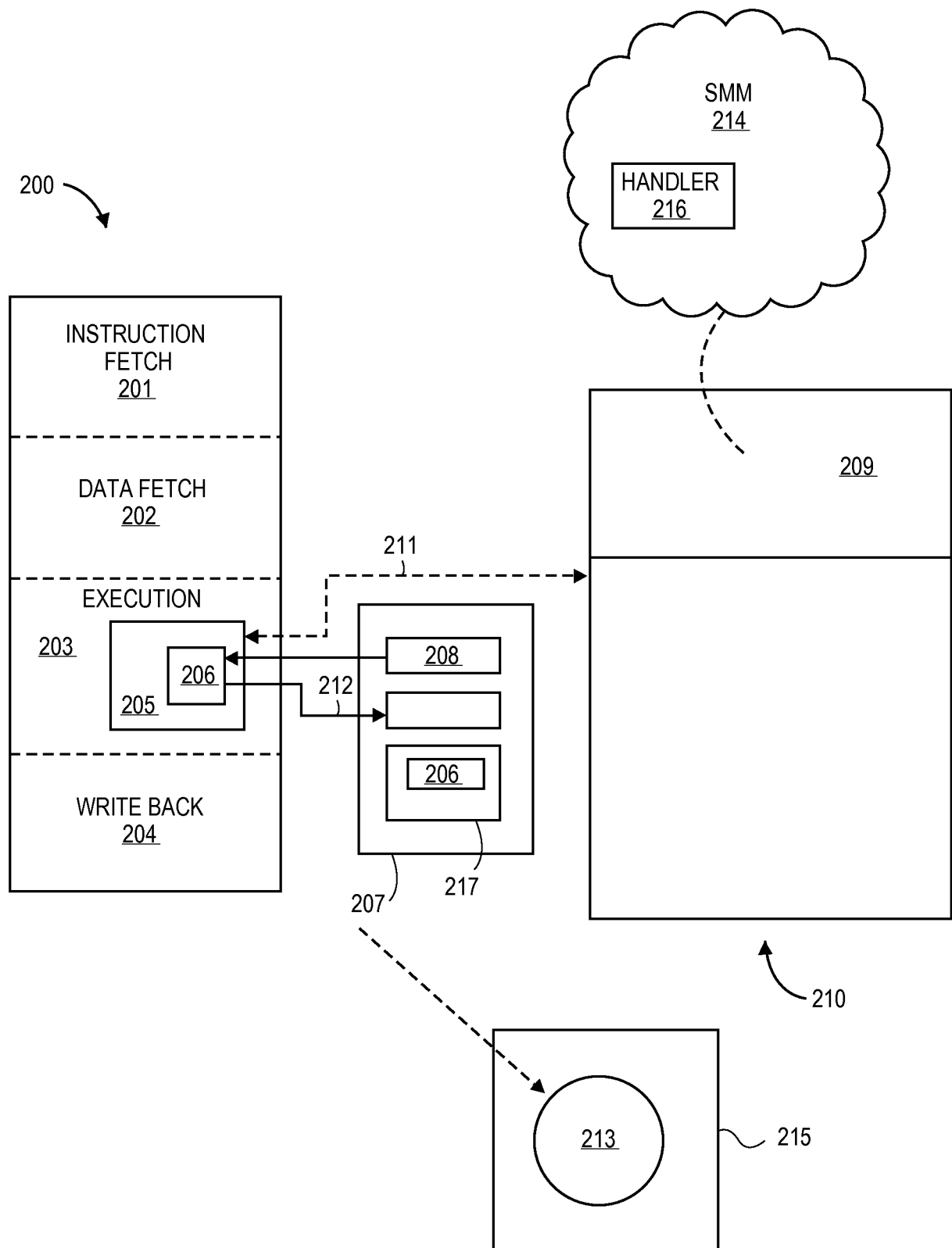
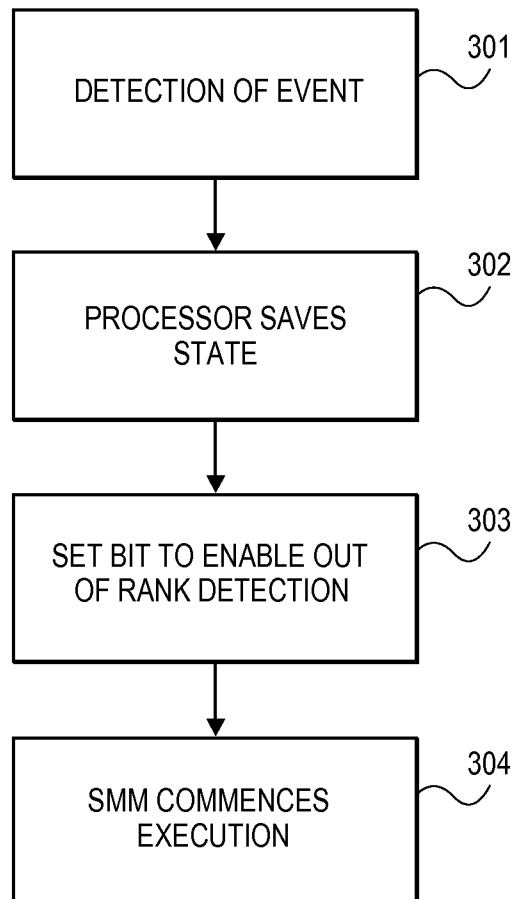


