(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년06월11일
(11) 등록번호 10-0902398
(24) 등록일자 2009년06월04일

(51) Int. Cl.
G06F 1/00 (2006.01) *G11C 16/04* (2006.01)
(21) 출원번호 10-2003-7014630
(22) 출원일자 2003년11월10일
심사청구일자 2007년04월17일
번역문제출일자 2003년11월10일
(65) 공개번호 10-2003-0094401
(43) 공개일자 2003년12월11일
(86) 국제출원번호 PCT/US2002/011930
국제출원일자 2002년04월17일
(87) 국제공개번호 WO 2002/93335
국제공개일자 2002년11월21일
(30) 우선권주장
09/852,942 2001년05월10일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
US6188602 B1
US6154819 A
US6122732 A
전체 청구항 수 : 총 15 항

(73) 특허권자
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 94088-3453 서니베일 원 에이엠디 플레이스 메일 스톱68
(72) 발명자
웨버프레드릭디.
미국캘리포니아95125산호세세틀애비뉴1137
구리크데일이.
미국텍사스78738오스틴애스토리아드라이브11715
스트롱인제프리에스.
미국텍사스78731오스틴몬타나노르테7210
(74) 대리인
박장원

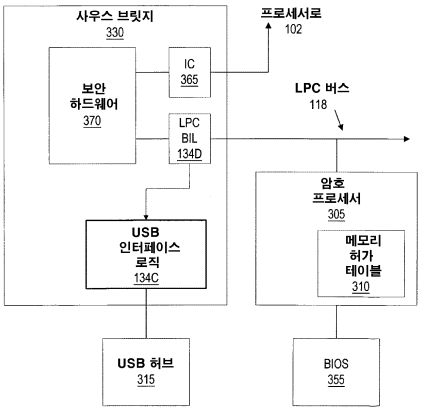
심사관 : 박상현

(54) 퍼스널 컴퓨터 메모리 장소를 위한 외부 로크 메커니즘

(57) 요 약

메모리 위치들에 대해 외부 로크 메커니즘을 제공하기 위한 방법 및 시스템이 개시된다. 메모리는 BIOS 데이터로 구성된 제1 복수의 저장 위치와; 제2 복수의 저장 위치를 포함한다. 제2 복수의 저장 위치는 오직 SMM에서 판독 가능한 제1 복수의 블록과, SMM 및 SMM을 제외한 적어도 하나의 동작 모드에서 판독 가능한 제2 복수의 블록들을 포함한다. 컴퓨터 시스템은 버스와, 상기 버스에 연결된 메모리와, 상기 버스를 통해 상기 메모리에 액세스하도록 연결된 디바이스를 포함한다. 메모리는 복수의 저장 위치를 포함하고, 이는 복수의 메모리 유닛(memory units)으로 나뉜다. 디바이스는 하나 이상의 상기 복수의 메모리 유닛에 대한 액세스를 제어하도록 구성된 하나 이상의 로크들을 포함한다.

대 표 도 - 도4



(30) 우선권주장
09/852,372 2001년05월10일 미국(US)
09/870,889 2001년05월30일 미국(US)

특허청구의 범위

청구항 1

삭제

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

삭제

청구항 7

컴퓨터 시스템으로서,

버스와;

상기 버스와 연결된 메모리 -상기 메모리는 복수의 저장 위치를 포함하고, 상기 복수의 저장 위치는 복수의 메모리 유닛으로 나뉨- 와; 그리고

상기 버스를 통해 상기 메모리에 액세스하도록 연결된 디바이스 -상기 디바이스는 상기 컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부를 결정하도록 구성되며-

를 포함하여 이루어지며,

상기 디바이스는, 상기 컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부의 결정에 근거하여 상기 복수 메모리 유닛중 하나 이상의 메모리 유닛으로의 액세스를 제어하도록 구성된 하나 이상의 로크들을 포함하며,

상기 로크들은 복수의 레지스터들을 포함하고, 상기 복수의 레지스터들 중 하나 이상의 레지스터 내의 하나 이상의 엔트리들은 상기 하나 이상의 메모리 유닛에 대한 액세스 제어 설정을 표시하며,

상기 복수의 레지스터들 중 적어도 하나는 메모리 블록들 중 하나를 위해 세 개의 로크 비트들을 저장하도록 구성되고,

상기 세개의 로크 비트들은 판독 로크 비트, 기록 로크 비트 및 엄중 잠금(lock-down) 비트를 포함하고, 그리고

상기 판독 로크 비트 및 상기 기록 로크 비트는 상기 엄중 잠금 비트가 세트되면 리셋될 때까지 영속하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 8

컴퓨터 시스템으로서,

버스와;

상기 버스와 연결된 메모리 -상기 메모리는 복수의 저장 위치를 포함하고, 상기 복수의 저장 위치는 복수의 메모리 유닛으로 나뉨- 와; 그리고

상기 버스를 통해 상기 메모리에 액세스하도록 연결된 디바이스 -상기 디바이스는 상기 컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부를 결정하도록 구성되며-

를 포함하여 이루어지며,

상기 디바이스는, 상기 컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부의 결정에 근거하여 상기 복수 메모리 유닛중 하나 이상의 메모리 유닛으로의 액세스를 제어하도록 구성된 하나 이상의 로크들을 포함하며,

상기 로크들은 복수의 레지스터들을 포함하고, 상기 복수의 레지스터들 중 하나 이상의 레지스터 내의 하나 이상의 엔트리들은 상기 하나 이상의 메모리 유닛에 대한 액세스 제어 설정을 표시하며,

상기 복수의 레지스터들의 적어도 하나는 8비트를 저장하도록 구성되고,

상기 8비트는 메모리 블록들 중 하나를 위해 세 개의 로크 비트들을 포함하고, 상기 메모리 블록들 중 또 다른 하나를 위해 또 다른 세 개의 로크 비트들을 포함하고,

상기 세개의 로크 비트들은 제1 판독 로크 비트, 제1 기록 로크 비트 및 제1 엄중 잠금 비트를 포함하고, 상기 제1 엄중 잠금 비트가 세트되면, 상기 제1 판독 로크 비트 및 상기 제1 기록 로크 비트는 리셋이 될 때 까지 영속하고, 그리고

상기 또 다른 세 개의 로크 비트들은 제2 판독 로크 비트, 제2 기록 로크 비트 및 제2 엄중 잠금 비트를 포함하고, 상기 제2 엄중 잠금 비트가 세트되면, 상기 제2 판독 로크 비트 및 상기 제2 기록 로크 비트는 리셋이 될 때 까지 영속하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 9

제 8항에 있어서,

상기 복수의 레지스터들 중 적어도 하나는 상기 제1 기록 로크 비트로서 비트 0으로, 상기 제1 엄중 잠금 비트로서 비트 1로, 상기 제1 판독 로크 비트로서 비트 2로, 상기 제2 기록 로크 비트로서 비트 4로, 상기 제2 엄중 잠금 비트로서 비트 5로 그리고 상기 제2 판독 로크 비트로서 비트 6으로 구성되는 것을 특징으로 하는 컴퓨터 시스템.

청구항 10

컴퓨터 시스템을 동작시키기 위한 방법으로서,

하나 이상의 메모리 어드레스에 대한 메모리 트랜잭션을 요청하는 단계와;

상기 하나 이상의 메모리 어드레스에 대한 로크 상태를 결정하는 단계와;

상기 하나 이상의 메모리 어드레스에 대한 상기 로크 상태를 리턴시키는 단계와;

상기 컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부를 결정하는 단계와;

상기 컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부의 결정에 근거하여, 상기 로크 상태가 상기 하나 이상의 메모리 어드레스에 대한 메모리 트랜잭션이 허용되지 않음을 표시하는 경우, 상기 하나 이상의 메모리 어드레스에 대한 로크 상태가 변경될 수 있는지 여부를 결정하는 단계와; 그리고

상기 하나 이상의 메모리 어드레스의 로크 상태가 변경될 수 있는 경우, 상기 메모리 트랜잭션을 허용하도록, 상기 하나 이상의 메모리 어드레스의 로크 상태를 변경시키는 단계를 포함하는 것을 특징으로 하는 컴퓨터 시스템을 동작시키기 위한 방법.

청구항 11

제 10항에 있어서,

상기 로크 상태를 결정하는 단계는 제1 로크 비트를 판독하는 것을 포함하고; 그리고

상기 로크 상태를 리턴시키는 단계는 상기 제1 로크 비트의 값을 리턴시키는 것을 포함하는 것을 특징으로 하는 컴퓨터 시스템을 동작시키기 위한 방법.

청구항 12

제 11항에 있어서,

상기 하나 이상의 메모리 어드레스에 대한 로크 상태가 변경될 수 있는지를 결정하는 단계는 제2 로크 비트를 판독하는 것을 포함하는 것을 특징으로 하는 컴퓨터 시스템을 동작시키기 위한 방법.

청구항 13

제 12항에 있어서,

상기 메모리 트랜잭션을 허용하도록 상기 하나 이상의 메모리 어드레스의 로크 상태를 변경시키는 단계는 상기 제1 로크 비트의 값을 변경시키는 것을 포함하는 것을 특징으로 하는 컴퓨터 시스템을 동작시키기 위한 방법.

청구항 14

하나 이상의 메모리 어드레스에 대한 메모리 트랜잭션을 요청하는 수단과;

상기 하나 이상의 메모리 어드레스에 대한 로크 상태를 결정하는 수단과;

상기 하나 이상의 메모리 어드레스에 대한 상기 로크 상태를 리턴시키는 수단과;

컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부를 결정하는 수단과;

상기 컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부의 결정에 근거하여, 상기 로크 상태가 상기 하나 이상의 메모리 어드레스에 대한 메모리 트랜잭션이 허용되지 않음을 표시하는 경우, 상기 하나 이상의 메모리 어드레스에 대한 로크 상태가 변경될 수 있는지 여부를 결정하는 수단과; 그리고

상기 하나 이상의 메모리 어드레스의 로크 상태가 변경될 수 있는 경우, 상기 메모리 트랜잭션을 허용하도록, 상기 하나 이상의 메모리 어드레스의 로크 상태를 변경시키는 수단을 포함하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 15

제 14항에 있어서,

상기 로크 상태를 결정하는 수단은 제1 로크 비트를 판독하는 수단을 포함하고; 그리고

상기 로크 상태를 리턴시키는 수단은 상기 제1 로크 비트의 값을 리턴시키는 수단을 포함하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 16

제 15항에 있어서,

상기 하나 이상의 메모리 어드레스에 대한 로크 상태가 변경될 수 있는지를 결정하는 수단은 제2 로크 비트를 판독하는 수단을 포함하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 17

제 16항에 있어서,

상기 메모리 트랜잭션을 허용하도록 상기 하나 이상의 메모리 어드레스의 로크 상태를 변경시키는 수단은 상기 제1 로크 비트의 값을 변경시키는 수단을 포함하는 것을 특징으로 하는 컴퓨터 시스템.

청구항 18

컴퓨터 시스템에 의해 실행될 때, 상기 컴퓨터 시스템을 동작시키는 방법을 수행하는 명령어들이 수록된 컴퓨터 판독 가능 프로그램 저장 디바이스로서, 여기서 상기 컴퓨터 시스템을 동작시키는 방법은:

하나 이상의 메모리 어드레스에 대한 메모리 트랜잭션을 요청하는 단계와;

상기 하나 이상의 메모리 어드레스에 대한 로크 상태를 결정하는 단계와;

상기 하나 이상의 메모리 어드레스에 대한 상기 로크 상태를 리턴시키는 단계와;

상기 컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부를 결정하는 단계와;

상기 컴퓨터 시스템이 시스템 관리 모드(SMM)에서 동작하는지 여부의 결정에 근거하여, 상기 로크 상태가 상기 하나 이상의 메모리 어드레스에 대한 메모리 트랜잭션이 허용되지 않음을 표시하는 경우, 상기 하나 이상의 메모리 어드레스에 대한 로크 상태가 변경될 수 있는지 여부를 결정하는 단계와; 그리고

상기 하나 이상의 메모리 어드레스의 로크 상태가 변경될 수 있는 경우, 상기 메모리 트랜잭션을 허용하도록, 상기 하나 이상의 메모리 어드레스의 로크 상태를 변경시키는 단계를 포함하는 것을 특징으로 하는 컴퓨터 판독 가능 프로그램 저장 디바이스.

청구항 19

제 18항에 있어서,

상기 로크 상태를 결정하는 단계는 제1 로크 비트를 판독하는 것을 포함하고; 그리고

상기 로크 상태를 리턴시키는 단계는 상기 제1 로크 비트의 값을 리턴시키는 것을 포함하는 것을 특징으로 하는 컴퓨터 판독 가능 프로그램 저장 디바이스.

청구항 20

제 19항에 있어서,

상기 하나 이상의 메모리 어드레스에 대한 로크 상태가 변경될 수 있는지를 결정하는 단계는 제2 로크 비트를 판독하는 것을 포함하는 것을 특징으로 하는 컴퓨터 판독 가능 프로그램 저장 디바이스.

청구항 21

제 20항에 있어서,

상기 메모리 트랜잭션을 허용하도록 상기 하나 이상의 메모리 어드레스의 로크 상태를 변경시키는 단계는 상기 제1 로크 비트의 값을 변경시키는 것을 포함하는 것을 특징으로 하는 컴퓨터 판독 가능 프로그램 저장 디바이스.

명 세 서

<1> 본 출원은 동시 출원중인 발명의 명칭이 "보안 실행 박스 및 방법(Secure Execution Box and Method)", 출원일이 2001년 5월 10일, 발명자가 Dale E. Gulick과 Geoffrey S. Strongin인 미국 특허 출원 제09/852,372호의 일부 계속 출원이다. 본 출원은 발명의 명칭이 "강화된 보안성 및 관리성을 위한 컴퓨터 시스템 구조(Computer System Architecture for Enhanced Security and Manageability)", 출원일이 2001년 5월 10일, 발명자가 Geoffrey S. Strongin와 Dale E. Gulick인 미국 특허 출원 제 09/852,942호의 일부 계속 출원이다.

기 술 분 야

<2> 본 발명은 포괄적으로는 컴퓨터 시스템에 관한 것이고, 보다 구체적으로는 퍼스널 컴퓨터 시스템에서 예를 들어 ROM BIOS와 같은 메모리 장소에의 액세스를 제어하기 위한 외부 로크 메커니즘에 관한 것이다.

배 경 기 술

<3> 도 1a는 예시적인 컴퓨터 시스템(100)을 도시한다. 컴퓨터 시스템(100)은 프로세서(102)와, 노스 브릿지(north bridge)(104)와, AGP 메모리(Advanced Graphics Port memory)(108)와, PCI버스(Peripheral Component Interconnect bus)(110), 사우스 브릿지(south bridge)(112)와, 배터리와, ATA(AT Attachment) 인터페이스(114)와(보다 일반적으로는 IDE(Integrated Drive Electronics) 인터페이스로 알려짐), USB(universal serial bus) 인터페이스(116)와, LPC(Low Pin Count) 버스(118)와, 입력 출력 제어기 칩(SuperI/O™)(120)과, 그리고 BIOS 메모리(122)를 포함한다. 주목할 사항으로, 노스 브릿지(104)와 사우스 브릿지(112)는 오직 하나의 칩이나 또는 복수의 칩들을 포함할 수 있어, 집합적인 용어인 "칩셋(chipset)"에 이른다. 또한 주목할 사항으로, 예를 들어 캐시(caches), 모뎀(modems), 병렬 또는 직렬의 인터페이스(interfaces), SCSI 인터페이스, 네트워크 인

터페이스 카드 등과 같은, 다른 버스들, 디바이스들, 및/또는 서브시스템들이 소망에 따라 컴퓨터 시스템(100)에 포함될 수 있다. ["SuperI/O"는 캘리포니아 산타 클라라의 National Semiconductor Corporation의 상표이다]

<4> 프로세서(102)는 노스 브릿지(104)에 연결된다. 노스 브릿지(104)는 프로세서(102)와, 메모리(106)와, AGP 메모리(108)와, PCI 버스(110) 사이에 인터페이스를 제공한다. 사우스 브릿지(112)는 PCI 버스(110)와, 주변장치, 디바이스 및 서브시스템들과 연결된 IDE 인터페이스(114) 및 USB 인터페이스(116)의 사이에 인터페이스를 제공한다. 배터리(113)가 사우스 브릿지(112)에 연결된 것으로 도시된다. SuperI/O™ 칩(120)이 LPC 버스(118)에 연결된다.

<5> 노스 브릿지(104)는 프로세서(102)와, 메모리(106)와, AGP 메모리(108)와, PCI 버스(118)에 연결된 디바이스들과, 사우스 브릿지(112)에 연결된 디바이스 및 서브시스템들 사이에 통신 액세스(communications access)를 제공한다. 전형적으로는, 제거 가능한 주변 장치들이 컴퓨터 시스템(100)에 연결되기 위해 PCI 버스(110)에 연결된 PCI "슬롯(slots)"(미도시)에 삽입된다.

<6> 사우스 브릿지(112)는 PCI 버스(110)와, 모뎀, 프린터, 키보드, 마우스 등과 같은 일반적으로 LPC 버스(118)(또는 이것의 전모델인, X-버스나 ISA 버스)를 통해 컴퓨터 시스템(100)에 연결되는 다양한 디바이스 및 서브시스템 사이에 인터페이스를 제공한다. 사우스 브릿지(112)는, IDE 인터페이스(114), USB 인터페이스(116) 및 LPC 버스(118)를 통하여, 디바이스들을 나머지 컴퓨터 시스템(100)과 인터페이스로 연결하는데 사용되는 로직을 포함한다.

<7> 도 1b는 종래 기술의 사우스 브릿지(112)의 어떤 양상을 도시한 것으로, 일명 "RTC 배터리 웰(125) 안에 존재하는" 배터리(113)에 의해 예비된 파워가 공급된다. 사우스 브릿지(112)는 사우스 브릿지 RAM(SB RAM)(126)과 클록 회로(clock circuit)(128)를 포함하며, 양자 모두 RTC 배터리 웰(125)에 존재한다.

<8> SB RAM(126)은 CMOS RAM(126A)과 RTC RAM(126B)을 포함한다. RTC RAM(126B)은 클록 데이터(129)와 검사합 데이터(checksum data)(127)를 포함한다. 사우스 브릿지(112)는 또한, RTC 배터리 웰(125)의 밖에, CPU 인터페이스(132), 파워 및 시스템 관리 장치(power and system management units)(133), PCI 버스 인터페이스 로직(134A), USB 인터페이스 로직(134C), IDE 인터페이스 로직(134B) 및 LPC 버스 인터페이스 로직(134D)을 포함한다.

<9> 클록 회로(128)로부터의 시간 및 날짜(date)에 관한 데이터가 RTC RAM(126B)에 클록 데이터(129)로 저장된다. RTC RAM(126B)의 검사합 데이터(127)는, 부팅 처리동안, 아래 기술되는 바와 같이(예를 들어 도 2a의 (148)블록 등과 같이) CMOS RAM(126A) 데이터에 기초하여 계산되어 BIOS에 의해 저장된다. CPU 인터페이스(132)는 인터럽트 신호 제어기(interrupt signal controllers)와 프로세서 신호 제어기(processor signal controllers)를 포함할 수 있다. 파워 및 시스템 관리 장치(133)는 ACPI(Advanced Configuration and Power Interface) 제어기를 포함할 수 있다.

<10> 시스템 관리 모드(SSM: System Management Mode)는 전력을 보존하도록 구현되었던 컴퓨터 시스템의 동작 모드이다. SMM은 제 4세대 x86 프로세서를 위해 생산되었다. 더 새로운 x86 세대 프로세서가 출현함에 따라, SMM은 운영 체제에 대하여 비교적 투명하게 되어 왔다. 즉, 컴퓨터 시스템들은 운영 체제에 거의 또는 전혀 충돌없이 SMM에 들어가고 나온다.

<11> 이제 도면, 특히 도 2a를 참조하면, BIOS(122)에 저장된 코드를 이용하여 컴퓨터 시스템을 시작하는 종래 기술에 의한 방법의 흐름도가 도시된다. 블록(136)에서, 전원 공급기가 초기화되는 동안, 전원 공급기가 파워 굿 신호(power good signal)를 노스 브릿지에 발생시킨다. 블록(138)에서, 전원 공급기로부터 파워 굿 신호를 받은 때, 사우스 브릿지(또는 노스 브릿지)는 프로세서에의 리셋 신호(reset signal) 어써트(assert)를 중지한다.

