(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년06월27일
(11) 등록번호 10-1751627
(24) 등록일자 2017년06월21일

(51) 국제특허분류(Int. Cl.)
G06F 9/455 (2006.01) *G06F 21/53* (2013.01)
G06F 9/50 (2006.01)
(52) CPC특허분류
G06F 9/45545 (2013.01)
G06F 21/53 (2013.01)
(21) 출원번호 10-2015-7027352
(22) 출원일자(국제) 2014년03월10일
심사청구일자 2016년08월31일
(85) 번역문제출일자 2015년10월02일
(65) 공개번호 10-2015-0128797
(43) 공개일자 2015년11월18일
(86) 국제출원번호 PCT/US2014/022731
(87) 국제공개번호 WO 2014/164536
국제공개일자 2014년10월09일
(30) 우선권주장
13/796,442 2013년03월12일 미국(US)
(56) 선행기술조사문헌
US20050076156 A1
(뒷면에 계속)
전체 청구항 수 : 총 48 항

(73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
정 토마스
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
투즈니 아제딘
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
(74) 대리인
특허법인코리아나

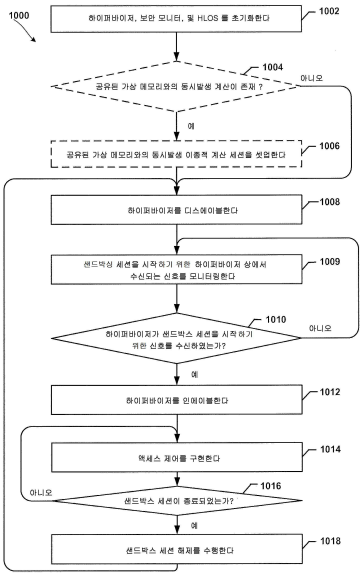
심사관 : 유진태

(54) 발명의 명칭 **온디맨드로 가상 머신 모니터의 동작들을 선택적으로 인에이블하기 위한 방법 및 장치**

(57) 요 약

여러 양태들에서, 성능을 향상시키고, 샌드박스 세션들 동안 컴퓨팅 디바이스 상에서 동작하는 하이퍼바이저를 선택적으로 인에이블함으로써 소모되는 전력의 양을 감소시키기 위해 가상화 기법들이 이용될 수도 있다. 여러 양태들에서, 하이-레벨 운영 시스템은 그의 중간 물리 어드레스들이 물리 어드레스들과 동일하게 메모리를 할당할 수도 있다. 하이퍼바이저가 디스에이블될 때, 하이퍼바이저는 중간 물리 어드레스들로부터 물리 어드레스들로의 제 2 스테이지 변환들을 일시정지할 수도 있다. 샌드박스 세션 동안, 하이퍼바이저는 인에이블되어, 제 2 스테이지 변환들을 수행하는 것을 재개할 수도 있다.

대 표 도 - 도10



(52) CPC특허분류
G06F 9/45533 (2013.01)
G06F 9/45558 (2013.01)
G06F 9/5016 (2013.01)
G06F 2009/45583 (2013.01)
G06F 2009/45587 (2013.01)
Y02B 60/146 (2013.01)

(72) 발명자
뮬러 필립 주니어
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
파텔 피유쉬
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(56) 선행기술조사문헌
US20070226795 A1
US20050091365 A1
US20050289542 A1
US20050223220 A1
JP2012220990 A

명 세 서

청구범위

청구항 1

컴퓨팅 디바이스 상에서 메모리를 관리하는 방법으로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 단계;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 단계;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 단계;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 단계; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 단계

를 포함하고,

상기 하이퍼바이저를 디스에이블하는 단계는 :

제 2 스테이지 변환 (translation) 을 바이패스하도록 모든 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 구성하는 단계; 및

상기 HLOS 에 대한 제 2 스테이지 변환들을 턴오프하는 단계

를 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 2

제 1 항에 있어서,

상기 하이퍼바이저를 디스에이블하는 단계는 :