FIG. 1
(PRIOR ART)

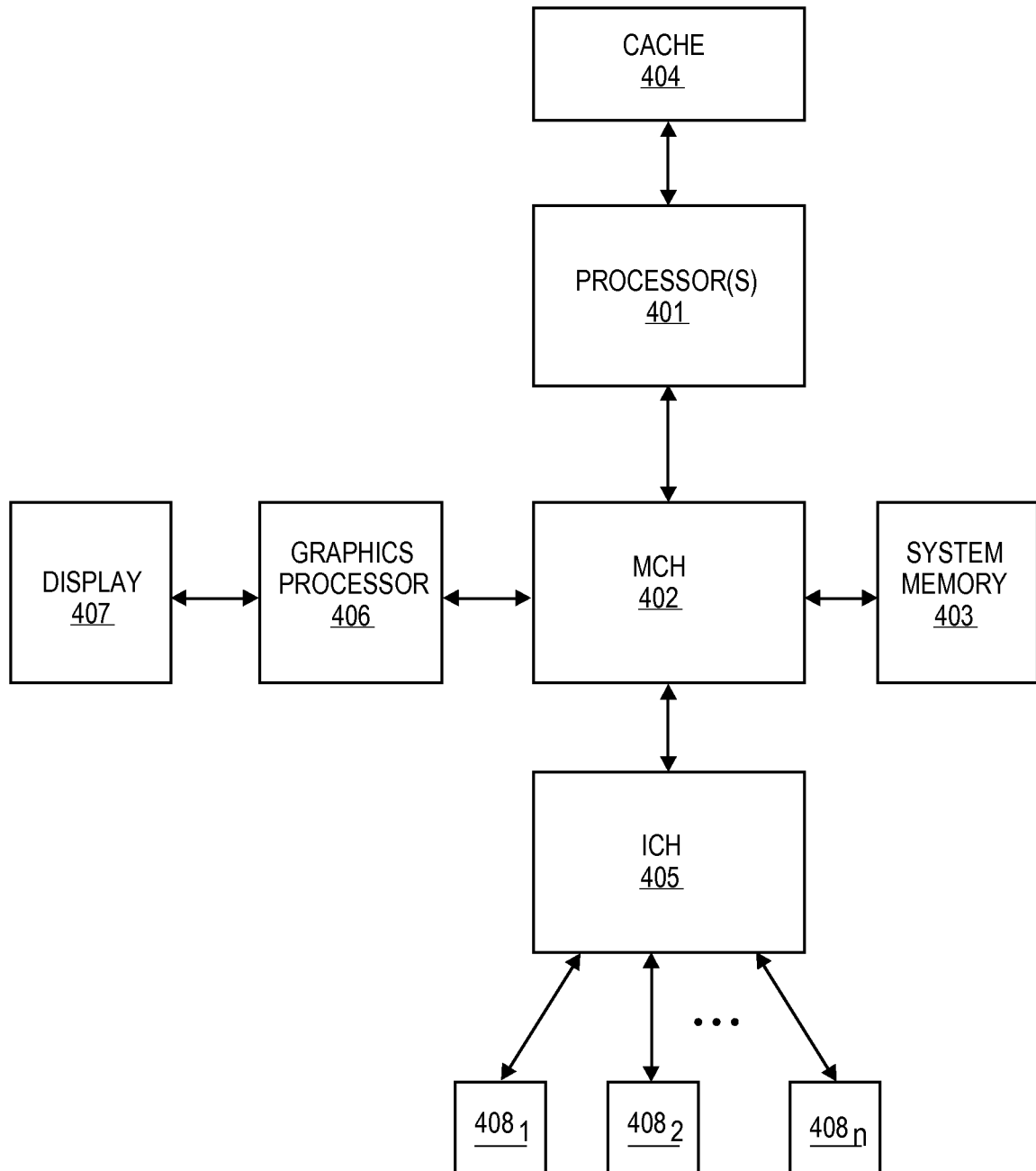
2/4

**FIG. 2**

3/4

**FIG. 3**

4/4

**FIG. 4**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2011/068281**A. CLASSIFICATION OF SUBJECT MATTER****G06F 21/20(2006.01)i, G06F 21/22(2006.01)1, G06F 11/08(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 21/20; G06F 11/30; G06F 12/00; H04L 9/00; G06F 21/00; G06F 7/04; G06F 11/00

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) &

Keywords:(detect*,monitor*,check*,identif*,watch*,DISCRIMINAT*,DISTINGUISH*,IDENTIF*,DISCERN*,SUPERVIS*)+(SMM,sy
stem n/1 management n/1 mode system n/1 management system n/1 control n/wpr n/1 manao* n/wpr n/1**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2009-0119748 A1 (YA0 JIEWEN et al.) 07 May 2009 See abstract, paragraph [0053], claims 1,2 and figure 7	1-20
A	US 7003676 B1 (Frederick D. Weber, et al.) 21 February 2006 See abstract, column 3, lines 15-28 and claims 21,22	1-20
A	US 7496966 B1 (MCGRATH KEVIN J. et al.) 24 February 2009 See abstract, claim 1 and figure 4	1-20
A	US 2009-0320128 A1 (PANT AL0K et al.) 24 December 2009 See abstract, paragraph [0006] and claim 1	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

17 SEPTEMBER 2012 (17.09.2012)

Date of mailing of the international search report

17 SEPTEMBER 2012 (17.09.2012)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan
City, 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

UHM, In Kwon

Telephone No. 82-42-481-5712



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US201 1/068281

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009-0 119748 A1	07.05.2009	US 2009-0063835 A1 US 7827371 B2	05.03.2009 02.11.2010