<12> 초기화 동안, 블록(140)에서, 프로세서는 디폴트 점프 위치(default jump location)를 판독한다. 메모리내의 디폴트 점프 위치는 보통 FFFF0h와 같은 위치에 있다. 블록(142)에서, 프로세서는, ROM BIOS의 적절한 BIOS 코드 위치(예를 들어 FFF0h)로 점프를 실행하고, BIOS 코드를 RAM 메모리에 복제(copy)하고, 그리고 RAM 메모리로부터의 BIOS 코드 명령의 처리를 시작한다. 프로세서에 의해 처리되는 BIOS 코드는 블록(144)에서 POST(power-on self test)를 실행한다.

<13> 그 다음, 블록(146)에서, BIOS 코드는 비디오 제어기, IDE 제어기, SCSI 제어기 등과 같은 것으로 부터의 추가적인 BIOS 코드를 검색하여, 시작 정보 스크린(start-up information screen)을 표시한다. 예로서, 비디오 제

어기 BIOS는 종종 C000h에서 발견되는 반면, IDE 제어기 BIOS 코드는 종종 C800h에서 발견된다. 블록(148)에서, BIOS 코드는 RAM 메모리 카운트업(count-up) 테스트와 같은 추가적인 시스템 테스트와, COM(직렬) 포트 및 LPT(병렬) 포트의 식별을 포함하는 체제 재고 관리(system inventory)를 실행할 수 있다. 블록(150)에서, BIOS 코드는 또한 플러그 앤드플레이(plug-and-play) 디바이스들 및 다른 유사한 디바이스들을 식별하며, 식별된 디바이스의 요약 스크린(summary screen)을 표시한다.

<14> 블록(152)에서, BIOS 코드는 부팅 위치와, 이에 대응하는 부팅 섹터를 식별한다. 부팅 장소는 플로피 드라이브, 하드 드라이브, CDROM, 원격 장소 등일 수 있다. 그 다음 BIOS 코드는, 블록(154)에서, 운영 체제 등과 같은 것으로, 컴퓨터 시스템을 부팅하기 위해 부트 위치(boot location)에서 부트 섹터 코드(boot sector code)를 호출한다.

<15> 주목할 사항으로, 콜드 부트(cold boot) 즉, 하드 (재)부트(hard (re)boot)는, 블록(136) 내지 블록(154)에서 주어진 설명의 모든 것 또는 거의 모든 것이 일어난다. 웜 부트(warm boot) 즉, 소프트 (재)부트(soft (re)boot) 동안, BIOS 코드는 보통 블록(142)에서 블록(148)으로 점프(jump)하여, POST, 메모리 테스트 등을 거치지 않는다.

<16> 도 2b에서, BIOS(122)에 저장된 코드를 이용하여 SMM에서 컴퓨터 시스템을 동작시키는 종래 기술에 의한 방법의 흐름도가 도시된다. 블록(172)에서, 인터럽트 제어기가 SMM에 대한 요청을 수신한다. 블록(174)에서, 인터럽트 제어기는 시스템 관리 인터럽트(SMI#) 신호를 어써트함으로써 SMM에 대한 요청을 프로세서에 시그널링한다.

<17> 블록(176)에서, 프로세서는 SMM에 대한 요청을 인식하여, SMI 활성(SMIACT#) 신호를 어써트한다. 블록(178)에서, 시스템은 SMIACT# 신호를 인식하여, 시스템 RAM으로의 액세스를 디스에이블 시키고, 시스템 관리 RAM(SMRAM) 공간으로의 액세스를 인에이블 시킨다.

<18> 블록(180)에서, 현재 프로세서 상태가 SMRAM에 저장된다. 프로세서는, 블록(182)에서, SMM 디폴트 상태로 리셋되어 SMM에 들어간다. 그 다음, 블록(184)에서, 프로세서는 디폴트 포인터(default pointer)를 판독하고, SMRAM 공간의 적절한 위치에 점프한다. 블록(186)에서, SMI 요청의 소스 및/또는 본성(nature)이 식별된다.

<19> SMI 처리기(SMI handler)가 블록(188)에서 SMI 요청을 서비스한다. SMI 요청을 서비스한 다음, 블록(190)에서, SMI 처리기는 SMM(RSM) 명령으로부터 프로세서로의 리턴(return)을 송신한다. 블록(192)에서, RSM 명령이 동작할 때, 프로세서는 저장된 상태 정보를 재저장하고, 정상 동작을 계속한다.

<20> 하드웨어의 관점에서, x86 연산 환경은 사용자 프라이버시의 보호, 회사 비밀 및 자산의 제공, 또는 컨텐츠 제공자의 소유 권한의 보호를 위해 거의 아무것도 제공하지 않는다. 이러한 모든 목적, 프라이버시, 비밀 및 소유권(집합적으로, PSO)이 인터넷 연결 컴퓨터 시대에 있어서 중요해진다. 오리지널 퍼스널 컴퓨터들은 PSO 요구들을 예상하여 설계되지 못했다.

<21> 소프트웨어의 관점에서, x86 연산 환경은 PSO에 대하여 동등하게 열등하다. 소프트웨어를 통하거나 혹은 간단히 퍼스널 컴퓨터의 커버의 개방을 통한 하드웨어로의 직접 액세스 용이성은 침입자 또는 도둑이 대부분의 보안 소프트웨어 및 디바이스들을 손상시킬 수 있게 한다. 예시한 퍼스널 컴퓨터 사용의 용이성은 PSO에 대한 문제를 더해줄 뿐이다.

발명의 상세한 설명

<22> 본 발명의 일 양상에서, 컴퓨터 시스템이 제공된다. 컴퓨터 시스템은 버스와, 상기 버스에 연결된 메모리와, 상기 버스를 통해 상기 메모리에 액세스하도록 연결된 디바이스를 포함한다. 상기 메모리는 복수의 저장 위치를 포함하고, 이는 복수의 메모리 유닛(memory units)으로 나뉜다. 상기 디바이스는 하나 이상의 상기 복수의 메모리 유닛에 대한 액세스를 제어하도록 구성된 하나 이상의 로크(lock)들을 포함한다. 다양한 실시예에서, 상기 로크들은 복수의 레지스터들을 포함할 수 있다. 하나 이상의 상기 복수의 레지스터들의 하나 이상의 엔트리는 하나 이상의 메모리 유닛에 대한 액세스 제어 설정을 표시할 수 있다.

<23> 본 발명의 또 다른 양상에서, 메모리가 제공된다. 상기 메모리는 BIOS 데이터로 구성된 제1 복수의 저장 위치와, 제2 복수의 저장 위치를 포함한다. 상기 제2 복수의 저장 위치는 오직 SMM에서 판독 가능한 제1 복수의 블록과, SMM 및 SMM을 제외한 적어도 하나의 동작 모드에서 판독 가능한 제2 복수의 블록들을 포함한다.

<24> 본 발명의 또 다른 양상에서, 컴퓨터 시스템을 동작시키기 위한 방법이 제공된다. 이 방법은 하나 이상의 메모리 어드레스에 대한 메모리 트랜잭션을 요청하는 단계와, 상기 하나 이상의 메모리 어드레스에 대한 로크 상태

를 결정하는 단계를 포함한다. 이 방법은 또한 하나 이상의 메모리 어드레스에 대한 로크 상태를 리턴시키는 단계와, 상기 로크 상태가 하나 이상의 메모리 어드레스에 대한 메모리 트랜잭션이 허용되지 않는다고 표시할 경우 상기 하나 이상의 메모리 어드레스에 대한 로크 상태가 변경될 수 있는지 여부를 결정하는 단계를 포함한다. 이 방법은 또한 하나 이상의 메모리 어드레스의 상기 로크 상태가 변경될 수 있는 경우, 메모리 트랜잭션을 허용하도록 상기 하나 이상의 메모리 어드레스의 로크 상태를 변경시키는 단계를 포함한다.

<25> 본 발명의 또 다른 양상에서, 컴퓨터 시스템을 동작시키는 또 다른 방법이 제공된다. 이 방법은 메모리 트랜잭션의 요청을 제1 디바이스로부터 메모리 위치로 송신하는 단계와, 메모리 위치 또는 상기 메모리 위치의 컨텐츠의 복제를 포함하지 않는 제2 디바이스에서 상기 메모리 트랜잭션의 요청을 수신하는 단계를 포함한다. 이 방법은 또한 상기 제2 디바이스로부터, 상기 메모리 트랜잭션의 요청을 송신하는 상기 제1 디바이스로 응답을 리턴시키는 단계를 포함한다.

실 시 예

<39> 본 발명의 예시적인 실시예들이 이하 기술된다. 명료함을 위하여, 실제 구현의 모든 특징들이 본 명세서에 기술되지는 않는다. 물론, 이와 같은 실제 임의의 실시예의 전개에서, 많은 구현-특정 결정들이 예를 들어 시스템 관련 제약 및 사업 관련 제약에 따르는, 실시예별로 변경될 수 있는 개발자의 특정 목표를 달성하도록 행해질 수 있음이 이해될 것이다. 또한, 그러한 개발 노력은 복잡하고 많은 시간이 소요되는 것임에도 불구하고, 본 개시에 대해 이익을 갖는 당업자에게는 일상적인 작업이 될 것이다. 참조 번호와 관련된 문자의 사용은 참조 번호가 결합된 아이템의 대안적인 실시예 또는 예시들을 보이기 위한 것이다.

<40> 도 3은 본 발명의 일 양상에 따른, 보안 실행 박스(secure execution box)(260)를 갖는 컴퓨터 시스템에서 데이터 및 명령 흐름을 보인 흐름도의 실시예의 블록도를 도시한다. 사용자 입력/출력(I/O) 데이터 및/또는 명령들(205)은 하나 이상의 어플리케이션(210)에 제공되고, 하나 이상의 어플리케이션(210)으로부터 수신된다. 어플리케이션들(210)은 컴퓨터 시스템(100)이나 다른 컴퓨터 시스템 등과 같은 컴퓨터 시스템 내에서 데이터 및 명령들을 암호화 서비스 제공자들(cryptography service providers)(215)과 교환한다. 암호화 서비스 제공자들(215)은 API(Application Programming Interface) 호출들(220)을 사용하여 하드웨어(230)에의 액세스를 제공하는 드라이버들(225)과 상호 작용을 할 수 있다.

<41> 본 발명의 일 양상에 따르면, 드라이버들(225) 및 하드웨어(230)는, 보안 실행 모드(SEM: secure execution mode)(260)에서 동작하도록 구성된 보안 실행 박스의 부분이다. 신뢰성 있는 프라이버시, 보안 및 소유 권한(PSO)동작들(또한, 간단히 보안 동작(security operations)이라 부른다)이 컴퓨터 시스템이 SEM(260)에 있는 동안 발생할 수 있다. 사용자 I/O(205) 및/또는 어플리케이션들(210)로부터 발송된 소프트웨어 호출들은 도 5b(또는, 도 5a)와 관련하여 하기에 기술되는 SMM 시작 레지스터(SMM initiation register)(425B)(또는 SMM 시동기(425A))를 통해 SMM(260)의 보안 실행 박스에 위치될 수 있다. 파라미터들이 도 5a 및 도 5b와 관련하여 하기에 기술되는 바와 같이, 액세스 보호 메일박스 RAM(access-protected mailbox RAM)(415)을 통해 SEM(260)의 보안 실행 박스로 그리고 보안 실행 박스로부터 송수신된다. 소프트웨어 호출들은, 하기에 자세히 기술되는 바와 같이, SEM(260)의 보안 실행 박스에 있는 다양한 보안 하드웨어 자원들에의 액세스를 갖는다.

<42> 도 4는 본 발명의 일 양상에 따른, 사우스 브릿지(330)의 보안 하드웨어(security hardware)(370)와 암호 프로세서(crypto-processor)(305)를 포함하는 개선된 버전의 컴퓨터 시스템(100)의 일부에 대한 실시예를 도시한 것이다. 사우스 브릿지(330)는 보안 하드웨어(370)와, 인터럽트 제어기(IC: interrupt controller)(365), USB 인터페이스 로직(134C) 및 LPC 버스 인터페이스 로직(LPC BIL: LPC bus interface logic)(134D)을 포함한다. IC(365)는 프로세서(102)에 연결된다. USB 인터페이스 로직(134C)은 선택적인 USB 허브(hub)(315)에 연결된다. LPC 버스(118)는 LPC BIL(134D)을 통해 사우스 브릿지(330)에 연결된다. 암호 프로세서(305)는 또한 LPC 버스(118)에 연결된다. 암호 프로세서(305)내의 메모리 허가 테이블(memory permission table)(310)은 어드레스 맵핑(address mappings) 및/또는 메모리 범위 허가 정보를 제공한다. 메모리 허가 테이블(310)은 비휘발성 메모리에 포함될 수 있다. BIOS(355)(즉, 어떤 메모리, 바람직하게는 판독 전용 메모리 또는 플래시 메모리)가 암호 프로세서(305)에 연결된다.

<43> 사우스 브릿지(330)의 보안 하드웨어(370)는 프로세서(102)에 IC(365)에 대한 SMI 인터럽트 요청을 제공하도록 동작할 수 있다. 보안 하드웨어(370)는 또한 암호 프로세서(305)와 상호 작용할 수 있다. BIOS(355)로의 액세스는 암호 프로세서(305)를 통해 라우팅된다. 암호 프로세서(305)는 BIOS(355)로의 액세스 요청들을 수신하고 이를 전송하도록 구성된다. 따라서, 암호 프로세서(305)는 BIOS(305)의 어드레스 맵핑을 이해할 수 있다. 본 발명의 일 양상에 따르면, 보안 하드웨어(370)는 컴퓨터 시스템(100)이 도 3에 도시된 보안 실행 박스(260)의 실시

예가 되도록 한다.

<44> 주목할 사항으로, IC(365)는 사우스 브릿지(330) 대신에 프로세서에 포함될 수 있다. IC(365)는 또한 개별 유닛으로서 또는 컴퓨터 시스템(100)의 다른 구성요소와 관계되어 고려될 수 있다. 또한 주목할 사항으로서, LPC 버스(118)의 동작은 1997년 9월 29일의 종래 기술 Low Pin Count Interface Specification Revision 1.0에 해당할 수 있다. 또한 주목할 사항으로, USB 인터페이스 로직(134C)은, 브릿지에서 서로 다른 버스 인터페이스 로직들이 연결되기 위한 종래 공지된 다양한 방법 중 임의의 방법으로, LPC BIL(134D)에 연결될 수 있다.

<45> 도 5a 및 도 5b는 본 발명의 다양한 양상에 따른, 보안 하드웨어(370A)를 포함하는, 사우스 브릿지(330)의 실시예들의 블록도를 도시한다. 도 5a에서, 사우스 브릿지(330A)는 보안 하드웨어(370A)와 IC(365)를 포함한다. 보안 하드웨어(370A)는 SMM 액세스 제어기(402A)와 제어 로직(420A) 등과 같은 서브 디바이스들을 포함한다. 서브 디바이스들은 컴퓨터 시스템(100)의 보안 하드웨어 또는 보안 자산으로서 참조될 수 있다. SMM 액세스 제어기(402A)는 SMM 액세스 필터들(410), 메일박스 RAM(415) 및 SMM 시동기(intiator)(425A)를 포함한다.

<46> 도 5a에서 도시된 바와 같이, 제어 로직(420)은 SMM 액세스 제어기(402A) 및 SMM 시동기(425A)의 동작을 제어하도록 연결된다. 보안 하드웨어(370A)에의 입력 및 출력(I/O)은 SMM 액세스 필터들(410)을 통과하여, 제어 로직(420A)을 통해 라우팅된다.

<47> SMM 액세스 제어기(402A)는 보안 하드웨어(370A) 내에서 서브 디바이스들에 대한 입력 요청들을 접수하도록 구성된 SMM 액세스 필터들(410)을 포함한다. 컴퓨터 시스템(100)이 SMM에 있을 때, SMM 액세스 필터들(410)은 제어 로직(420A) 및/또는 목표 서브 디바이스에 액세스 요청들(예를 들어, 판독 또는 기입)을 건내주도록 구성된다. 컴퓨터 시스템(100)이 SMM에 있지 않을 때, SMM 액세스 필터들(410)은 소정의 값(예를 들어, 모두 '1')으로 모든 액세스 요청들에 응답하도록 구성된다. SMM 액세스 제어기(402A)는 또한 메일박스 RAM(415)을 포함한다. 일 실시예에서, 메일 박스 RAM(415)은 각각 512 바이트를 갖는 2개의 RAM 뱅크를 포함하며, 파라미터들을 보안 실행 박스(260) 내로 그리고 이 박스(260)로부터 송수신한다. 보안 하드웨어(370) 내에 포함된 서브 디바이스들로 및 이들로부터 송수신되는 파라미터들은 메일 박스 RAM(415)에서 교환된다. RAM(415)의 한 뱅크(bank), 즉, 내부박스(inbox)는 대부분의 동작 모드에서 컴퓨터 시스템의 거의 대부분에 대해 기입 전용(write-only)으로 된다. 따라서, 보안 하드웨어(370) 내에 포함된 서브 디바이스에 송신될 파라미터들은 내부박스에 기입될 것이다. 선택된 동작 모드들, 예를 들어 SMM 동안, 판독 액세스 및 기입 액세스 양자 모두가 내부박스에 허용된다. RAM(415)의 또 다른 뱅크 즉, 외부박스(outbox)는 대부분의 동작 모드에서 컴퓨터 시스템의 거의 대부분에 대하여 판독 전용(read-only)으로 된다. 따라서, 보안 하드웨어(370) 내에 포함된 서브 디바이스로부터 수신될 파라미터들은 외부박스로부터 판독될 수 있다. 선택된 동작 모드들, 바람직하게는 SMM과 같은 보안 모드들 동안, 판독 액세스 및 기입 액세스 양자 모두는 외부 박스에 허용된다.

<48> SMM 시동기(425A)는 장점적으로 컴퓨터 시스템(100)이 SMM에 들어갈 것을 요청하는 편리한 방식을 제공할 수 있다. 신호가 요청(REQ) 라인을 통해 SMM 시동기(425A)에 제공될 수 있다. 이 신호는 SMM 메모리 내의 점프 위치의 표시를 제공하여야 한다. SMM 시동기(425A)는 예를 들어 인터럽트 제어기(365)에 SMI#을 제공함으로써, SMM 요청(SMM REQ) 라인을 통해 SMM 에 대한 요청을 만들도록 구성된다. SMM 시동기(425A)는 또한 SMM에 대한 요청이 수신되어 인터럽트 제어기(365)에 전송되었음을 제어 로직(420A)에 통지하도록 구성된다.

<49> 도 5b에서, 사우스 브릿지(330B)는 보안 하드웨어(370B)를 포함한다. IC(365)는 사우스 브릿지(330B)의 외부에 도시된다. 보안 하드웨어(370B)는 SMM 액세스 제어기(402B)와 제어 로직(420B)을 포함한다. SMM 액세스 제어기(402B)는 SMM 액세스 필터들(410) 및 메일박스 RAM(415)를 포함한다. SMM 시작 레지스터(425B)는 사우스 브릿지(330B)의 외부에 도시된다.

<50> 도 5b에 도시된 바와 같이, 제어 로직(420B)은 SMM 액세스 제어기(402B)의 동작을 제어하도록 연결된다. 보안 하드웨어(370B)에 대한 입력 및 출력(I/O) 신호들은 SMM 액세스 필터들(410)을 통과하여, 제어 로직(420B)을 통해 라우팅된다. 제어 로직(420B)은 또한 SMM 시작 레지스터(425B)로부터 SMM에 대한 요청의 표시를 수신하도록 연결된다.

<51> SMM 액세스 제어기(402B)는 SMM 액세스 필터들(410)을 포함하며, 이 필터들은 보안 하드웨어(370B) 내의 서브 디바이스들에 대한 입력 요청들을 접수하도록 구성된다. 컴퓨터 시스템(100)이 SMM에 있을 때, SMM 액세스 필터들(410)은 제어 로직(420B) 및/또는 목표 서브 디바이스에 액세스 요청들(예를 들어, 판독 및 기입)을 전송하도록 구성된다. 컴퓨터 시스템(100)이 SMM에 있지 않을 때, SMM 액세스 필터들(410)은 소정의 값(예를 들어, 모두 '1')으로 모든 액세스 요청들에 응답하도록 구성될 수 있다. SMM 액세스 제어기(402B)는 또한 도 5a에 관하여

상기 기술된, 메일박스 RAM(415)를 포함한다.