하드웨어 인터럽트들에의 상기 HLOS 의 액세스들을 제한하는 것을 일시정지하는 단계;

하드웨어 타이머들에의 상기 HLOS 의 액세스들을 제한하는 것을 일시정지하는 단계; 및

상기 HLOS 의 I/O 액세스들을 제한하는 것을 일시정지하는 단계

중 적어도 하나를 더 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 3

컴퓨팅 디바이스 상에서 메모리를 관리하는 방법으로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 단계;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 단계;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 단계;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 단계;

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 단계;

상기 샌드박스 세션이 종료되었는지 여부를 결정하는 단계;

상기 샌드박스 세션이 종료되었다고 결정하는 것에 응답하여 샌드박스 세션 해제 프로시저를 수행하는 단계; 및

상기 샌드박스 세션 해제 프로시저를 수행한 후에 상기 하이퍼바이저를 디스에이블하는 단계

를 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 4

제 3 항에 있어서,

상기 샌드박스 세션이 종료되었는지 여부를 결정하는 단계는 상기 샌드박스 세션이 종료되었다는 것을 표시하는 또 다른 신호를 수신하는 단계를 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 5

제 3 항에 있어서,

상기 샌드박스 세션 해제 프로시저를 수행하는 단계는 :

샌드박스화된 컴포넌트에 대한 모든 버퍼들을 해방 (freeing) 하는 단계; 및

모든 단편화들을 제거하기 위해 제 2 스테이지 변환 페이지 테이블들을 복원하는 단계

를 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 6

컴퓨팅 디바이스 상에서 메모리를 관리하는 방법으로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 단계;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 단계;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 단계;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 단계; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 단계

를 포함하고,

상기 하이퍼바이저를 인에이블하는 단계는 :

PL0 및 PL1 제 2 스테이지 메모리 관리 유닛들을 인에이블하는 단계;

인터럽트 요청들을 하이퍼바이저 모드에서 취해지도록 구성하는 단계; 및

모든 활성 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 (nested) 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 단계

를 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 7

제 6 항에 있어서,

상기 모든 활성 SMMU 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 단계 이후에,

디지털 신호 프로세서와 프로세서간 통신을 시작하는 단계를 더 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 8

제 6 항에 있어서,

상기 모든 활성 SMMU 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 단계 이후에,

SMMU 결함들을 핸들링하는 단계를 더 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 9

컴퓨팅 디바이스 상에서 메모리를 관리하는 방법으로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 단계;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 단계;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 단계;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 단계; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 단계

를 포함하고,

상기 액세스 제어를 구현하는 단계는 제 2 스테이지 변환들을 구현하는 단계를 포함하고 :

하드웨어 인터럽트들에의 상기 HLOS 의 액세스들을 제한하는 것을 재개하는 단계;

하드웨어 타이머들에의 상기 HLOS 의 액세스들을 제한하는 것을 재개하는 단계; 및

상기 HLOS 의 I/O 액세스들을 제한하는 것을 재개하는 단계

중 적어도 하나를 더 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 10

컴퓨팅 디바이스 상에서 메모리를 관리하는 방법으로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 단계;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 단계;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 단계;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 단계; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 단계

를 포함하고,

상기 액세스 제어를 구현하는 단계는 제 2 스테이지 변환들을 구현하는 단계를 포함하고,

상기 제 2 스테이지 변환들을 구현하는 단계는 :

상기 HLOS 가 메모리를 할당하려는 시도를 모니터링하는 단계; 및

상기 HLOS 가 메모리를 할당하려고 시도할 때, 상기 HLOS 에 액세스가능한 물리 어드레스 공간에서의 하나 이상의 물리 어드레스들을 상기 HLOS 에 제공하는 단계