US 7003676 B1	21.02.2006	AU 2002-25059 1 A8 AU 2002-254625 A8 CN 100533330 C CN 101118583 A CN 101118583 CO CN 1520537 A CN 1520537 CO CN 1524206 A CN 1524206 CO EP 1410 143 A2 EP 1410 143 B1 EP 1428095 A2 JP 2005-50 1313 A JP 2005-5155 17 A KR 10-0902398 B1 KR 10-0938924 B1 KR 10-2003-0094396 A TW 1222285B TW 1228889B TW 222285 A TW 222285 B TW 228889 A TW 228889 B US 2003-002878 1 A1 US 2003-0041248 A1 US 6823451 B1 US 68323 17 B1 US 6862641 B1 US 6968460 B1 US 7007300 B1 US 7043581 B1 US 7065654 B1 US 7 149854 B2 US 72 16362 B1 WO 02-093335 A2 WO 02-093335 A3 WO 02-093336 A2 WO 02-093336 A3	25.11.2002 25.11.2002 26.08.2009 06.02.2008 06.02.2008 11.08.2004 11.08.2004 25.08.2004 25.08.2004 21.04.2004 26.10.2005 16.06.2004 13.01.2005 26.05.2005 11.06.2009 27.01.2010 11.12.2003 11.10.2004 01.03.2005 11.10.2004 11.10.2004 01.03.2005 01.03.2005 06.02.2003 27.02.2003 23.11.2004 14.12.2004 01.03.2005 22.11.2005 28.02.2006 09.05.2006 20.06.2006 12.12.2006 08.05.2007 21.11.2002 21.11.2002 21.11.2002 21.11.2002
US 7496966 B1	24.02.2009	AU 2003-23 1070 A1 CN 100339780 C CN 16470 11 A DE 60322366 D1 EP 1495394 A2 EP 1495394 B1 JP 2005-528677 A	03.11.2003 26.09.2007 27.07.2005 04.09.2008 12.01.2005 23.07.2008 22.09.2005

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2011/068281Patent document
cited in search reportPublication
datePatent family
member(s)Publication
date

US 2009-0320128 A1

24. 12.2009

JP 2005-528677 T

22.09.2005

JP 4762494 B2

31.08.2011

US 7401358 B1

15.07.2008

WO 0309-0052A2

30. 10.2003

WO 0309-0052A3

12.08.2004

None