<52> SMM 시작 레지스터(425B)는 장점적으로 컴퓨터 시스템(100)이 SMM에 들어가도록 요청하기 위한 편리한 방식을 제공할 수 있다. 신호가 요청(REQ) 라인을 통해 SMM 시작 레지스터(425B)에 제공될 수 있다. 이 신호는 SMM 메모리 내의 점프 위치의 표시를 제공하여야 한다. SMM 시작 레지스터(425B)는 상기 표시를 제어 로직(420B)에 제공하도록 구성된다. 제어 로직(420B)은 예를 들어, 인터럽트 제어기(365)에 SMI#을 제공함으로써, SMM 요청(SMM REQ) 라인을 통해 SMM 에 대한 요청을 만들도록 구성된다.

<53> 주목할 사항으로, 도 5b에 도시된 실시예에서, SMM 시작 레지스터(425B)는 SMM 요청을 처리하기 위한 내부 로직(internal logic)을 포함한다. 도 5b에 도시된 실시예에서, SMM 시작 레지스터(425B)는 SMM 요청을 처리하기 위해 제어 로직(420B)에 의존한다. 또한 주목할 사항으로, SMM 시작 레지스터(425B)는 보안 하드웨어(370A)의 일부인 반면에, SMM 시작 레지스터(425B)는 보안 하드웨어(370B)의 일부가 아니다.

<54> 도 6은 본 발명의 일 양상에 따른, 보안 하드웨어(370C)를 포함하는 사우스 브릿지(330C)의 실시예의 블록도이다. 도시된 바와 같이, 보안 하드웨어(370C)는 SMM 액세스 제어기(402), 제어 로직(420), TCO 카운터(430), 스크래치패드 RAM(scratchpad RAM)(440), 난수 발생기(Random Number Generator : RNG)(455), 보안 시스템(또는 SMM) 관리 레지스터들(470), OAR-(Open At Reset) 로크(locks)(450), OAR 오버라이드 레지스터(OAR override register)(445) 등과 같은 서브 디바이스들을 포함한다. SMM 액세스 제어기(402)는 SMM 액세스 필터들(410) 내에 하나 이상의 액세스 로크(460)를 포함한다. SMM 액세스 제어기(402) 및 제어 로직(420)의 실시예들의 몇몇 양상들은, 상기의 도 5a 및 도 5b에 관하여 기술된다.

<55> 도 6에 도시된 SMM 액세스 제어기(402)의 실시예는 SMM 액세스 필터들(410) 내에 하나 이상의 액세스 로크(460)를 포함한다. 액세스 로크(460)는 보안 하드웨어(370C) 내의 디바이스들중 하나 이상의 디바이스에의 액세스를 방지(즉, 로크) 및 허용(즉, 언로크)하는 수단을 제공한다. 하나 이상의 액세스 로크(460)에 대한 다양한 실시예들은 도 10a 내지 도 10c에 도시되며, 이 도면들을 참조로 하여 기술된다.

<56> 일 실시예에서, 액세스 로크(460)는 리셋(OAR)에서 열려, 보안 하드웨어(370)에 대한 BIOS 소프트웨어 액세스를 허용한다. 그 다음, 이 BIOS 소프트웨어는, 도 2a의 블록(154)에서 보인 부팅 섹터 코드를 호출하기 전에, 액세스 로크(460)를 닫는다. 다양한 실시예들에서, 액세스 로크(460)는 보안 하드웨어(370)에 대한 액세스를 허용하기 위한 소프트웨어 또는 하드웨어에 의해 열릴 수 있다. 예를 들어, 액세스 로크(460)는 IC(365) 또는 프로세서(102) 또는 제어 로직(420)으로부터의 신호에 의해 열릴 수 있다. 액세스 로크(460)는 SMI#에 응답하거나 또는 SMM에 들어가는 프로세서(102)에 응답하여 열릴 수 있다. 액세스 로크(460)에 관한 추가의 정보는 도 9a 내지 도 9c에 관하여 이하 기술된 방법들 900A 내지 900C중 하나 이상으로부터 얻어질 수 있다.

<57> TCO 카운터(또는 타이머)(430)는 카운트다운 타이머와 같은 프로그램 가능한 타이머를 포함할 수 있으며, 이것은 컴퓨터 시스템(100)의 룩업(look-up)을 검출하는데 사용된다. 룩업은, 하나 이상의 서브시스템 또는 구성 요소들이 소정 기간 이상 동안 입력 신호들에 응답하지 않을 경우, 컴퓨터 시스템(100)의 조건으로 정해질 수 있다. 입력 신호들은 컴퓨터 시스템(100) 내부로 부터의 내부 신호와, 예를 들어 사용자 입력 디바이스(예를 들어, 키보드, 마우스, 트랙볼(trackball), 바이오메트릭 디바이스(biometric device) 등)로 부터의 것과 같은, 컴퓨터 시스템(100) 외부로 부터의 신호들을 포함한다. 또한 주목할 사항으로, 사실상 룩업들은 소프트웨어 또는 하드웨어일 수 있다. 본 발명의 다양한 양상에 따르면, TCO 카운터(430)는 내부 SMM으로부터 프로그래밍 되고 판독될 수 있다. TCO 카운터(430)는 바람직하게는 킥 아웃 타이머(kick-out timer)(407)를 위한 디폴트 기간보다 작은 값으로 프로그래밍 될 수 있다. 일 실시예에서, TCO 타이머(430)는 TCO 타이머(430)가 첫번째 종료할 때 SMI#를 생성하고, TCO 타이머(430)는 두번째 즉, 후속의 TCO 타이머(430)가 종료한 때 컴퓨터 시스템을 위한 리셋 신호를 생성한다.

<58> 일 실시예에서, TCO 타이머(430)는, 컴퓨터 시스템(100)이 SMM에 있지 않을 때, 컴퓨터 시스템(100)이 로크업들(lock-ups)로부터 회복할 수 있도록, 컴퓨터 시스템(100) 또는 컴퓨터 시스템(100)에서 동작하는 소프트웨어에 의해 액세스될 수 있다. 또 다른 실시예에서, TCO 타이머(430)는 SMM의 안팎 모두에서 컴퓨터 시스템(100)에 의해 액세스될 수 있다.

<59> 스크래치패드 RAM(440)는 컴퓨터 시스템(100)이 SMM과 같은 어떤 동작 모드에 있는 동안에만 이용가능한 하나 이상의 메모리 블록을 포함한다. 또한, 보안 하드웨어(370)의 다른 서브 디바이스들은 사적인 메모리(private memory)로서 스크래치패드 RAM(440)을 사용하는 것이 고려될 수 있다. 스크래치패드 RAM(440)의 일 실시예는 1kB의 메모리를 포함하지만, 다른 크기의 메모리도 또한 고려될 수 있다. 일 실시예에서, 스크래치패드 RAM은

리셋시 컴퓨터 시스템(100)의 모두 또는 대부분에 개방되며, 또 다른 실시예에서, 스크래치패드 RAM는 컴퓨터 시스템이 부팅하고 있는 동안 액세스 할 수 없다.

<60> 난수 발생기(RNG)(455)는 소정의 범위내의 다수의 비트를 갖는 난수(random number)를 제공하도록 구성된다. 일 실시예에서, 1부터 32까지의 비트 길이를 갖는 새로운 난수가 난수에 대한 요청에 응답하여 제공된다. 주목할 사항으로, 오직 SMM에만과 같이, RNG에의 액세스 제한은 소프트웨어가 표준 API(application programming interface)를 통해 RNG를 액세스하도록 하여, 보안 증진 및 하드웨어 설계 제약의 완화를 가능케 한다는 이점이 있다.

<61> OAR 로크(450)는 복수의 메모리 유닛(예를 들어, 레지스터들)을 포함할 수 있는바, 이 메모리 유닛은 BIOS 정보 또는 다른 데이터를 저장하는데 사용되는 메모리(들)(예를 들어, 하기 도 7a 및 도 7b의 BIOS ROM(355) 및 SMM ROM(550))을 로크하는 관련 프로그래밍 비트(또는, 로크 비트들)를 포함한다. 각 메모리 유닛은 예시된 방법에 의해, 그것과 관계된 세개의 로크 비트를 가질 수 있다. 일 실시예에서, 네개의 8비트 레지스터들이 각 512kB ROM 페이지에 대하여 상기 로크 비트들을 저장하며, 한 개의 레지스터는 매 두 개의 64kB 세그먼트에 대응한다. 네 개의 레지스터의 16개의 블록들로, 최대 8MB의 ROM이 로크될 수 있다. 어드레스는 다음과 같이 될 수 있다:

<62>

64kB 세그먼트	레지스터	어드레스
0, 1	레지스터 0	FFBx,E000h
2, 3	레지스터 1	FFBx,E001h
4, 5	레지스터 2	FFBx,E002h
6, 7	레지스터 3	FFBx,E003h

<63> 각각의 물리적 ROM 칩은 스트래핑 핀으로 알려진 네 개의 식별 핀들(ID[3:0])을 포함할 수 있다. 이 스트래핑 핀들은 각각 64KB인 16개의 공간을 구성하는데 이용된다. 어드레스에 있는 'x'는 스트래핑 핀들의 디코드(decode), 또는 그것의 역(inverse)을 나타낸다.

<64> OAR 로크(450)로부터의 로크 레지스터(lock registers)는 다음을 포함한다:

<65>

레지스터/비트	7	OAR 로크 6:4	3	OAR 로크 2:0
레지스터 0	예약됨(Reserved)	세그먼트 1	예약됨	세그먼트 0
레지스터 1	예약됨	세그먼트 3	예약됨	세그먼트 2
레지스터 2	예약됨	세그먼트 5	예약됨	세그먼트 4
레지스터 3	예약됨	세그먼트 7	예약됨	세그먼트 6

<66> 일 실시예에서, 일 비트는 기입 액세스를 제어하고, 일 비트는 판독 액세스를 제어하고, 그리고 일 비트는 다른 두 비트가 변경하는 것을 방지한다. 일 실시예에서, 일단 로크 비트가 세트되면(로크 다운(lock down; 엄중 잠금)이 되는 상태로도 또한 기술됨), 기입 액세스 비트와 판독 액세스 비트는 메모리가 리셋 신호를 수신할 때까지 재프로그래밍될 수 없다. 각 레지스터의 레이아웃(layout)은 다음을 포함한다:

<67>

비트	7	6	5	4	3	2	1	0
값	예약됨	로크 2	로크 1	로크 0	예약됨	로크 2	로크 1	로크 0

<68> 세개의 로크 비트들의 디코드인 경우 다음을 포함한다:

<69>

디코드	판독 로크 데이터 2	로크 다운 데이터 1	기입 로크 데이터 0	결과적인 블록 상태
0x00	0	0	0	완전 액세스
0x01	0	0	1	기입 로크(디폴트 상태)
0x02	0	1	0	로크 열림(완전 액세스 로크 다운)
0x03	0	1	1	기입 로크 다운
0x04	1	0	0	판독 로크
0x05	1	0	1	판독 및 기입 로크
0x06	1	1	0	판독 로크 다운

0x07	1	1	1	판독 및 기입 로크 다운

<70> 도 6에 도시된 보안 하드웨어(370C)의 실시예는 또한 OAR 오버라이드 레지스터(445)를 포함한다. OAR 오버라이드 레지스터(445)는 보안 하드웨어(370C) 내의 하나 이상의 디바이스로의 액세스를 허용(즉, 언로크) 및 방지(즉, 로크)하기 위한 메커니즘을 제공한다. OAR 오버라이드 레지스터(445)는 또한 액세스 로크(460)를 오버라이드하기 위한 메커니즘을 제공한다. 일 실시예에서, OAR 오버라이드 레지스터(445)는, 액세스 로크들이 무시될 것임을 표시하는 제1 표시기(indicator)를 포함하며, 여기서 보안 하드웨어에의 액세스는 구현에 따라 항상 사용가능하거나 아니면 절대 사용할 수 없게 액세스 로크들(460)에 의해 로크된다. OAR 오버라이드 레지스터(445)는 또한 제1 표시기가 변경되었는지 아닌지의 상태를 나타내는 제 2 표시기를 포함한다. 만약 제 2 표시기가 제1 표시기가 변경하지 않았음을 보이면, OAR 오버라이드 레지스터(445)를 포함하는 디바이스는 바람직하게는 제2 표시기가 변경하도록 리셋을 필요로 한다. 즉, 제 2 표시기는, 액세스 로크(460)의 일 실시예와 유사한, OAR인 것이 바람직하다.

<71> 액세스 로크(460) 및/또는 OAR 오버라이드 표시기들의 사용을 포함하는 방법은 하기의 도 9a 내지 도 9f와 관련하여 기술된다. 하나 이상의 액세스 로크(460)에 대한 다양한 실시예들이 도 10a 내지 도 10c에 도시되고, 이를 참조로 기술되며, OAR 오버라이드 레지스터(445)의 일 실시예는 도 10에 도시되고 이를 참조로 기술된다.

<72> 일 실시예에서, 액세스 로크(460)는 리셋(OAR)에서 개방되며, BIOS 소프트웨어가 보안 하드웨어(370)에 액세스할 수 있도록 한다. 그 다음, BIOS 소프트웨어는, 도 2a의 블록(154)에 도시된 부팅 섹터 코드를 호출하기 전에, 액세스 로크(460)를 닫는다. 다양한 실시예에서, 액세스 로크(460)는 보안 하드웨어(370)에의 액세스를 허용하기 위해 소프트웨어 또는 하드웨어에 의해 개방될 수 있다. 예를 들어, 액세스 로크(460)는 IC(365) 또는 프로세서(102) 또는 제어 로직(420)으로부터의 신호에 의해 개방될 수 있다. 액세스 로크(460)는 SMI# 또는 SMM에 들어가는 프로세서(102)에 응답하여 개방될 수 있다. 액세스 로크(460)에 관한 추가의 정보는 하기의 도 9a 내지 도 9c에 관하여 기술된 방법들 900A 내지 900C중 하나 이상으로부터 얻어질 수 있다.

<73> 주목할 사항으로, 일 실시예에서, 모든 보안 하드웨어(370)와 SMM 시작 레지스터(425B)는 RTC 배터리 웰(125) 안에 있다. 또 다른 실시예에서, 보안 하드웨어(370)의 선택된 서브 디바이스들은 RTC 배터리 웰(125)에서 제외된다. 일 실시예에서, 스크래치패드 RAM(440)의 오직 일부분만 RTC 배터리 웰(125) 내부에 있고, 나머지 부분은 RTC 배터리 웰(125)의 외부에 있다. 예를 들어, 일 실시예에서, 메일박스 RAM(415)는 RTC 배터리 웰(125)의 외부에 있다.

<74> 도 7a 및 도 7b는 본 발명의 다양한 실시예에 따른, 확장 BIOS(extended BIOS) 보안의 실시예들을 도시한다. 도 7a에서 BIOS ROM(355) 및 SMM ROM(550)은 LPC 버스(118)에 연결된다. 도시된 바와 같이, 비밀(610A)을 포함하는 암호 프로세서(305)는 BIOS ROM(355)과 LPC 버스(118) 사이에 연결된다. 도 7b에서, 확장 BIOS ROM(555)는 LPC 버스(118)에 연결된 것으로 도시된다. 확장 BIOS ROM(555)은 BIOS ROM(355)과 SMM ROM(550)을 포함한다.

<75> 컴퓨터 시스템(100)의 BIOS ROM(355) 메모리 공간은 64kB 세그먼트로 나뉘어져, 128kB부터 4MB까지의 그 어느 공간을 포함할 수 있다. 추가적인 하나 이상의 4MB의 SMM ROM(550) 메모리 공간이 예를 들어, ROM 메모리 공간의 두번째 페이지가 개별 칩들 내에 존재하는 페이징 메커니즘을 통하여 어드레스될 수 있고, 추가적인 식별 선택(IDSEL) 핀들의 세트에 의해 선택될 수 있다. BIOS ROM(355) 메모리 공간의 각 세그먼트 및 SMM ROM(550) 메모리 공간은 로크될 수 있으며, 리셋시 개방된다. 일 실시예에서, 액세스 보호 메커니즘(즉, 로크)은, BIOS ROM(355) 또는 SMM ROM(550)에서 구현되지 않고, 예를 들어 도 6과 관련하여 앞서 기술된 바와 같이 보안 하드웨어(370C)의 사우스 브릿지(330C)에서 구현된다.

<76> 일 실시예에서, BIOS ROM(355)은 4MB의 메모리 공간을 구비한다. BIOS ROM(355) 메모리 공간에의 판독 액세스는 어느 때나 제한되지 않을 수 있다. BIOS ROM(355) 메모리 공간 상의 기입 로크는 OAR이며, LPC 버스(145) 상의 32비트 어드레스 공간에서 FFFF,FFFFh부터 FFC0,0000H까지의 메모리 공간을 커버한다.

<77> 일 실시예에서, 암호 프로세서(305)는 특수한 암호화 하드웨어를 포함하는 특수한 프로세서이다. 또 다른 실시예에서, 암호 프로세서(305)는 암호화 펌웨어(firmware) 또는 소프트웨어로 프로그램된 일반 목적의 프로세서를 포함한다. 또 다른 실시예에서, 암호 프로세서(305)는 특수한 암호화 하드웨어로 변형된 범용 프로세서를 포함한다.

<78> 또 다른 실시예들이 또한 고려될 수 있다. 예를 들어, BIOS ROM(355)은 LPC 버스(118)에 연결될 수 있으며, 암호 프로세서(305)는 SMM ROM(550)과 LPC 버스(118)사이에 연결될 수 있다. 또한, 암호 프로세서(305)는 확장

BIOS ROM(555)과 LPC 버스(118) 사이에 연결될 수 있다.

<79> 도 8a 및 도 8b는 본 발명의 다양한 양상에 따라, 보안 SMM 동작을 위한 BIOS ROM(355)과 SMM ROM(550)의 실시예들의 블록도를 각각 도시한다. 도 8a에서 보여지는 바와 같이, BIOS ROM(355)은 데이터 저장공간(data storage)(608B), 비밀(secret)(610C) 및 사적 메모리(private memory)(606)를 포함할 수 있다.

<80> 도 8b에 도시된 바와 같이, SMM ROM(550)은 복수의 SMM ROM 블록들(615 내지 617), 저장된 비밀(610D), 복수의 공용 ROM 블록(public ROM blocks)(625 내지 630), 하나 이상의 예약된 ROM 블록(reserved ROM blocks) 및 하나 이상의 레지스터들(640)로 나뉠 수 있다.

<81> 복수의 SMM ROM 블록들(615~617)은 SMM ROM 0 블록(615), SMM ROM 1 블록(616) 및 SMM ROM 2 블록(617)을 포함한다. 복수의 공용 ROM 블록들(625~630)은 공용 ROM 블록 0(625) 및 공용 ROM 블록 1(630)을 포함할 수 있다. LPC 버스(118) 공간의 액세스 권한, 로크 상태 및 32비트 어드레스 범위가 이하 테이블 형태로 주어진다.

<82>

ROM 블록	판독 액세스	기입 로크	어드레스 범위
SMM ROM 0 615	SMM 전용	한번 기입	FFBx,1FFFh:FFBx,0000h
SMM ROM 1 616	SMM 전용	절대 소거하지 않음	FFBx,3FFFh:FFBx,2000h
SMM ROM 2 617	SMM 전용	없음	FFBx,5FFFh:FFBx,4000h
공용 0 625	비제한	SMM에서 한번 기입	FFBx,9FFFh:FFBx,8000h
공용 1 630	비제한	절대 소거하지 않음, SMM에서 기입	FFBx,BFFFh:FFBx,A000h
예약 635	응답없음	응답없음	FFBx,DFFFh:FFBx,C000h
레지스터들 640	응답없음	응답없음	FFBx,FFFFh:FFBx,E000h

<83> 테이블에 주어진 어드레스 범위의 'x'는 스트래핑 핀 디코드 또는 그것들의 역(inverse)을 표시한다. 일 실시예에서, 테이블의 ROM 블록들(615~617 및 625~630)은 각각 64kB의 크기를 갖는다. 일 실시예에서, 컴퓨터 시스템은 최대 8MB까지의 확장 BIOS ROM(555) 저장공간을 지원하며, 이를 각 512kB의 16개 페이지로 분할한다. 일 실시예에서, FFBx,FFFh로부터 최소 FFBx,0000h까지의 메모리 어드레스 범위는 SMM ROM 블록들(615~617), 복수의 공용 ROM 블록들(625~630) 및 하나 이상의 레지스터(640)를 포함한다. 하나 이상의 지정된 ROM 블록들(635)은 추가의 확장에 사용될 수 있다. 하나 이상의 레지스터들(640)은 필요하다면, 추가의 데이터를 저장할 수 있다.