를 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 11

제 10 항에 있어서,

상기 HLOS 가 메모리를 할당하려는 시도를 모니터링하는 단계 이후에,

상기 HLOS 가 메모리를 할당하려고 시도하지 않을 때, 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있는지 여부를 결정하는 단계; 및

상기 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있다고 결정하는 것에 응답하여, 상기 물리 어드

레스 공간에서의 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 단계

를 더 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 12

제 11 항에 있어서,

상기 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 단계는 :

상기 HLOS 에 액세스가능한 상기 물리 어드레스 공간에서의 상기 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 제공될 상기 물리 어드레스들을 제거하는 단계; 및

상기 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 상기 물리 어드레스들을 제공하는 단계

를 포함하는, 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법.

청구항 13

컴퓨팅 디바이스로서,

메모리; 및

상기 메모리에 커플링된 프로세서를 포함하고,

상기 프로세서는,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것

을 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되고,

상기 프로세서는, 상기 하이퍼바이저를 디스에이블하는 것이 :

제 2 스테이지 변환을 바이패스하도록 모든 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 구성하는 것; 및

상기 HLOS 에 대한 제 2 스테이지 변환들을 턴오프하는 것

을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 14

제 13 항에 있어서,

상기 프로세서는, 상기 하이퍼바이저를 디스에이블하는 것이 :

하드웨어 인터럽트들에의 상기 HLOS 의 액세스들을 제한하는 것을 일시정지하는 것;

하드웨어 타이머들에의 상기 HLOS 의 액세스들을 제한하는 것을 일시정지하는 것; 및

상기 HLOS 의 I/O 액세스들을 제한하는 것을 일시정지하는 것

중 적어도 하나를 더 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 15

컴퓨팅 디바이스로서,

메모리; 및

상기 메모리에 커플링된 프로세서를 포함하고,

상기 프로세서는,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것;

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것;

상기 샌드박스 세션이 종료되었는지 여부를 결정하는 것;

상기 샌드박스 세션이 종료되었다고 결정하는 것에 응답하여 샌드박스 세션 해제 프로시저를 수행하는 것; 및

상기 샌드박스 세션 해제 프로시저를 수행한 후에 상기 하이퍼바이저를 디스에이블하는 것

을 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 16

제 15 항에 있어서,

상기 프로세서는, 상기 샌드박스 세션이 종료되었는지 여부를 결정하는 것이 상기 샌드박스 세션이 종료되었다는 것을 표시하는 또 다른 신호를 수신하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 17

제 15 항에 있어서,

상기 프로세서는, 상기 샌드박스 세션 해제 프로시저를 수행하는 것이 :

샌드박스화된 컴포넌트에 대한 모든 버퍼들을 해방 (freeing) 하는 것; 및

모든 단편화들을 제거하기 위해 제 2 스테이지 변환 페이지 테이블들을 복원하는 것

을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 18

컴퓨팅 디바이스로서,

메모리; 및

상기 메모리에 커플링된 프로세서를 포함하고,

상기 프로세서는,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것

을 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되고,

상기 프로세서는, 상기 하이퍼바이저를 인에이블하는 것이 :

PL0 및 PL1 제 2 스테이지 메모리 관리 유닛들을 인에이블하는 것;

인터럽트 요청들을 하이퍼바이저 모드에서 취해지도록 구성하는 것; 및

모든 활성 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 (nested) 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 것

을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는,컴퓨팅 디바이스.

청구항 19

제 18 항에 있어서,

상기 프로세서는, 디지털 신호 프로세서와 프로세서간 통신을 시작하는 것을 더 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 20

제 18 항에 있어서,

상기 프로세서는, SMMU 결함들을 핸들링하는 것을 더 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 21

컴퓨팅 디바이스로서,

메모리; 및

상기 메모리에 커플링된 프로세서를 포함하고,

상기 프로세서는,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것

을 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되고,

상기 프로세서는, 상기 액세스 제어를 구현하는 것이 제 2 스테이지 변환들을 구현하는 것을 포함하고 상기 액세스 제어를 구현하는 것이 :