<84> 도 9a 내지 도 9g는 본 발명의 다양한 양상에 따른, 로크될 수 있는 보안 하드웨어(370)의 액세스를 시도하는 방법들(900A 내지 900G)의 실시예의 흐름도를 도시한다. 도 9a는 부트(즉, 콜드 재부트) 과정의 일부로서 보안 하드웨어(370)를 로크하는 방법(900A)를 도시한다. 도 9b는 재부트(즉, 웜 부트) 과정의 일부로서 보안 하드웨어(370)를 언로크하고, 그 다음 로크하는 방법(900B)을 도시한다. 도 9c는 보안 하드웨어(370)를 로크 또는 언로크 하기 위한 권한을 검사하고, 권한 변경을 디스에이블시키기 위해 비트를 검사하는 방법(900C)을 도시한다. 도 9d는 컴퓨터 시스템(100)이 SMM에 있지 않은 동안, 보안 하드웨어(370)를 사용하도록 시도하는 방법(900D)을 도시한다. 도 9e는 OAR 액세스 로크(460) 상의 로크를 검사 및/또는 결정하고, 로크의 변경을 디스에이블 시키기 위해 비트를 검사하는 방법(900E)을 도시한다. 도 9f는 컴퓨터 시스템(100)이 SMM에 있는 동안 보안 하드웨어(370)를 언로크하고, 그 다음 로크하는 방법(900F)을 도시한다. 도 9g는 컴퓨터 시스템(100)이 SMM에 있는 동안, 보안 하드웨어(370)를 언로크하고, 그 다음 로크하기 위한 권한을 검사하는 방법(900G)을 도시한다.

<85> 이제 도 9a를 참조하면, 방법(900A)은, 블록(920)에서, RAM 메모리의 SMM 공간으로부터의 BIOS 코드 명령들을 실행하는 프로세서를 포함한다. 프로세서에 의해 실행되는 이 BIOS 코드는, 블록(925)에서, POST(Power-On Self Test)를 실행한다. 블록(930)에서, 방법(900A)는 보안 하드웨어(370) 액세스를 포함한다. 컴퓨터 하드웨어(370)에의 액세스는, 보안 하드웨어(370)가 리셋에서 개방되지 않으면, 보안 하드웨어의 언로크를 시작할 수 있다. 보안 하드웨어(370)로의 액세스는 컴퓨터 시스템(100) 내의, 또는 허용된다면 컴퓨터 시스템(100) 외부로 부터의, BIOS 코드 또는 다른 디바이스 또는 서브시스템에 의한다. 방법(900A)은 블록(932)에서 BIOS 관리 모드로

들어가는 것을 선택적으로 포함할 수 있다. BIOS 관리 모드는 예를 들어, 원격 부팅 명령, 부트 시퀀스를 계속하기 위한 원격 또는 보안 허가, 다른 원격 동작 또는 원격 하드웨어 액세스들 또는 셋업들, 또는 하드웨어 구조 및/또는 운영 체제 간의 선택, 또는 다른 소프트웨어 선택들을 허용할 수 있다.

<86> 그 다음, BIOS 코드는 비디오 제어기, IDE 제어기, SCSI 제어기 등과 같은 것으로부터의 추가적인 BIOS 코드를 찾아, 시작 정보 스크린(start-up information screen)을 표시한다. 예를 들어, 비디오 제어기 BIOS는 C000h에서 흔히 발견되는 반면에, IDE 제어기 BIOS 코드는 C800h에서 흔히 발견된다. 블록(940)에서, BIOS 코드는 RAM 메모리 카운트업 테스트 등과 같은 추가의 시스템 테스트를 실행할 수 있고, COM(직렬) 포트 및 LPT(병렬) 포트의 식별을 포함하는 시스템 재고관리(system inventory)가 실행될 수 있다. 블록(945)에서, BIOS 코드는 또한 플러그 앤드 플레이 디바이스들 및 다른 유사한 디바이스들을 식별하며, 식별된 디바이스들의 요약 스크린을 표시한다.

<87> 이 방법은 블록(950)에서 보안 하드웨어에의 액세스 로크 폐쇄를 포함한다. BIOS 코드 또는 컴퓨터 시스템(100)의 또 다른 디바이스 또는 에이전트는 액세스 로크들을 폐쇄할 수 있다. 블록(955)에서, BIOS 코드는 부트 위치와 그리고 대응하는 부트 섹터를 식별한다. 부트 위치는 플로피 드라이브, 하드 드라이브, CDROM, 원격 장소 등이 될 수 있다. 그 다음, BIOS 코드는, 블록(960)에서, 운영 체제와 같은 것으로, 컴퓨터 시스템을 부트하기 위해, 부트 위치에서 부트 섹터를 호출한다.

<88> 이제 도 9b를 참조하면, 방법(900B)은, 블록(915)에서, 보안 하드웨어로의 액세스 로크 개방을 포함한다. 블록(920)에서, 프로세서는 RAM 메모리의 SMM 공간으로부터의 BIOS 코드 명령들을 실행한다. 블록(930)에서, 컴퓨터 시스템은 SMM에서 부팅하는 동안, 보안 하드웨어(370)에 액세스한다. 블록(932)에서, 방법(900B)은 BIOS 관리 모드에 들어가는 것을 선택적으로 포함한다.

<89> 그 다음, 블록(935)에서, BIOS 코드는 비디오 제어기, IDE 제어기, SCSI 제어기 등과 같은 추가적인 BIOS 코드를 찾고, 시작 정보 스크린을 표시한다. 예를 들어, 비디오 제어기 BIOS는 C000h에서 흔히 발견되는 반면에, IDE 제어기 BIOS 코드는 C800h에서 흔히 발견된다. 블록(945)에서, BIOS 코드는 또한 플러그 앤드 플레이 디바이스들 및 다른 유사한 디바이스들을 식별하며, 식별된 디바이스들의 요약 스크린을 표시한다.

<90> 이 방법은 블록(950)에서 보안 하드웨어에의 액세스 로크 폐쇄를 포함한다. BIOS 코드 또는 컴퓨터 시스템(100)의 또 다른 디바이스 또는 에이전트는 액세스 로크들을 폐쇄할 수 있다. 블록(955)에서, BIOS 코드는 부트 위치와 그리고 대응하는 부트 섹터를 식별한다. 부트 위치는 플로피 드라이브, 하드 드라이브, CDROM, 원격 장소 등이 될 수 있다. 그 다음, BIOS 코드는, 블록(960)에서, 운영 체제 등과 같은 것으로, 컴퓨터 시스템을 부트하기 위해, 부트 위치에서 부트 섹터를 호출한다.

<91> 이제 도 9c를 보면, 방법(900C)는, 결정 블록(946)에서, OAR-로크를 설정하는지 여부를 결정하는 단계를 포함한다. 결정 블록(946)에서, OAR-로크는 도 6과 관련하여 상기 기술된 제1 표시기와 대응할 수 있다. 결정 블록(946)에서, OAR-로크는 또한 도 10d와 관련하여 하기에서 기술될 OAR 로크 오버라이드 비트(1050)의 설정에도 또한 대응할 수 있다. 만약 결정이 OAR-로크를 설정하는 것으로 되면, 일 실시예에 따라, 그 다음 블록(947)에서, 보안 하드웨어(370)로의 모든 액세스가 로크 아웃될 것이다. 만약 결정이 OAR-로크를 설정하지 않는 것으로 되면, 그 다음, 방법(900C)은 결정 블록(948)으로 이동한다. 결정 블록(948)에서, 방법(900C)은 OAR-로크의 설정이 비트를 변경시키는지 여부를 결정한다. 결정 블록(948)에서, OAR-로크가 비트를 변경시키는 것은 도 6에 관하여 상기 기술된 제2 표시기에 대응된다. 결정 블록(948)에서, OAR-로크가 비트를 변경시키는 것은 또한 도 10d에 관하여 하기 기술될 변경 OAR 로크 오버라이드 비트(1055)를 설정하는 것에 대응한다. 결정 블록(948)에서, 만약 결정이 OAR-로크가 비트를 변경시키는 것으로 정해지면, 그 다음, 일 실시예에 따라, OAR-로크가 변경될 수 없고, 그런 다음, 블록(949)에서, OAR-로크로의 변경은 스스로 로크 아웃된다.

<92> 이제 도 9d로 돌아가면, 블록(904)에서, 방법(900D)이, SMM이 아닌 모드에서 동작하는 프로세서(102) 등과 같은 프로세서를 포함한다. 블록(906)에서, 프로세서에 의해 처리되는 코드는, 보안 하드웨어(370) 또는 그 액세스가 액세스 로크(460)와 유사한 액세스 로크의 검사를 필요로 할 수도 있는 다른 하드웨어의 임의의 부분에 액세스를 시도한다. 이 방법은, 결정 블록(907)에서, 보안 하드웨어(370)가 유효한 것인지 여부를 알기 위해 검사한다. 결정 블록(907)에서, 만약 보안 하드웨어(370)가 유효하지 않다면, 방법(900D)은 종료하거나 또는 되돌아간다. 결정 블록(907)에서, 만약 보안 하드웨어(370)가 유효하다면, 방법(960D)은 블록(930)에서 보안 하드웨어(370)에 액세스한다. 이 방법은, 만약 필요 하다면 블록(950)에서, 보안 하드웨어에의 액세스 로크를 선택적으로 폐쇄한다.

<93> 이제 도 9e를 보면, 방법(900E)는 도 9d로부터의 결정 블록(907)의 실시예를 포함한다. 결정 블록(990)에서 방법(900E)은 모든 보안 하드웨어로의 액세스가 로크 아웃(즉, 금지) 되었는지 여부를 검사하는 단계를 포함한다. 만약 모든 보안 하드웨어로의 액세스가 로크 아웃되었다면, 결정 블록(990)에서 방법(900E)는 결정 블록(992)으로 이동한다. 만약 모든 보안 하드웨어로의 액세스가 로크 아웃되지 않았다면, 방법(900E)는 결정 블록(991)으로 이동한다. 결정 블록(991)에서, 방법(900E)은 요청된 보안 하드웨어가 로크 아웃되었는지 여부를 (예를 들어, 하나 이상의 액세스 로크들을 개별적으로 사용하여) 검사한다.

<94> 만약 요청된 보안 하드웨어가 로크 아웃되었다면, 방법(900E)은 결정 블록(992)로 이동한다. 만약 요청된 보안 하드웨어가 로크 아웃되지 않았다면, 방법(900E)은 바로 블록(993)으로 이동한다. 결정 블록(992)에서, 방법(900E)는 요청된 보안 하드웨어를 위한 액세스 로크가 바뀔수 있는지(예를 들어, 언로크 될 수 있는지) 여부를 검사한다. 만약 요청된 보안 하드웨어를 위한 액세스 로크가 바뀔 수 없다면, 결정 블록(992)에서 방법(900E)는 보안 하드웨어로의 액세스를 중지한다. 만약 요청된 보안 하드웨어를 위한 액세스 로크가 바뀔 수 있다면, 결정 블록(992)에서, 방법(900E)는 요청된 보안 하드웨어를 위한 액세스 로크를 변경시키기 위하여, 사용자 등으로부터 인증(authorization)을 요청한다. 만약, 요청된 보안 하드웨어를 위한 액세스 로크를 바꾸기 위한 인증이 주어지지 않는다면, 방법(900E)은 보안 하드웨어로의 액세스를 중지한다. 만약, 요청된 보안 하드웨어를 위한 액세스 로크를 바꾸기 위한 인증이 얻어지면, 방법(900E)은 블록(994)로 이동하여, 요청된 보안 하드웨어로의 액세스를 허용하기 위해 로크를 변경시킨다.

<95> 주목할 사항으로, 임의의 인증 방법이 결정 블록(993)에서 사용될 수 있다. 관찰자(observer)의 존재하에 보안 특성을 갖는 종래의 공지된 임의의 인증 방법이 사용될 수 있다.

<96> 이제 도 9f로 가면, 블록(905)에서, 방법(900F)는 RAM 메모리의 SMM 공간에 코드 명령을 로딩하는 프로세서를 포함한다. 예를 들어, SMM에의 코드 명령 로딩은 SMI#에 응답하여 일어날 수 있다. 블록(915)에서, 보안 하드웨어로의 액세스 로크들은 개방된다. 액세스 로크들의 개방은 SMM 코드 명령이나 또는 하드웨어 메커니즘을 통해서, 또는 둘 다를 통해서 일어날 수 있다.

<97> 프로세서는 블록(920)에서, RAM 메모리의 SMM 공간으로부터의 코드 명령을 처리한다. 방법(900F)은 블록(930)에서, 보안 하드웨어(370)의 액세스를 포함한다. 컴퓨터 시스템이 SMM에 있고, 블록(915)에서 액세스 로크가 개방되었기 때문에, 보안 하드웨어는 원하는 바에 따라 컴퓨터 시스템(100)의 서브시스템들 모두 또는 거의 모두에 이용가능하다.

<98> 블록(950)에서, 방법(900F)은 보안 하드웨어(370)에의 액세스 로크의 폐쇄를 포함한다. 블록(965)에서, 프로세서는 이전 상태를 재로딩하고 동작을 계속한다. 주목할 사항으로, 블록(920)에서, SMM 코드 명령은 블록(930)에서 기술되는 동작이 일어나는 동안 계속될 수 있다. 바람직하게는, 블록(950)에서 기술되는 동작은 블록(920)에서 SMM 코드 명령이 처리된 다음 중지된다.

<99> 이제 도 9g를 보면, 블록(905)에서, 방법(900G)이 RAM 메모리의 SMM 공간에 코드 명령을 로딩하는 프로세서를 포함한다. 예를 들어, SMM에의 코드 명령 로딩은 SMI#에 응답하여 일어날 수 있다. 그 다음, 방법(900G)은, 결정 블록(907)에서, 보안 하드웨어가 유효한지 여부를 검사한다. 보안 하드웨어가 유효하지 않다면, 결정 블록(907)에서, 방법(900G)은 보안 하드웨어로의 액세스를 중지한다. 만약 보안 하드웨어가 유효하다면, 방법(900G)은 블록(920)으로 계속된다.

<100> 블록(920)에서, 프로세서는 RAM 메모리의 SMM 공간으로부터의 코드 명령을 실행한다. 블록(930)에서, 방법(900G)은 보안 하드웨어(370)의 액세스를 포함한다. 컴퓨터 시스템이 SMM에 있고 액세스 로크가 개방되기 때문에, 결정 블록(907)에서 결정된 바와 같이, 보안 하드웨어는 바라는 바에 따라 컴퓨터 시스템(100)의 거의 모든 또는 모든 서브시스템에 이용가능하게 된다.

<101> 블록(950)에서, 방법(900G)은 보안 하드웨어(370)에의 액세스 로크 폐쇄를 포함한다. 블록(965)에서, 프로세서는 이전 상태를 재로딩하고, 동작을 계속한다. 주목할 사항으로, 블록(920)에서, SMM 코드 명령들은 블록(930)에서 기술되는 동작이 발생하는 동안 계속될 것이다. 바람직하게는, 블록(950)에서 기술되는 동작들은 블록(920)에서 SMM 코드 명령들이 처리된 다음 중지된다.

<102> 주목할 사항으로, 액세스 로크 이외의, 보안 하드웨어(370)를 로크 및 언로크하는 다른 과정들이 이용될 수 있다. 방법들(900A~900G)은 다른 과정들로 확장될 수 있다.

<103> 설명을 위하여, 컴퓨터 시스템은 정상 모드와 SMM인 두개의 동작 모드를 갖는 것으로 고려되었다. SMM에 있지는

않으나, 정의에 의해서 SMM에 있는 것으로 신뢰되고, 따라서 본 명세서에서 SMM과 균등한 것으로 고려되는 부트 단계들이 있다. 부트 코드는 SMM이 동작하는 방식을 구성 및 배열한다. SMM은 부트 코드의 신뢰성으로부터 그의 신뢰성을 끌어낸다. 표준 부트 시퀀스가 수정될 수 있다는 것이 고려된다. 수정들은 사용자가 파라미터들을 입력할 수 있는 기회를 가질 수 있는 셋업 환경으로의 전이를 포함한다. 입력 파라미터들은 예를 들어 BIOS 코드를 변경할 수 있다. 대부분의 셋업 환경들은 운영 체제를 로딩하여 정상 모드로 동작하기 전에 리셋으로 리턴한다. 이것은 운영 체제를 로딩하는 대안적이고도, 정상 모드의 부분이 아닌, 유지 보수 모드(maintenance mode)의 형태이다. 고려되는 바와 같이, 액세스 로크는 이 모드에 설정되지 않을 수 있다. 이것은 부트 과정의 일부가 될 수 있고, SMM과 동일하도록 신뢰되고 있지만, 만일 원격 액세스들이 셋업 환경 내부에 있을 수 있는 경우에는 보안 측정들(security measures)이 사용될 수 있다.

<104> 도 10a, 도 10b 및 도 10c는 도 6에 도시된 액세스 로크(460)의 실시예들(460A, 460B, 460C)의 블록도를 도시한 것이다. 도 10d에서, 도 6으로부터의 OAR 오버라이드 레지스터(445)의 실시예의 블록도가 도시된다. 도 10a에 도시된 실시예(460A)에서, 하나 이상의 액세스 로크들(460)이 시퀘스터(sequester) 비트 레지스터(1005)를 포함한다. 시퀘스터 비트 레지스터(1005)에 저장된 비트는 플래그로서 설정되거나 소거될 수 있다. 도 10b에 도시된 실시예(460B)에서, 하나 이상의 액세스 로크들(460)은 보안 하드웨어(370) 내의 모든 디바이스들을 로크 또는 언로크하기 위해 두개 이상의 시퀘스터 비트를 저장하도록 구성된 두 개 이상의 시퀘스터 레지스터를 포함한다. 예를 들어, 기입 특권(write privilege)이 로크되는 반면에, 판독 특권(read privilege)이 언로크 될 수 있다. 도 10c의 실시예에서, 하나 이상의 액세스 로크들(460)은 보안 하드웨어(370C) 내의 각 디바이스를 위한 하나 이상의 시퀘스터 레지스터들(1015A~1015N)을 포함한다.

<105> 도 10d에서, OAR 오버라이드(445)는 적어도 하나의 OAR-로크 오버라이드 비트를 저장하는 OAR-로크 오버라이트 레지스터(1050)과, 적어도 하나의 변경 OAR-로크 오버라이드 비트(change OAR-lock override bit)를 저장하는 변경 OAR-로크 오버라이드 레지스터(change OAR-lock override register)(1055)를 포함한다. 본 발명의 일 실시예에 따르면, OAR-로크 오버라이드 비트가 설정되지 않으면, 보안 하드웨어(370)로의 액세스는 액세스 로크들(460)의 설정에 의해 결정된다. 만약 OAR-로크 오버라이드 비트가 설정되면, 액세스 로크들(460)은 항상 유효하거나 또는 절대 유효하지 않은 보안 하드웨어(370)를 위하여, 구현에 근거하여, 무시된다. 바람직하게는, OAR-로크 오버라이드 비트가 설정된 때, 보안 하드웨어는 절대 유효하지 않다. OAR-로크 오버라이드 비트의 설정은 변경 OAR-로크 오버라이드 비트가 설정되지 않는다면 SMM으로(또는 인증으로) 변경할 수 있다. 바람직하게는, 변경 OAR-로크 오버라이드 비트는 액세스 로크들(460)의 일 실시예와 유사한 OAR이며, 다양한 실시예에서, 블록(950)에서와 같이 부트 시간에 액세스 로크들(460)로 설정될 수 있다.

<106> 도 11a, 도 11b, 도 12 및 도 13은, 본 발명의 다양한 양상에 따라, 저장공간에의 보안 액세스를 위한 방법들(1100A, 1100B, 1110A 및 1120)의 실시예들의 흐름도를 도시한다. 도 11a는, 본 발명의 일 양상에 따라, 보안 디바이스가 저장 디바이스에 보안 액세스를 유지하는 방법(1100A)의 흐름도를 도시한다. 도 11b는 본 발명의 일 양상에 따라, 암호 프로세서가 메모리로의 보안 액세스를 유지하는 방법(1100B)의 흐름도를 도시한다. 도 12는, 본 발명의 일 양상에 따라, 보안 디바이스가 도전 응답 인증 규약(challenge-response authentication protocol)을 이용하여 저장 디바이스에 대한 보안 액세스 제어를 제공하는 방법(1110A)의 흐름도를 도시한다. 도 13은 비밀이 보안 저장 디바이스에의 데이터 액세스를 언로크하는데 이용되는 방법(1120)의 흐름도를 도시한다.

<107> 도 11a를 보면, 방법(1100A)는 보안 디바이스에 연결된 저장 디바이스에 관계된 저장 위치에 대한 트랜잭션 요청을 수신하는 보안 디바이스를 포함한다(블록(1105A)). 보안 디바이스는 저장 디바이스에 액세스 제어를 제공한다(블록(1110A)). 블록(1110A)에서 보인 액세스 제어의 일 실시예가 도 12에서 보이는 방법(1100B)에 의해 도시된다.

<108> 방법(1100A)에 따르면, 보안 디바이스는 저장 디바이스의 어드레스 맵핑에 따라 트랜잭션 요청에 저장 위치를 맵핑한다(블록(1115A)). 보안 디바이스는 저장 디바이스에 트랜잭션 요청을 제공한다(블록(1120A)). 정상 환경하에서, 저장 디바이스는 요청된 트랜잭션을 실행할 것이다(블록(1125A)).

<109> 다양한 실시예에서, 방법(1110A)과 관계된 보안 디바이스는 저장 디바이스에 보안을 제공하도록 구성된 암호 프로세서 또는 로직 블록을 포함한다. 메모리는 RAM, ROM 또는 플래쉬 메모리를 포함할 수 있다. 하드 드라이브 또는 광학 드라이브(optical drive)는 고정될 수도 있고 제거 가능할 수도 있다. 트랜잭션 요청은 예를 들어, 판독 요청, 기입 요청, 또는 이들 모두를 포함할 수 있다. 주목할 사항으로, 다양한 실시예에서, 메모리(또는, 저장 디바이스)는 그것 자체의 보안 하드웨어를 추가로 포함할 수 있다.

<110> 도 11b를 보면, 방법(1100B)는 암호 프로세서에 연결된 메모리와 관계되는 메모리 위치에 대한 트랜잭션 요청을 수신하는 암호 프로세서를 포함한다(블록(1105B)). 암호 프로세서는 메모리에 액세스 제어를 제공한다(블록(1110B)). 블록(1110B)에서 보인 액세스 제어의 일 실시예는 도 12에 도시된다.

<111> 방법(1100B)에 따르면, 암호 프로세서(crypto-processor)는 메모리의 어드레스 맵핑에 따라 트랜잭션 요청에 메모리 위치를 맵핑한다(블록(1115B)). 암호 프로세서는 메모리에 트랜잭션 요청을 제공한다(블록(1120B)). 정상 환경하에서, 메모리는 요청된 트랜잭션을 실행할 것이다(블록(1125B)).

<112> 도 12를 보면, 방법(1110A)는 로크가 저장 위치(storage location)를 위한 적소에 있는지 여부를 결정하는 보안 디바이스를 포함한다(블록(1205)). 저장 위치를 위해 트랜잭션 요청이 수신되었을 수 있다. 만약 로크가 적소에 없다면(블록(1210)), 방법(1110A)은 인증 부분으로 되돌아 간다. 만약 로크가 적소에 존재한다면(블록(1210)), 보안 디바이스는 저장 위치에 도전을 제공한다(블록(1215)). 도전은 저장 위치 또는 저장 위치를 포함하는 저장 디바이스와 관계될 수 있다. 도전은 트랜잭션 요청에 응답할 수 있다. 그 다음, 보안 디바이스는 도전에 대한 응답을 수신한다(블록(1220)). 보안 디바이스는 이 응답을 기대 응답(expected response)과 비교함으로써 응답을 평가한다(블록(1225)). 만약 평가가 일치하지 않는다면(블록(1230)), 이 방법은 종료한다. 만약 평가가 일치한다면(블록(1230)), 이 방법은 보안 디바이스로 진행되어, 이 디바이스가 저장 디바이스에 트랜잭션 요청한다(블록(1235)).

<113> 다양한 실시예에서, 방법(1110A)과 관계된 보안 디바이스는 저장 디바이스에 보안을 제공하도록 구성된 암호 프로세서 또는 로직 블록을 포함한다. 저장 디바이스는 메모리와 같은 전기적 저장 매체나, 또는 하드 드라이브 또는 광학 드라이브와 같은 자기적 또는 광학적 저장 매체를 포함할 수 있다. 메모리는 RAM, ROM 또는 플래쉬 메모리를 포함할 수 있다. 하드 드라이브 또는 광학 드라이브는 고정될 수도 있고, 제거가능할 수도 있다. 트랜잭션 요청은 예를 들어 판독 요청 또는 기입 요청을 포함할 수 있고, 양자 모두를 포함할 수도 있다.

<114> 도 13을 보면, 방법(1120)은 저장 디바이스에 비밀을 저장하는 것을 포함한다(블록(1305)). 저장 디바이스는 오직 물리적 디바이스의 일부분만을 포함할 수 있다. 저장 디바이스 자체는 종래 공지된 어떠한 저장 디바이스로서도 구현될 수 있다. 방법(1120)은 또한 저장 디바이스에 데이터를 저장하는 단계(블록(1310))와 저장 디바이스에 코드를 저장하는 단계(블록(1315))를 포함할 수 있다. 방법(1120)은 또한 저장 디바이스에 저장된 데이터 또는 저장 디바이스 자체를 보안하기 위하여 로크(예를 들어, 로크 비트(들))를 제공하는 단계를 포함할 수 있다(블록(1315)). 주목할 사항으로, 방법(1120)의 상기 단계들(블록(1305) 내지 블록(1320))은, 예를 들어 저장 디바이스가 제작되고, 인스톨되고 또는 초기화 되는 때, 시간적으로 비교적 가까이 실행될 수 있다.

<115> 방법(1120)은 또한, 예를 들어 저장 디바이스를 포함하거나 또는 저장 디바이스와 통신하도록 연결된 컴퓨터 시스템이 부트되는 때, 저장 디바이스로부터 비밀을 판독하는 단계를 포함한다(블록(1325)). 보안이 유지되는 비밀을 위하여, 비밀의 판독은 바람직하게는 저장 디바이스가 보안 또는 신뢰 구성 상태에 있을 때 일어난다. 방법(1120)은 또한 저장 디바이스로부터 코드를 판독할 수 있다(블록(1330)). 방법(1120)은 보안 위치에 비밀을 저장하고(블록(1325)), 또한 보안 위치에 코드를 저장할 수 있다(블록(1330)). 보안 위치는 전술한 바와 같이 SMM 메모리 공간에 존재하거나, 또는 보안 메모리, 레지스터, 프로세서(102)나 사우스 브릿지(330) 등과 같은 컴퓨터 시스템의 다른 저장 위치에 존재할 수 있다.

<116> 다양한 실시예에서, 방법(1120)에 관계된 저장 디바이스는 메모리와 같은 전기적 저장 매체나, 하드 드라이브 또는 광학 드라이브와 같은 자기적 또는 광학적 저장 매체를 포함할 수 있다. 메모리는 RAM, ROM 또는 플래쉬 메모리를 포함할 수 있다. 하드 드라이브 또는 광학 드라이브는 고정될 수도 있고, 제거가능할 수도 있다. 방법(1120)의 판독이, 예를 들어 판독 요청, 기입 요청, 또는 이들 요청의 결합과 같은 임의의 트랜잭션 요청을 기술할 수 있다.

<117> 본 개시의 목적을 위해, ROM에 대한 언급은 플래시 메모리 및 실질적으로 비휘발성인 다른 메모리 타입들에 대해 적용한 것으로서 해석되어야 한다. 주목할 사항으로서, 본원에 개시된 본 발명의 방법들이 흐름도로서 설명되었지만, 이러한 흐름도의 다양한 요소들은 다양한 실시예들에서 생략되거나 또는 다른 순서로 실행될 수 있다. 또한, 주목할 사항으로서, 본원에서 개시된 본 발명의 방법들은 변형되어 구현될 수 있다.

<118> 상기 설명된 본 발명의 일부 양상들은 하드웨어 또는 소프트웨어로 구현될 수 있다. 따라서, 본원의 상세한 설명의 일부는 하드웨어 구현 프로세스에 관련하여 제시되었으며, 상세한 설명의 일부는 컴퓨팅 시스템 또는 컴퓨팅 디바이스의 메모리 내에서의 데이터 비트들에 대한 상징적인 동작 표현들을 포함하는 소프트웨어 구현 프로세스에 관련하여 제시되었다. 이러한 설명들 및 표현들은 당업자들이 하드웨어 및 소프트웨어를 이용하여 그들

의 작업 내용을 다른 당업자들에게 전달하기 위해 이용되는 수단이다. 프로세스 및 동작은 모두 물리량들의 물리적인 처리를 필요로 한다. 소프트웨어에서, 반드시 그런 것은 아니지만, 이러한 물리량들은 저장, 전달, 결합, 비교될 수 있는(그렇지 않으면 처리될 수 있는) 전기, 자기 또는 광학 신호들의 형태를 가질 수 있다. 주로 공통 이용을 위해, 이러한 신호들을 비트들, 값들, 요소들, 심볼들, 기호들, 용어들, 숫자들 등으로서 설명하는 것이 종종 편리함이 입증되었다.

<119> 그러나, 이들 및 유사한 용어들 모두는 적절한 물리량들과 관련되며 이러한 물리량들에 적용된 단지 편리한 라벨들일 뿐 이라는 것을 염두해야한다. 특별하게 지정하지 않는한, 그렇지 않으면 명백한 것으로서, 본 개시 전체를 통해, 이러한 설명들은 어떠한 전자 디바이스의 저장 장치 내의 물리(전기, 자기 또는 광학)량들로서 표현된 데이터를 처리하고, 저장 장치, 또는 전송 또는 디스플레이 디바이스들 내의 물리량들로서 유사하게 표현되는 다른 데이터로 변환하는 전자 디바이스의 실행 및 프로세스들을 설명한다. 이러한 설명을 나타내는 용어들의 예로는, 한정하는 것은 아니지만, "프로세싱", "컴퓨팅", "계산(calculating)", "결정", "디스플레잉" 등이 있다.

<120> 또한, 주목할 사항으로, 본 발명의 소프트웨어로 구현되는 양상들은 전형적으로는 몇몇 형태의 프로그램 저장 매체로 부호화 되거나, 몇몇 형태의 전송 매체를 통해 구현된다. 프로그램 저장 매체는 자기적(예를 들어, 플로피 디스크 또는 하드 드라이브), 또는 광학적(예를 들어, 컴팩트 디스크 판독 전용 메모리, 또는 "CD ROM")일 수 있고, 판독 전용 또는 랜덤 액세스일 수 있다. 유사하게, 전송 매체는 연선(twisted wire pairs), 동축 케이블(coaxial cable), 광섬유(optical fiber), 또는 종래 공지된 다른 적절한 전송 매체일 수 있다. 본 발명은 주어진 구현의 이들 양상에 의해 한정되지 않는다.

<121> 상기 기술된 구체적인 실시예들은 오직 설명을 위한 것이므로, 본 발명은 본 발명과 다르나, 본 명세서에서 개시한 이익을 갖는 당업자에게 명백한, 균등한 방식으로 변형되거나 실시될 수 있다. 또한, 이하 청구항에서 기술되는 것 이외의, 본 명세서에서 도시된 구성 또는 설계의 상세한 설명은 한정을 목적으로 하는 것이 아니다. 상기 기술된 구체적인 실시예들은 대체되거나 변형될 수 있으며, 모든 그러한 변형은 본 발명의 범주내에 있는 것으로 고려된다. 따라서, 본 명세서에서 구해지는 보호는 이하의 청구항에서 기술된다.

도면의 간단한 설명

<26> 본 발명은 첨부된 도면들과 관련한 후술의 설명을 참조하여 이해될 수 있을 것이며, 도면에서 유사한 요소들은 같은 참조 번호로 표시한다.

<27> 도 1a는 종래 기술의 컴퓨터 시스템의 블록도를 도시하며, 도 1b는 종래 기술의 사우스 브릿지의 블록도를 도시한다;

<28> 도 2a 및 도 2b는 ROM에 저장된 코드를 이용하여 컴퓨터 시스템을 동작시키기 위한 종래 기술에 의한 방법의 흐름도를 도시한다;

<29> 도 3은 본 발명의 일 양상에 따른, 보안 실행 박스를 가진 컴퓨터 시스템에서 데이터 및 명령 흐름의 실시예의 흐름도를 도시한다;

<30> 도 4는 본 발명의 일 양상에 따른, 사우스 브릿지의 보안 하드웨어 및 암호-프로세서(crypto-processor)를 포함한 컴퓨터 시스템의 실시예의 블록도를 도시한다;

<31> 도 5a 및 도 5b는 본 발명의 다양한 양상에 따른, SMM의 제어를 위한 보안 하드웨어를 포함한 사우스 브릿지의 실시예의 블록도를 도시한다;

<32> 도 6은 본 발명의 일 양상에 따른, SMM 동작의 보안을 위한 보안 하드웨어를 포함한 사우스 브릿지의 실시예의 블록도를 도시한다;

<33> 도 7a 및 도 7b는 본 발명의 다양한 양상에 따른, 보안 메모리의 실시예들을 도시한다;

<34> 도 8a 및 도 8b는 본 발명의 다양한 양상에 따른, SMM 동작의 보안을 위한 BIOS ROM과 SMM ROM의 실시예의 블록도를 각각 도시한다;

<35> 도 9a, 도 9b, 도 9c, 도 9d, 도 9e, 도 9f 및 도 9g는 본 발명의 다양한 양상에 따른, 로크 상태로 될 수 있는 보안 하드웨어에 액세스하기 위한 방법의 실시예의 흐름도를 도시한다;

<36> 도 10a, 도 10b 및 도 10c는 도 6에 도시된 액세스 로크(460)의 실시예의 블록도를 도시하며, 도 10d는 오버라

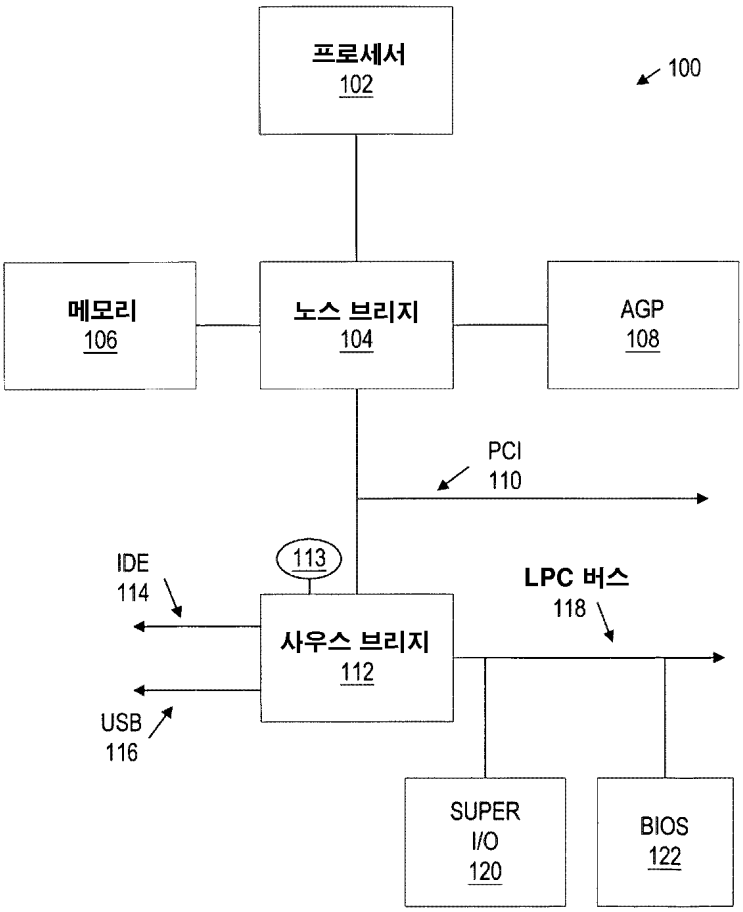
이드 레지스터(override register)의 실시예의 블록도를 도시하며, 이들 모두는 본 발명의 다양한 양상에 따른 것이다;

<37> 도 11a, 도 11b, 도 12 및 도 13은 본 발명의 다양한 양상에 따른, 기억장소(storage)에의 보안 액세스를 위한 방법의 실시예의 흐름도를 도시한다.

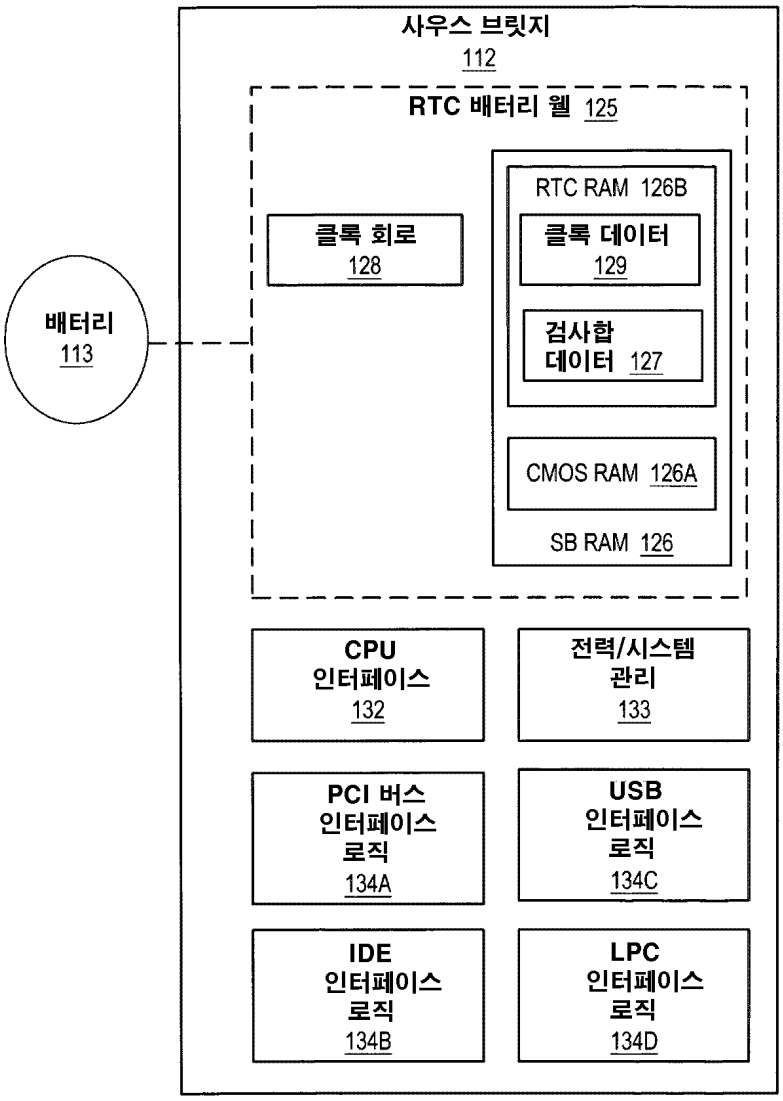
<38> 본 발명은 다양한 변형과 대안적인 형태가 가능하나, 그들의 구체적인 실시예는 도면에서 예시로서 도시되고, 본 명세서에서 구체적으로 기술된다. 그러나, 구체적인 실시예에 대한 본 명세서의 설명은 개시된 특정한 형태로만 본 발명을 한정시키려는 의도가 아니라, 첨부된 청구항에 의해 정의된 바와 같이 본 발명의 범주에 드는 모든 변형물, 균등물 및 대안물을 커버하도록 의도된 것임이 이해되어야 한다.

도면

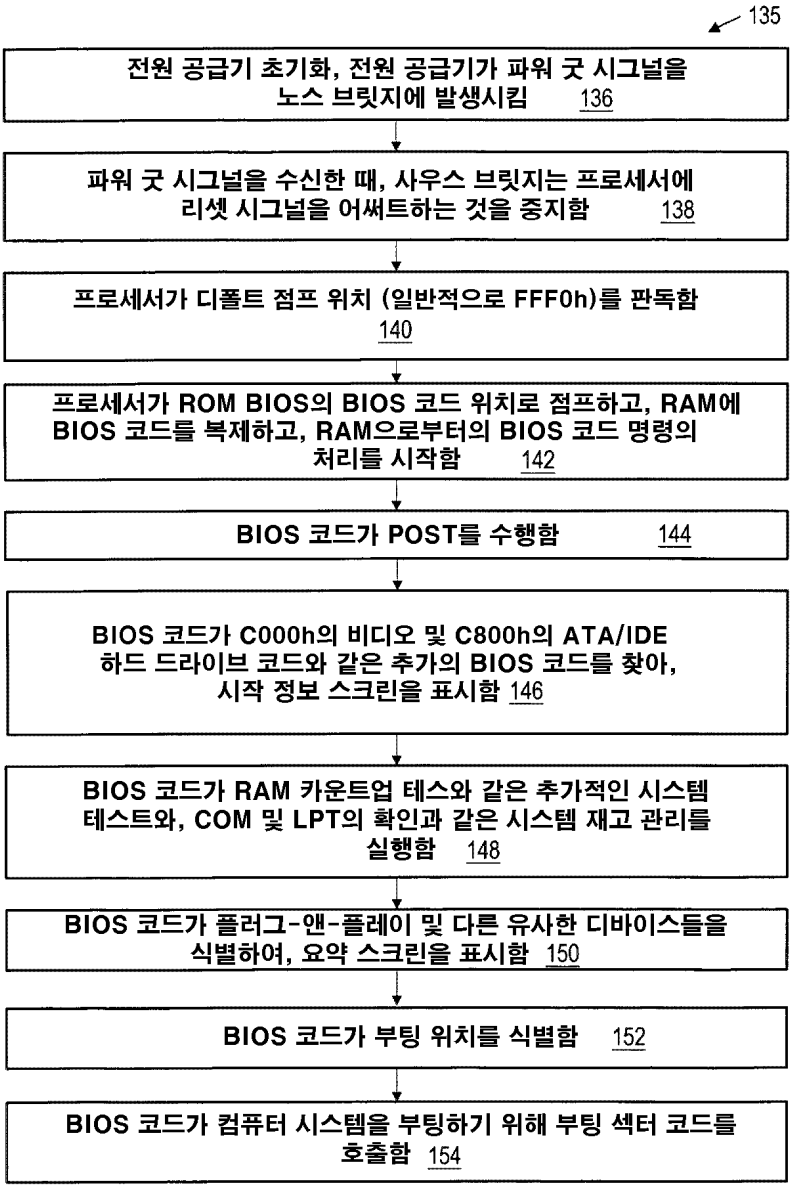
도면1A



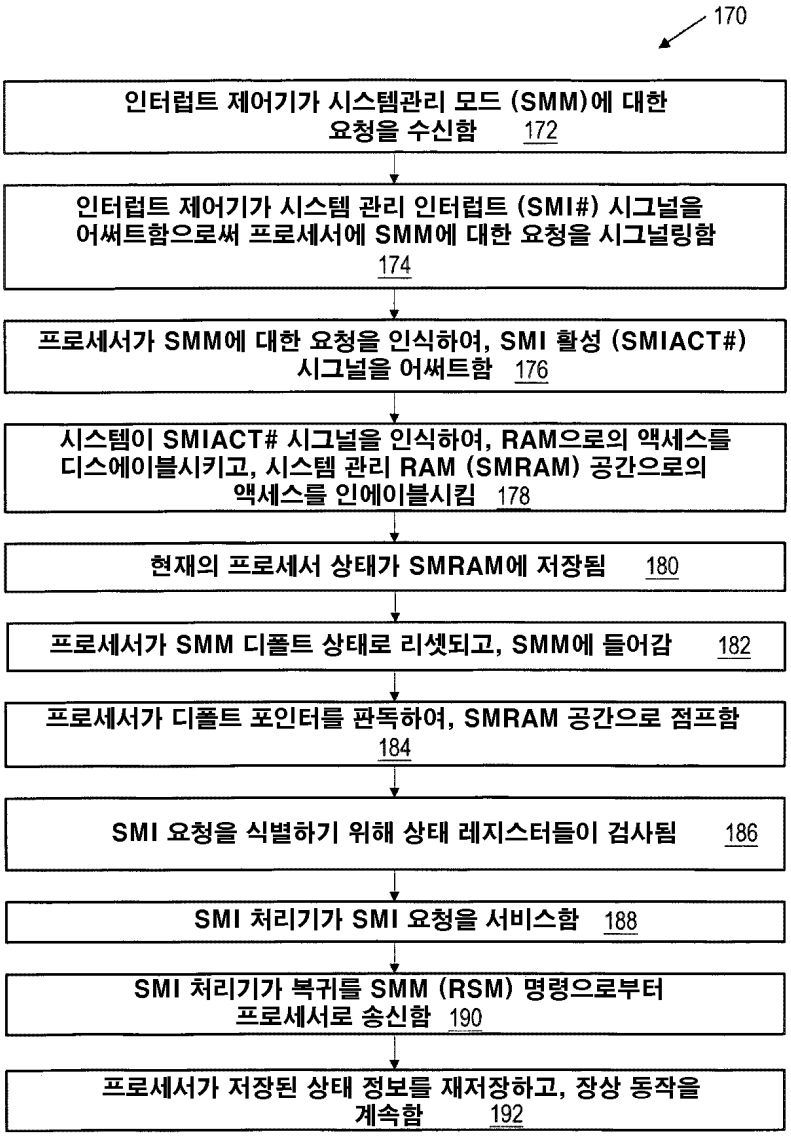
도면1B


사우스 브릿지
112
RTC 배터리 웰 125
RTC RAM 126B
클록 데이터
129
클록 회로
128
검사합
데이터 127
배터리
113
CMOS RAM 126A
SB RAM 126
CPU
인터페이스
132
전력/시스템
관리
133
PCI 버스
인터페이스
로직
134A
USB
인터페이스
로직
134C
IDE
인터페이스
로직
134B
LPC
인터페이스
로직
134D

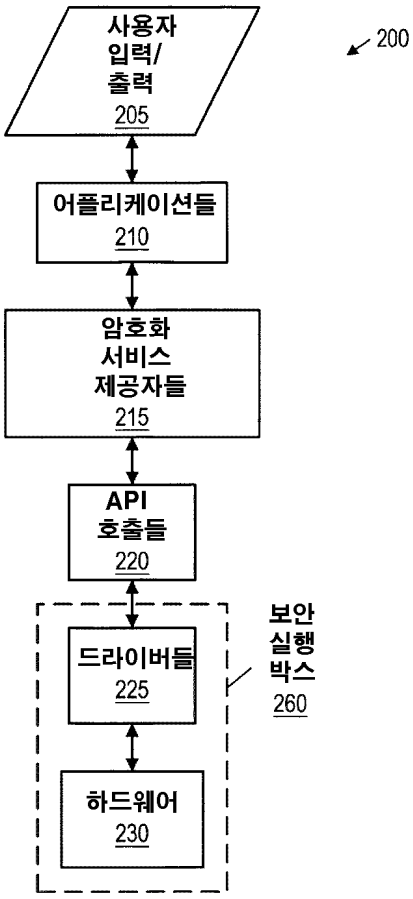
도면2A


135
전원 공급기 초기화, 전원 공급기가 파워 굿 시그널을 노스 브릿지에 발생시킴 136
파워 굿 시그널을 수신한 때, 사우스 브릿지는 프로세서에 리셋 시그널을 어써트하는 것을 중지함 138
프로세서가 디폴트 점프 위치 (일반적으로 FFF0h)를 판독함 140
프로세서가 ROM BIOS의 BIOS 코드 위치로 점프하고, RAM에 BIOS 코드를 복제하고, RAM으로부터의 BIOS 코드 명령의 처리를 시작함 142
BIOS 코드가 POST를 수행함 144
BIOS 코드가 C000h의 비디오 및 C800h의 ATA/IDE 하드 드라이브 코드와 같은 추가의 BIOS 코드를 찾아, 시작 정보 스크린을 표시함 146
BIOS 코드가 RAM 카운트업 테스와 같은 추가적인 시스템 테스트와, COM 및 LPT의 확인과 같은 시스템 재고 관리를 실행함 148
BIOS 코드가 플러그-앤-플레이 및 다른 유사한 디바이스들을 식별하여, 요약 스크린을 표시함 150
BIOS 코드가 부팅 위치를 식별함 152
BIOS 코드가 컴퓨터 시스템을 부팅하기 위해 부팅 섹터 코드를 호출함 154

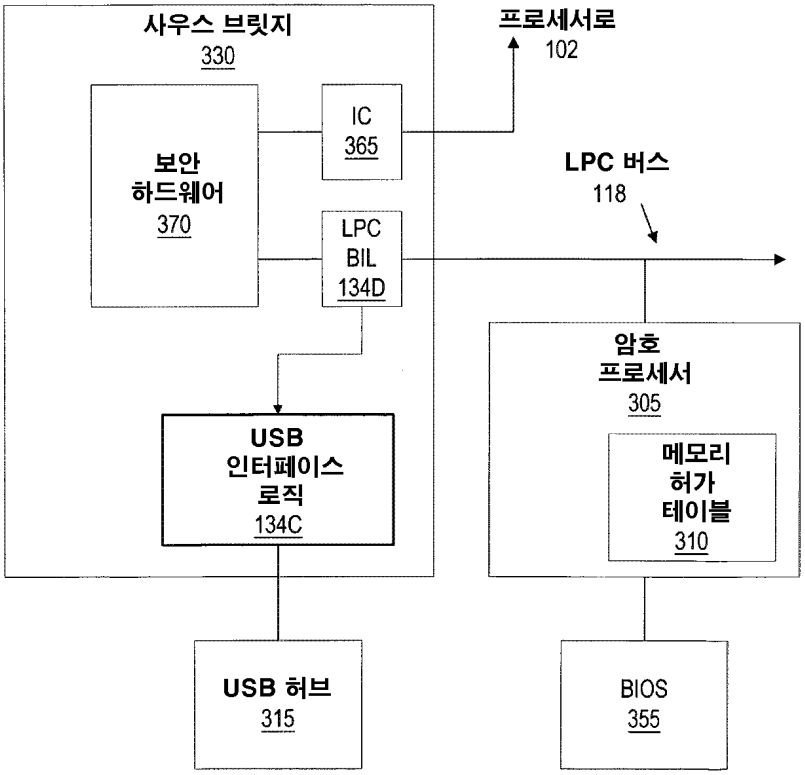
도면2B


170
인터럽트 제어기가 시스템관리 모드 (SMM)에 대한 요청을 수신함 172
인터럽트 제어기가 시스템 관리 인터럽트 (SMI#) 시그널을 어써트함으로써 프로세서에 SMM에 대한 요청을 시그널링함 174
프로세서가 SMM에 대한 요청을 인식하여, SMI 활성 (SMIACT#) 시그널을 어써트함 176
시스템이 SMIACT# 시그널을 인식하여, RAM으로의 액세스를 디스에이블시키고, 시스템 관리 RAM (SMRAM) 공간으로의 액세스를 인에이블시킴 178
현재의 프로세서 상태가 SMRAM에 저장됨 180
프로세서가 SMM 디폴트 상태로 리셋되고, SMM에 들어감 182
프로세서가 디폴트 포인터를 판독하여, SMRAM 공간으로 점프함 184
SMI 요청을 식별하기 위해 상태 레지스터들이 검사됨 186
SMI 처리기가 SMI 요청을 서비스함 188
SMI 처리기가 복귀를 SMM (RSM) 명령으로부터 프로세서로 송신함 190
프로세서가 저장된 상태 정보를 재저장하고, 장상 동작을 계속함 192

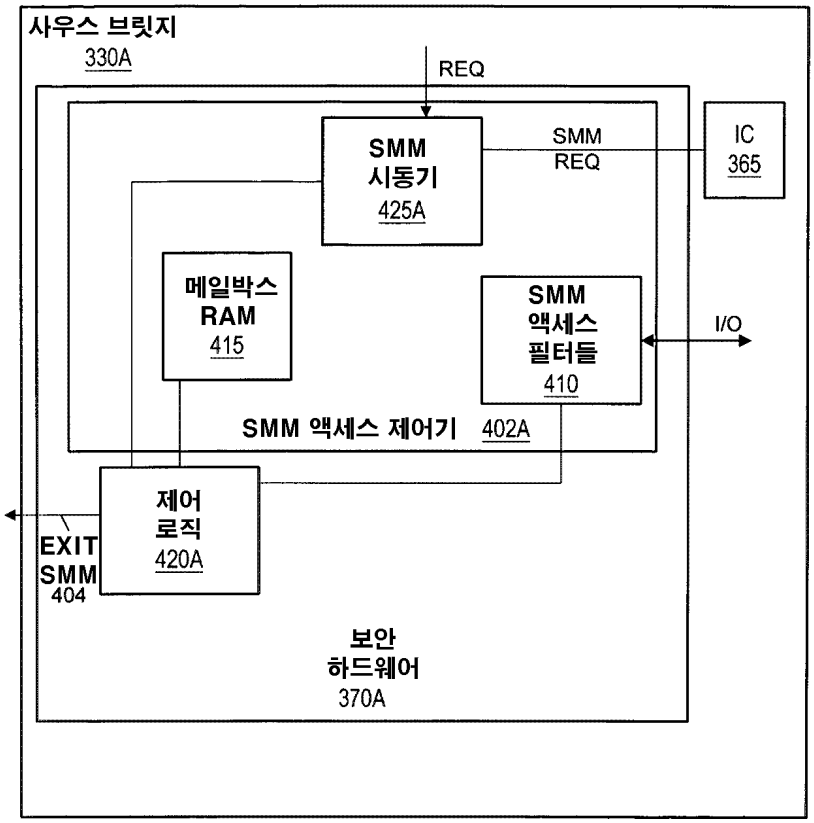
도면3


사용자
입력/
출력
205
200
어플리케이션들
210
암호화
서비스
제공자들
215
API
호출들
220
보안
실행
박스
260
드라이버들
225
하드웨어
230

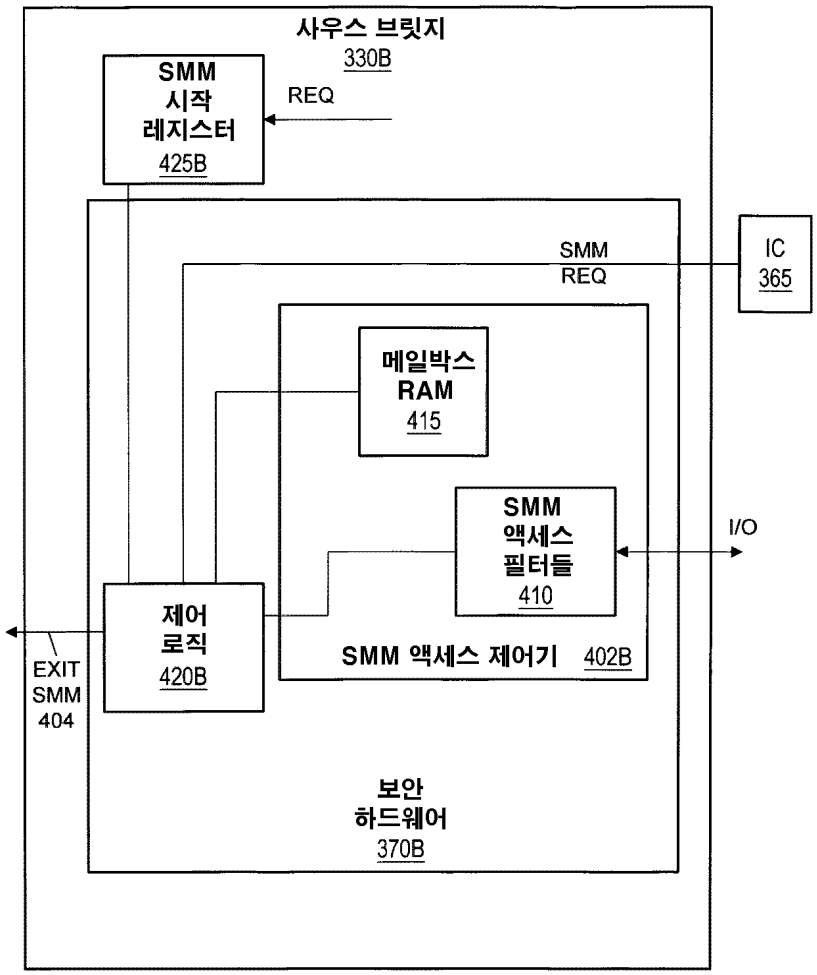
도면4


사우스 브릿지
330
보안
하드웨어
370
IC
365
LPC
BIL
134D
USB
인터페이스
로직
134C
USB 허브
315
프로세서로
102
LPC 버스
118
암호
프로세서
305
메모리
허가
테이블
310
BIOS
355

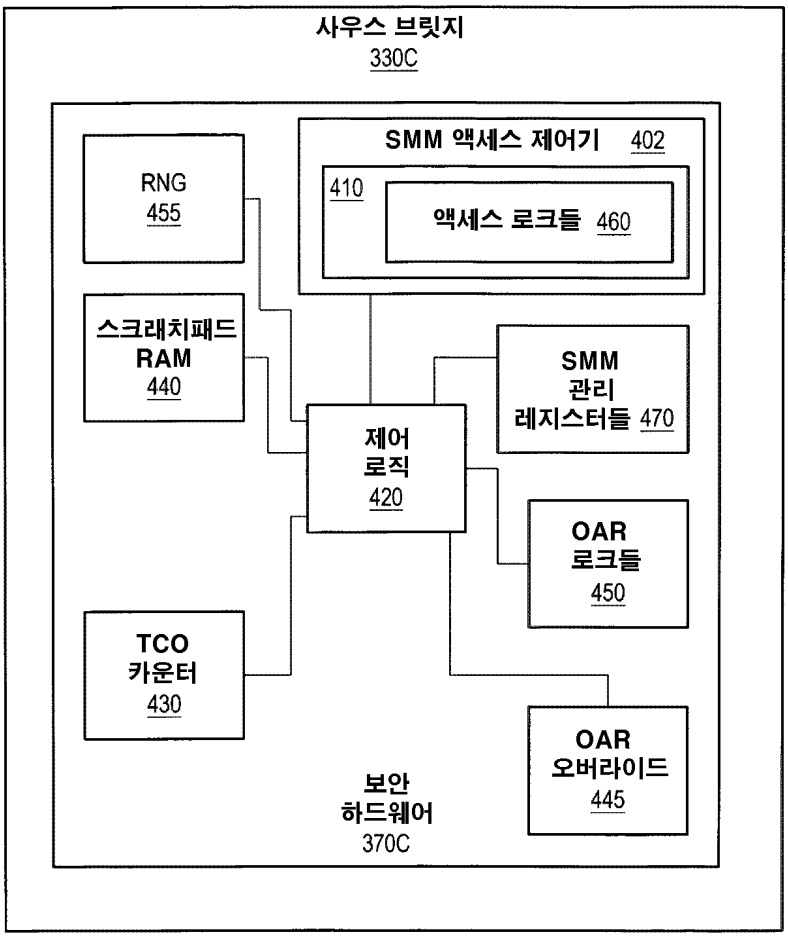
도면5A


사우스 브릿지
330A
REQ
SMM
시동기
425A
SMM
REQ
IC
365
메일박스
RAM
415
SMM
액세스
필터들
410
I/O
SMM 액세스 제어기 402A
제어
로직
420A
EXIT
SMM
404
보안
하드웨어
370A

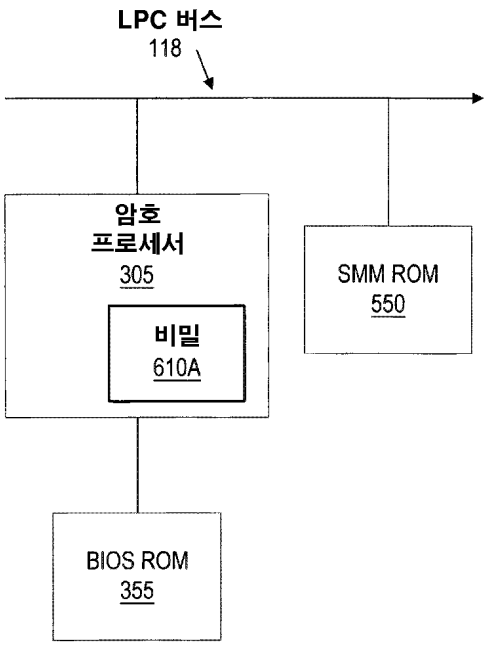
도면5B


사우스 브릿지
330B
SMM
시작
레지스터
425B
REQ
SMM
REQ
IC
365
메일박스
RAM
415
SMM
액세스
필터들
410
I/O
제어
로직
420B
SMM 액세스 제어기 402B
EXIT
SMM
404
보안
하드웨어
370B

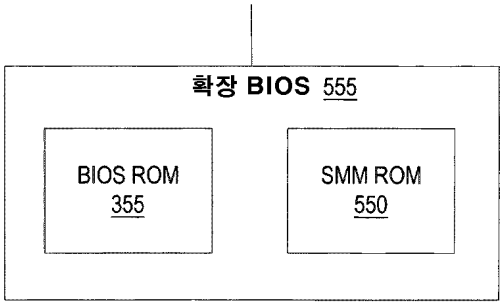
도면6


사우스 브릿지
330C
SMM 액세스 제어기 402
410
액세스 로크들 460
RNG
455
스크래치패드
RAM
440
SMM
관리
레지스터들 470
제어
로직
420
OAR
로크들
450
TCO
카운터
430
OAR
오버라이드
445
보안
하드웨어
370C

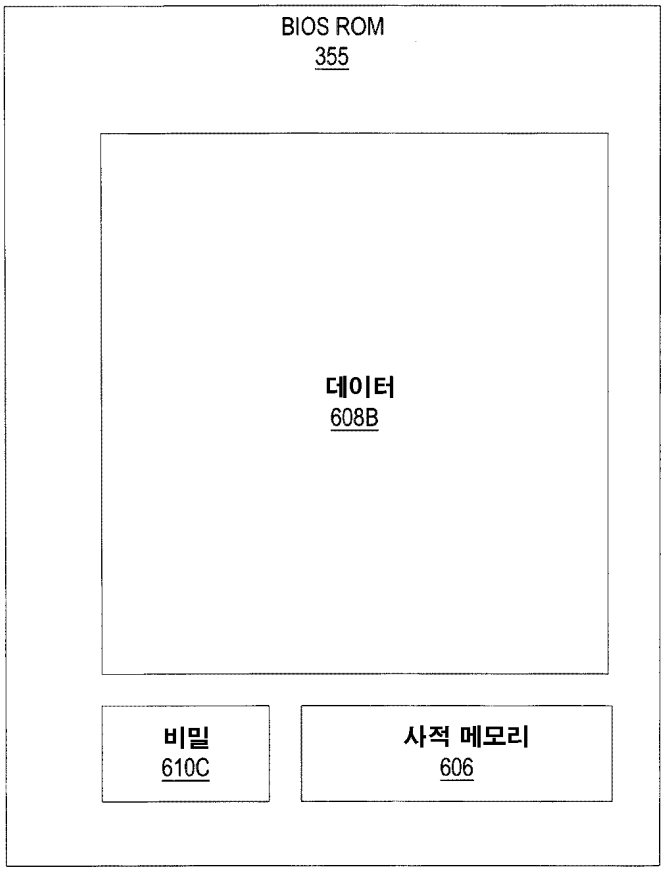
도면7A


LPC 버스
118
암호
프로세서
305
비밀
610A
SMM ROM
550
BIOS ROM
355

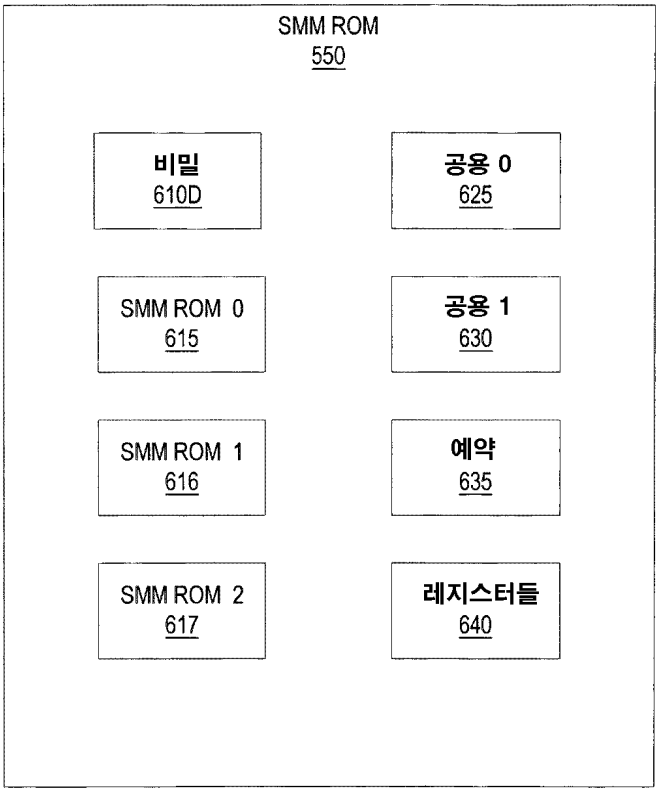
도면7B


확장 BIOS 555
BIOS ROM
355
SMM ROM
550

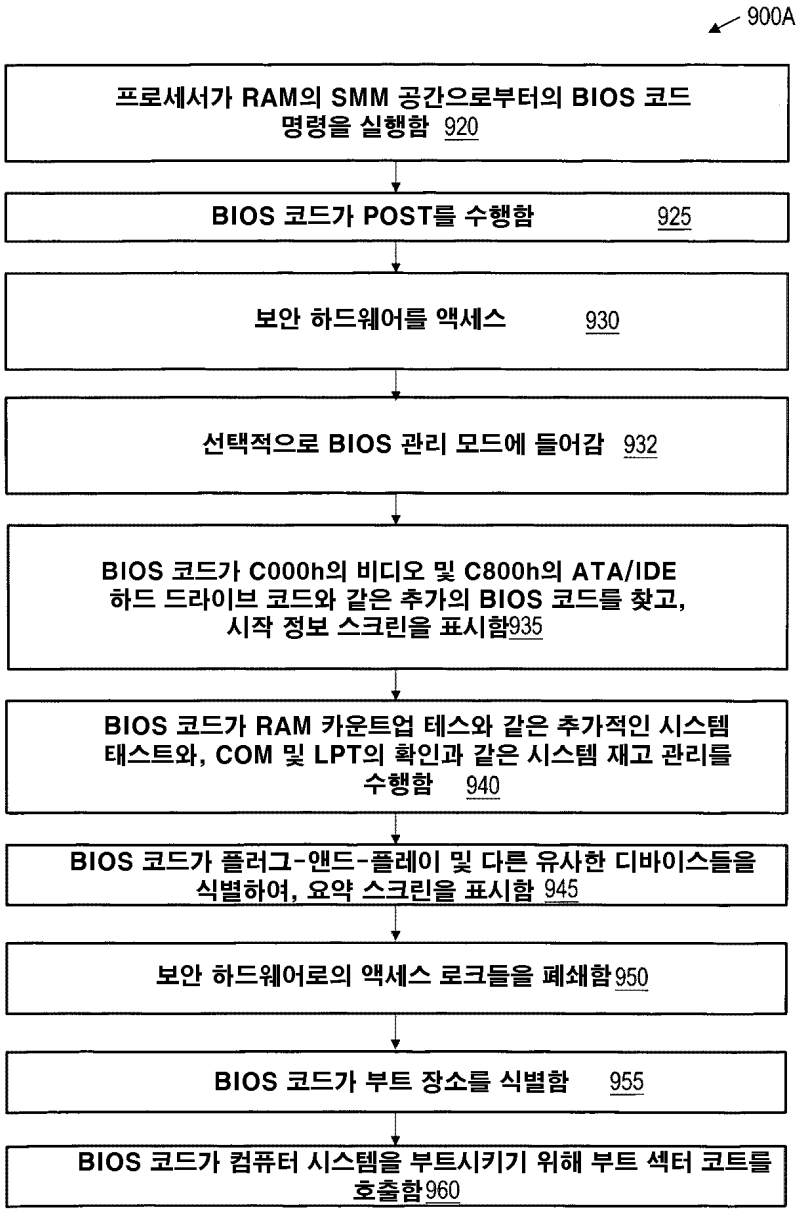
도면8A


BIOS ROM
355
데이터
608B
비밀
610C
사적 메모리
606

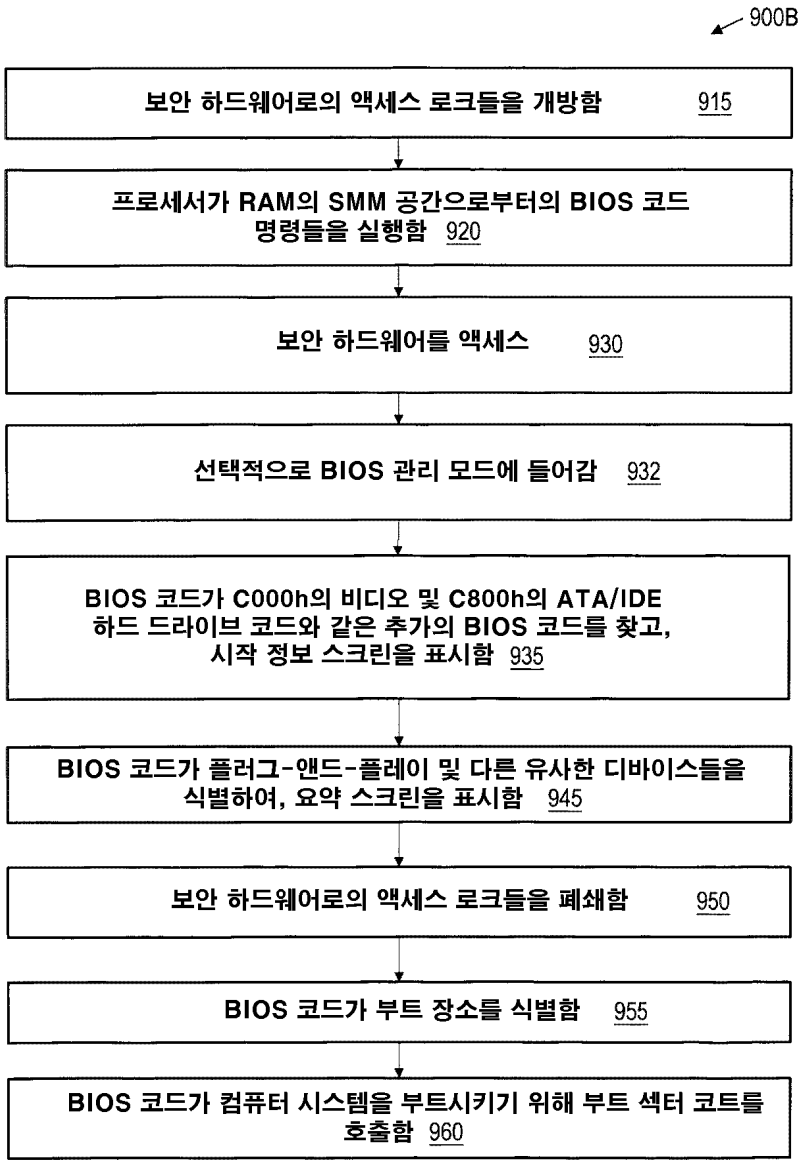
도면8B


SMM ROM
550
비밀
610D
공용 0
625
SMM ROM 0
615
공용 1
630
SMM ROM 1
616
예약
635
SMM ROM 2
617
레지스터들
640

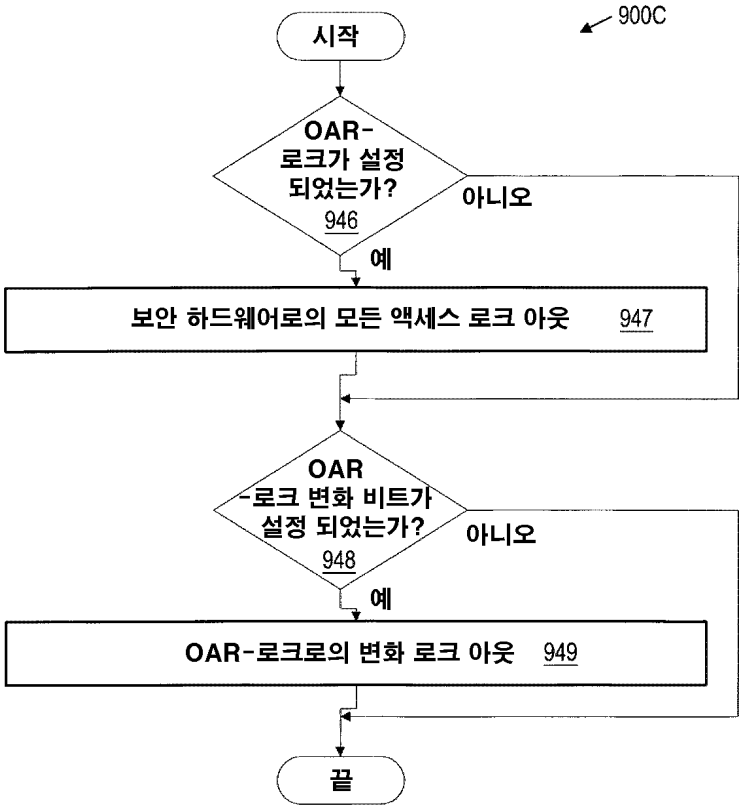
도면9A


900A
프로세서가 RAM의 SMM 공간으로부터의 BIOS 코드 명령을 실행함 920
BIOS 코드가 POST를 수행함 925
보안 하드웨어를 액세스 930
선택적으로 BIOS 관리 모드에 들어감 932
BIOS 코드가 C000h의 비디오 및 C800h의 ATA/IDE 하드 드라이브 코드와 같은 추가의 BIOS 코드를 찾고, 시작 정보 스크린을 표시함935
BIOS 코드가 RAM 카운트업 테스와 같은 추가적인 시스템 태스트와, COM 및 LPT의 확인과 같은 시스템 재고 관리를 수행함 940
BIOS 코드가 플러그-앤드-플레이 및 다른 유사한 디바이스들을 식별하여, 요약 스크린을 표시함 945
보안 하드웨어로의 액세스 로크들을 폐쇄함950
BIOS 코드가 부트 장소를 식별함 955
BIOS 코드가 컴퓨터 시스템을 부트시키기 위해 부트 섹터 코트를 호출함960

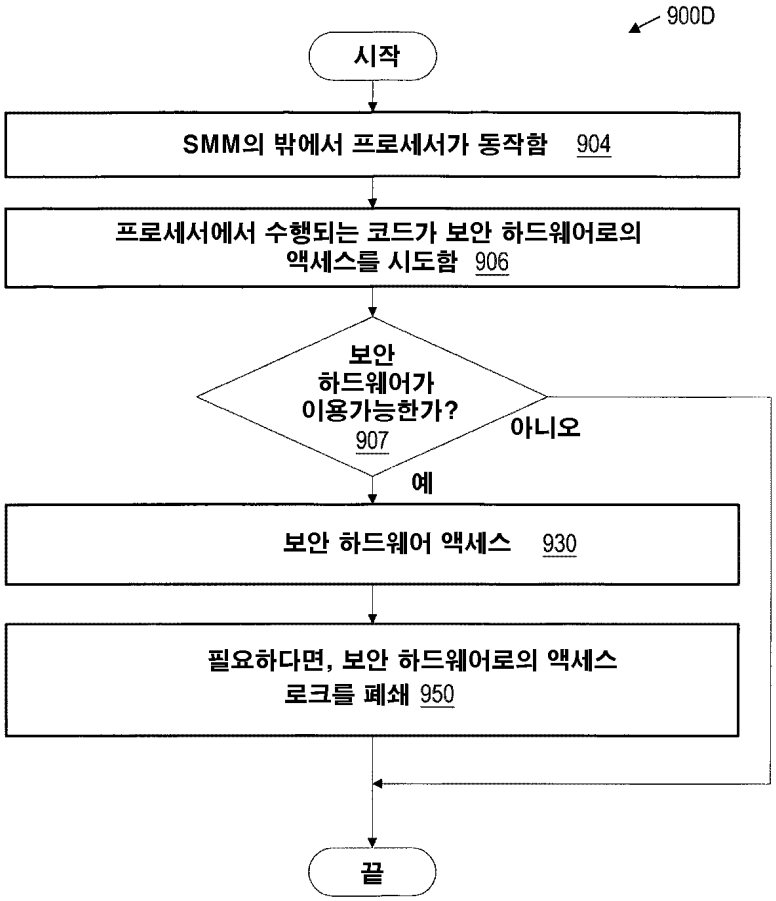
도면9B


900B
보안 하드웨어로의 액세스 로크들을 개방함 915
프로세서가 RAM의 SMM 공간으로부터의 BIOS 코드 명령들을 실행함 920
보안 하드웨어를 액세스 930
선택적으로 BIOS 관리 모드에 들어감 932
BIOS 코드가 C000h의 비디오 및 C800h의 ATA/IDE 하드 드라이브 코드와 같은 추가의 BIOS 코드를 찾고, 시작 정보 스크린을 표시함 935
BIOS 코드가 플러그-앤드-플레이 및 다른 유사한 디바이스들을 식별하여, 요약 스크린을 표시함 945
보안 하드웨어로의 액세스 로크들을 폐쇄함 950
BIOS 코드가 부트 장소를 식별함 955
BIOS 코드가 컴퓨터 시스템을 부트시키기 위해 부트 섹터 코트를 호출함 960

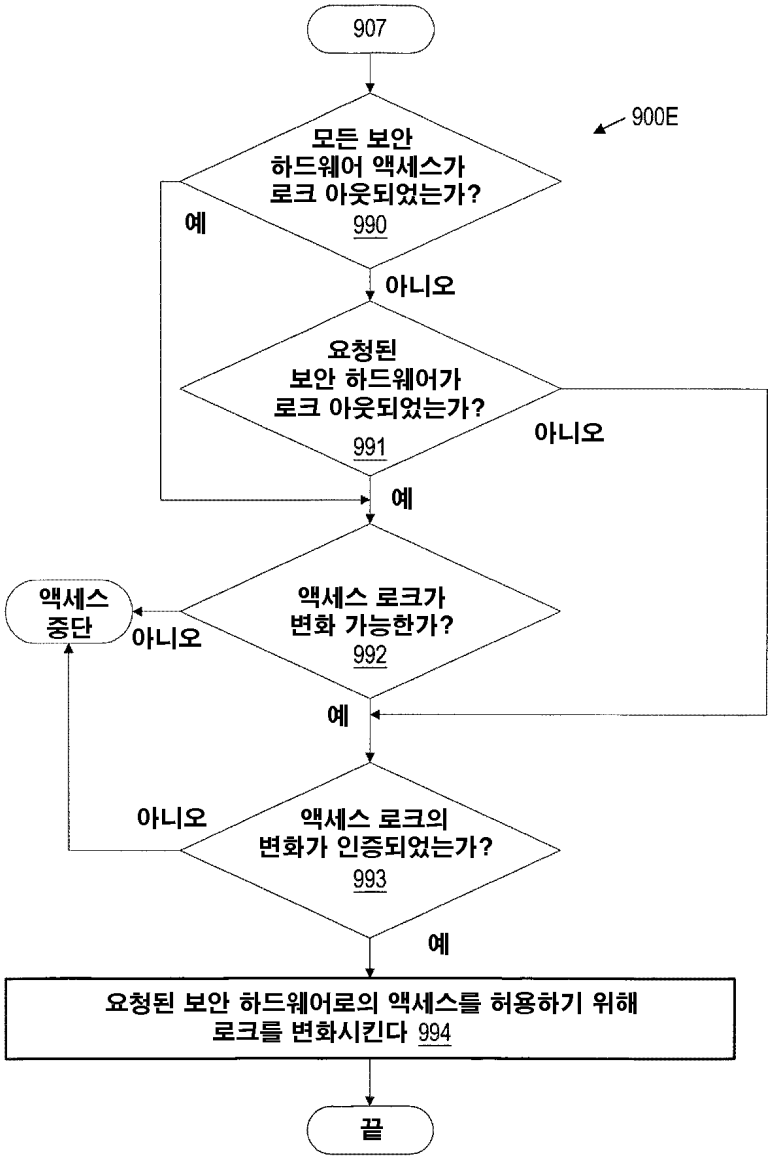
도면9C


900C
시작
OAR-로크가 설정 되었는가?
946
아니오
예
보안 하드웨어로의 모든 액세스 로크 아웃 947
OAR -로크 변화 비트가 설정 되었는가?
948
아니오
예
OAR-로크로의 변화 로크 아웃 949
끝

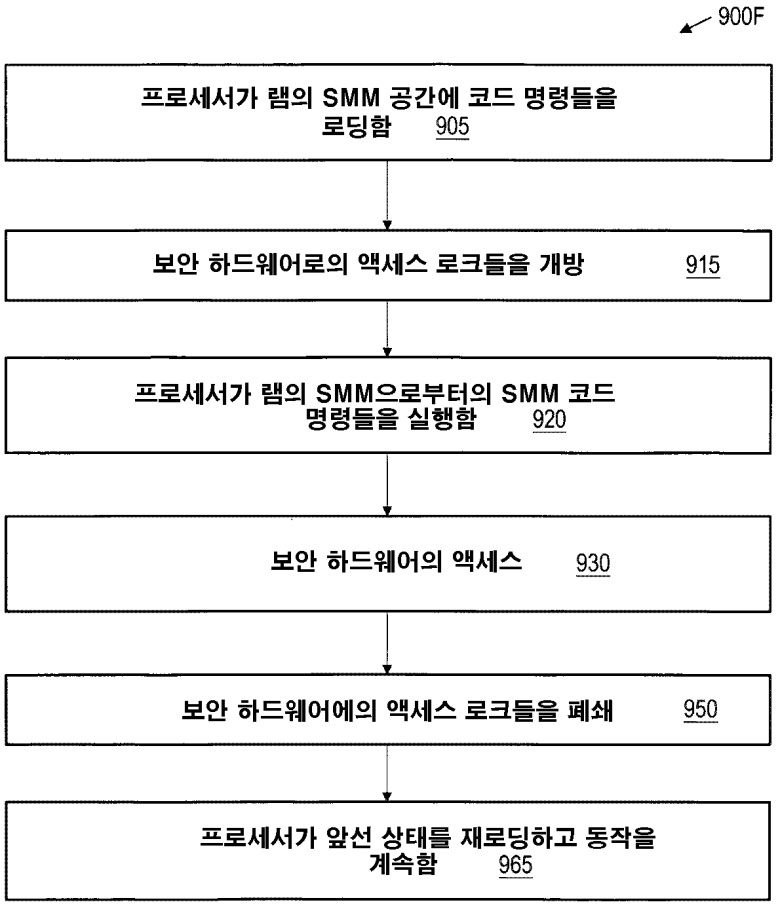
도면9D


900D
시작
SMM의 밖에서 프로세서가 동작함 904
프로세서에서 수행되는 코드가 보안 하드웨어로의 액세스를 시도함 906
보안 하드웨어가 이용가능한가? 907
아니오
예
보안 하드웨어 액세스 930
필요하다면, 보안 하드웨어로의 액세스 로크를 폐쇄 950
끝

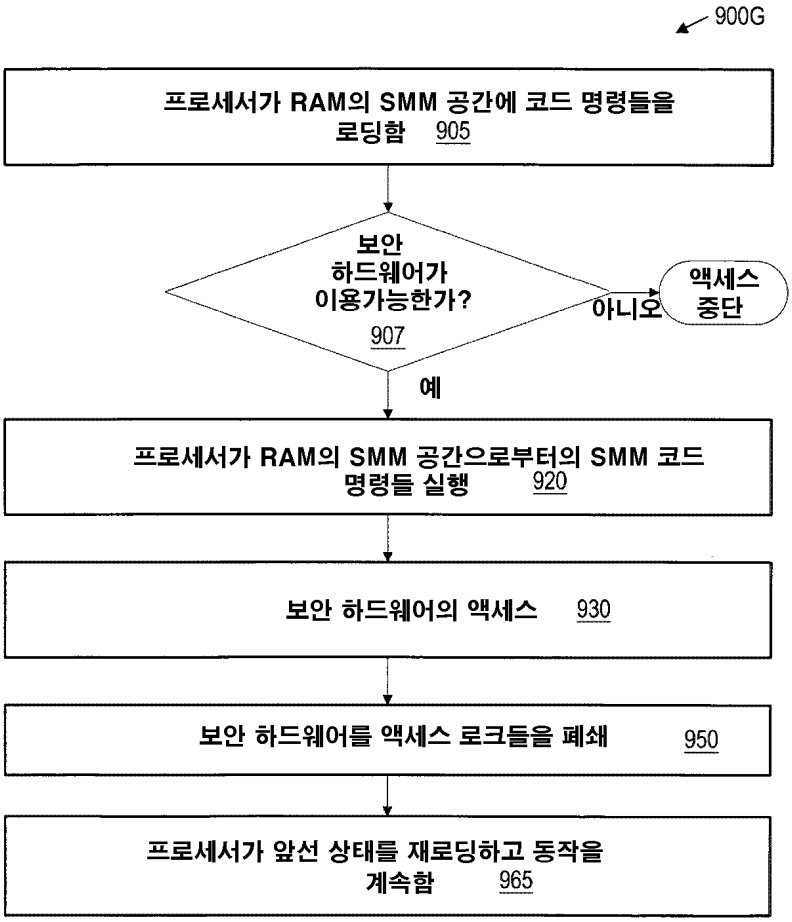
도면9E


907
900E
모든 보안 하드웨어 액세스가 로크 아웃되었는가? 990
예
아니오
요청된 보안 하드웨어가 로크 아웃되었는가? 991
아니오
예
액세스 로크가 변화 가능한가? 992
액세스 중단
아니오
예
액세스 로크의 변화가 인증되었는가? 993
아니오
예
요청된 보안 하드웨어로의 액세스를 허용하기 위해 로크를 변화시킨다 994
끝

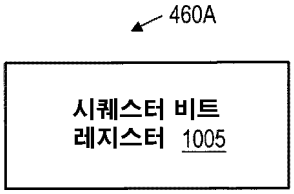
도면9F


900F
프로세서가 램의 SMM 공간에 코드 명령들을 로딩함 905
보안 하드웨어로의 액세스 로크들을 개방 915
프로세서가 램의 SMM으로부터의 SMM 코드 명령들을 실행함 920
보안 하드웨어의 액세스 930
보안 하드웨어에의 액세스 로크들을 폐쇄 950
프로세서가 앞선 상태를 재로딩하고 동작을 계속함 965

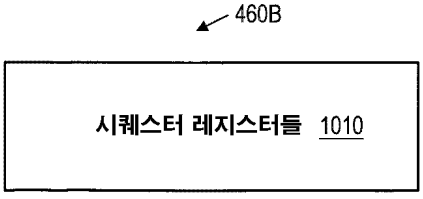
도면9G



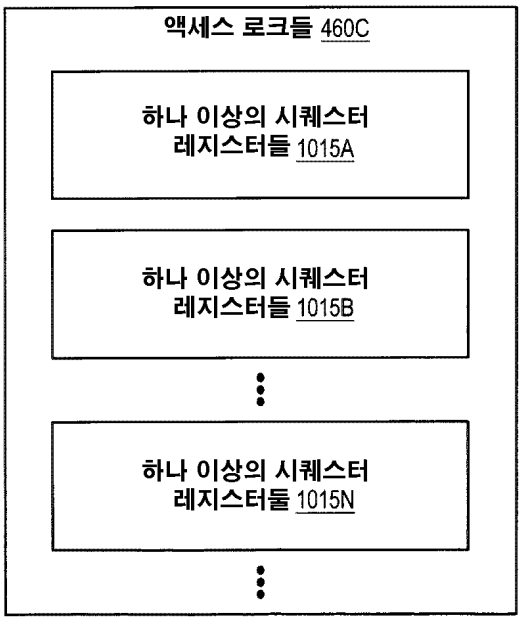
도면10A



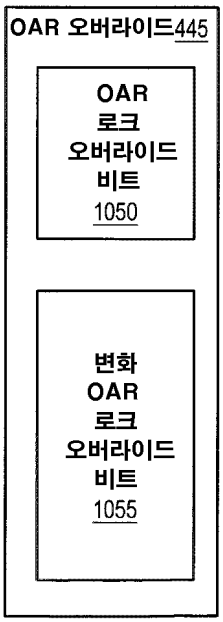
도면10B



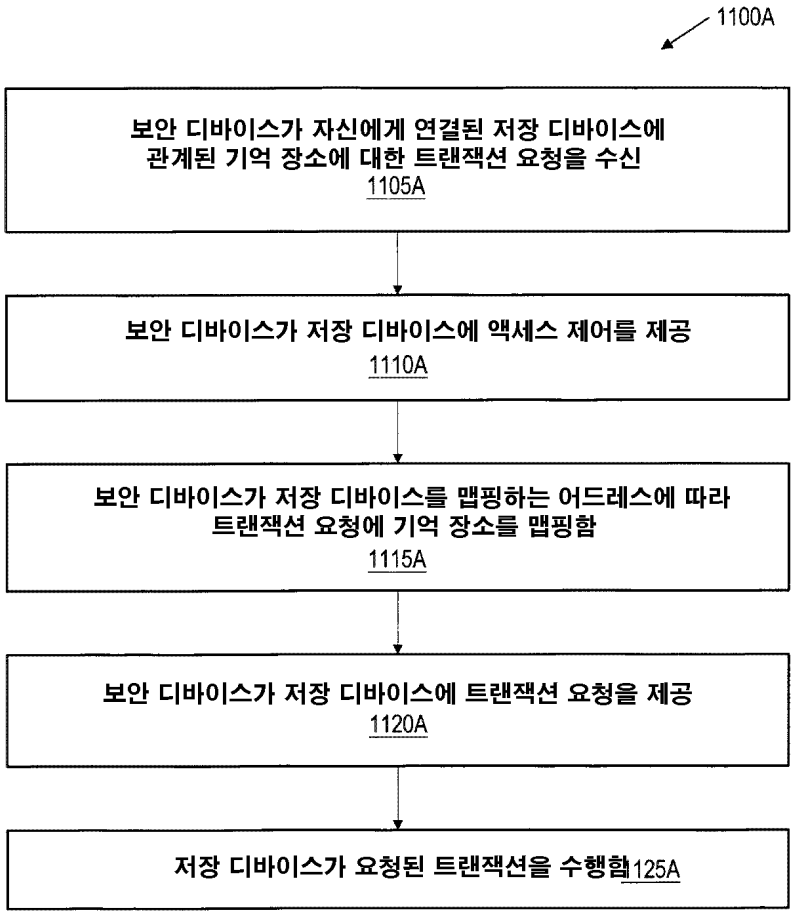
도면10C


액세스 로크들 460C
하나 이상의 시퀘스터 레지스터들 1015A
하나 이상의 시퀘스터 레지스터들 1015B
하나 이상의 시퀘스터 레지스터들 1015N

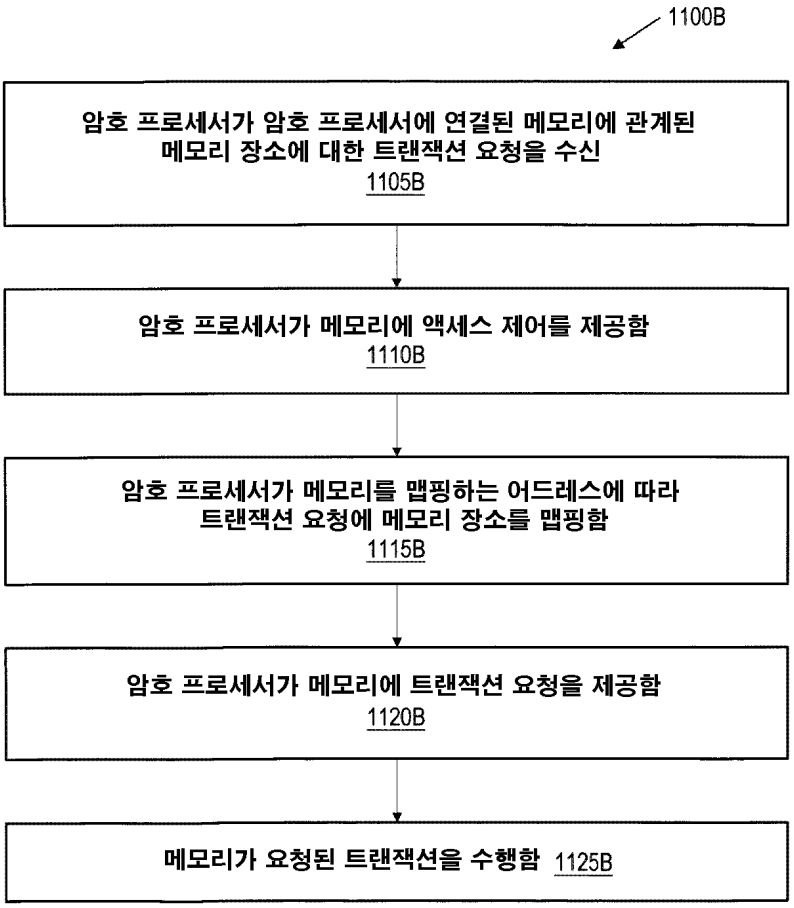
도면10D


OAR 오버라이드 445
OAR 로크 오버라이드 비트 1050
변화 OAR 로크 오버라이드 비트 1055

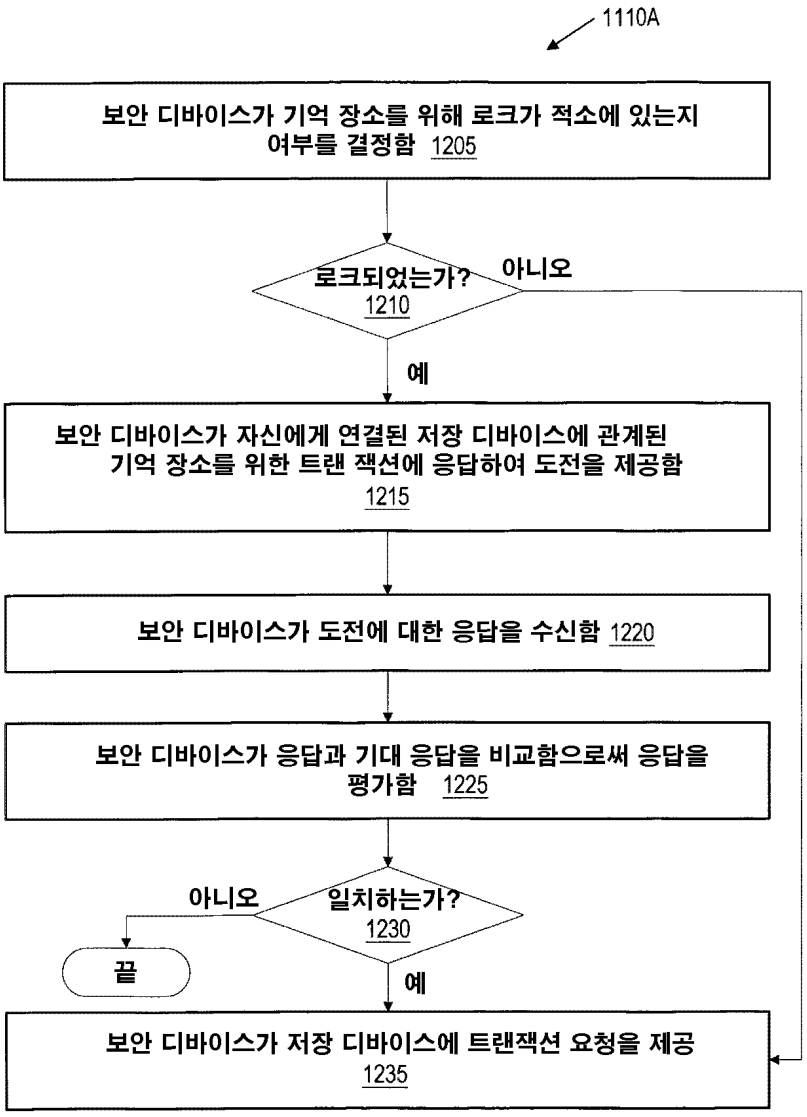
도면11A


1100A
보안 디바이스가 자신에게 연결된 저장 디바이스에 관계된 기억 장소에 대한 트랜잭션 요청을 수신
1105A
보안 디바이스가 저장 디바이스에 액세스 제어를 제공
1110A
보안 디바이스가 저장 디바이스를 맵핑하는 어드레스에 따라 트랜잭션 요청에 기억 장소를 맵핑함
1115A
보안 디바이스가 저장 디바이스에 트랜잭션 요청을 제공
1120A
저장 디바이스가 요청된 트랜잭션을 수행함1125A

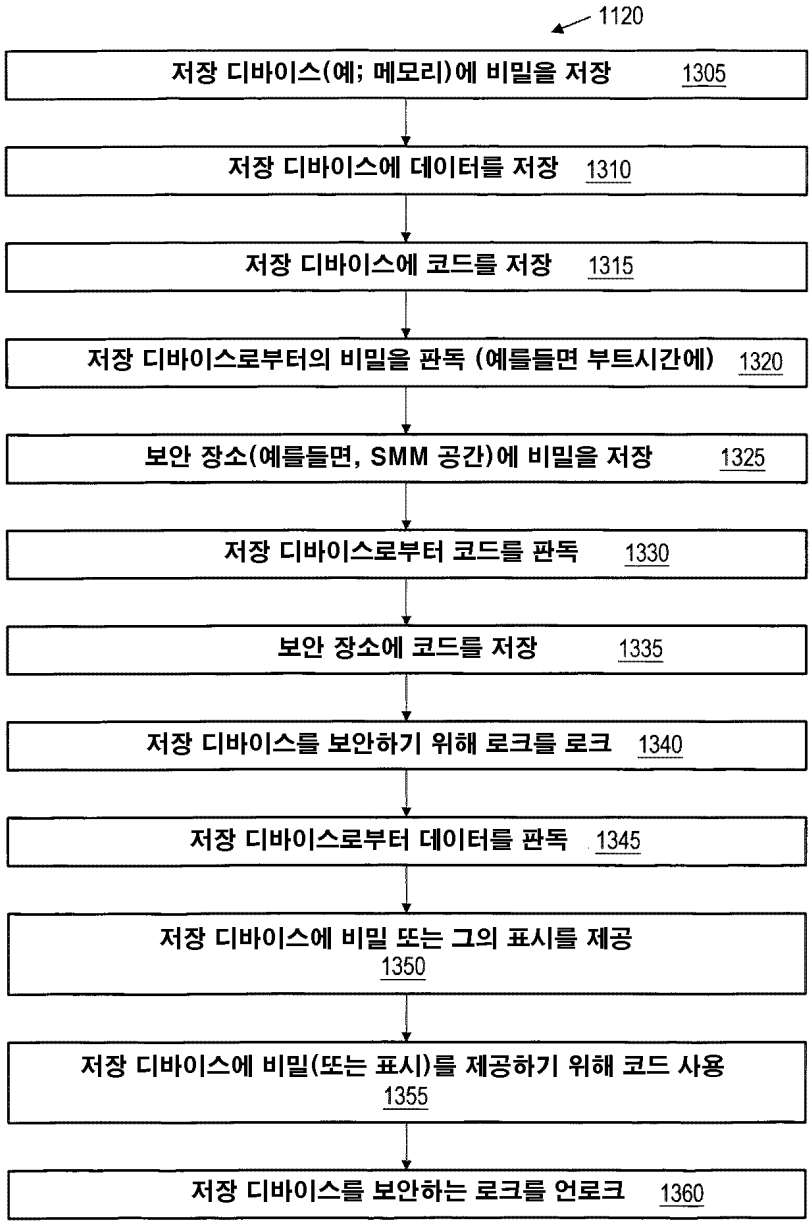
도면11B


1100B
암호 프로세서가 암호 프로세서에 연결된 메모리에 관계된 메모리 장소에 대한 트랜잭션 요청을 수신
1105B
암호 프로세서가 메모리에 액세스 제어를 제공함
1110B
암호 프로세서가 메모리를 맵핑하는 어드레스에 따라 트랜잭션 요청에 메모리 장소를 맵핑함
1115B
암호 프로세서가 메모리에 트랜잭션 요청을 제공함
1120B
메모리가 요청된 트랜잭션을 수행함 1125B

도면12


1110A
보안 디바이스가 기억 장소를 위해 로크가 적소에 있는지 여부를 결정함 1205
로크되었는가? 1210
아니오
예
보안 디바이스가 자신에게 연결된 저장 디바이스에 관계된 기억 장소를 위한 트랜 잭션에 응답하여 도전을 제공함 1215
보안 디바이스가 도전에 대한 응답을 수신함 1220
보안 디바이스가 응답과 기대 응답을 비교함으로써 응답을 평가함 1225
일치하는가? 1230
아니오
끝
예
보안 디바이스가 저장 디바이스에 트랜잭션 요청을 제공 1235

도면13


1120
저장 디바이스(예; 메모리)에 비밀을 저장 1305
저장 디바이스에 데이터를 저장 1310
저장 디바이스에 코드를 저장 1315
저장 디바이스로부터의 비밀을 판독 (예를들면 부트시간에) 1320
보안 장소(예를들면, SMM 공간)에 비밀을 저장 1325
저장 디바이스로부터 코드를 판독 1330
보안 장소에 코드를 저장 1335
저장 디바이스를 보안하기 위해 로크를 로크 1340
저장 디바이스로부터 데이터를 판독 1345
저장 디바이스에 비밀 또는 그의 표시를 제공 1350
저장 디바이스에 비밀(또는 표시)를 제공하기 위해 코드 사용 1355
저장 디바이스를 보안하는 로크를 언로크 1360