하드웨어 인터럽트들에의 상기 HLOS 의 액세스들을 제한하는 것을 재개하는 것;

하드웨어 타이머들에의 상기 HLOS 의 액세스들을 제한하는 것을 재개하는 것; 및

상기 HLOS 의 I/O 액세스들을 제한하는 것을 재개하는 것

중 적어도 하나를 더 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 22

컴퓨팅 디바이스로서,

메모리; 및

상기 메모리에 커플링된 프로세서를 포함하고,

상기 프로세서는,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것

을 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되고,

상기 프로세서는,

상기 액세스 제어를 구현하는 것이 제 2 스테이지 변환들을 구현하는 것을 포함하고;

상기 제 2 스테이지 변환들을 구현하는 것이 :

상기 HLOS 가 메모리를 할당하려는 시도를 모니터링하는 것; 및

상기 HLOS 가 메모리를 할당하려고 시도할 때, 상기 HLOS 에 액세스가능한 물리 어드레스 공간에서의 하나 이상의 물리 어드레스들을 상기 HLOS 에 제공하는 것

을 포함하도록

동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 23

제 22 항에 있어서,

상기 프로세서는 :

상기 HLOS 가 메모리를 할당하려고 시도하지 않을 때, 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있는지 여부를 결정하는 것; 및

상기 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있다고 결정하는 것에 응답하여, 상기 물리 어드레스 공간에서의 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것

을 더 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 24

제 23 항에 있어서,

상기 프로세서는, 상기 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것이 :

상기 HLOS 에 액세스가능한 상기 물리 어드레스 공간에서의 상기 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 제공될 상기 물리 어드레스들을 제거하는 것; 및

상기 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 상기 물리 어드레스들을 제공하는 것

을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 25

컴퓨팅 디바이스로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 수단;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 수단;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 수단;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 수단; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 수단

을 포함하고,

상기 하이퍼바이저를 디스에이블하는 수단은 :

제 2 스테이지 변환을 바이패스하도록 모든 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 구성하는 수단; 및

상기 HLOS 에 대한 제 2 스테이지 변환들을 턴오프하는 수단

을 포함하는, 컴퓨팅 디바이스.

청구항 26

제 25 항에 있어서,

상기 하이퍼바이저를 디스에이블하는 수단은 :

하드웨어 인터럽트들에의 상기 HLOS 의 액세스들을 제한하는 것을 일시정지하는 수단;

하드웨어 타이머들에의 상기 HLOS 의 액세스들을 제한하는 것을 일시정지하는 수단; 및

상기 HLOS 의 I/O 액세스들을 제한하는 것을 일시정지하는 수단

중 적어도 하나를 더 포함하는, 컴퓨팅 디바이스.

청구항 27

컴퓨팅 디바이스로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 수단;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 수단;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 수단;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 수단;

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 수단;

상기 샌드박스 세션이 종료되었는지 여부를 결정하는 수단;

상기 샌드박스 세션이 종료되었다고 결정하는 것에 응답하여 샌드박스 세션 해제 프로시저를 수행하는 수단; 및

상기 샌드박스 세션 해제 프로시저를 수행한 후에 상기 하이퍼바이저를 디스에이블하는 수단

을 포함하는, 컴퓨팅 디바이스.

청구항 28

제 27 항에 있어서,

상기 샌드박스 세션이 종료되었는지 여부를 결정하는 수단은 상기 샌드박스 세션이 종료되었다는 것을 표시하는 또 다른 신호를 수신하는 수단을 포함하는, 컴퓨팅 디바이스.

청구항 29

제 27 항에 있어서,

상기 샌드박스 세션 해제 프로시저를 수행하는 수단은 :

샌드박스화된 컴포넌트에 대한 모든 버퍼들을 해방 (freeing) 하는 수단; 및

모든 단편화들을 제거하기 위해 제 2 스테이지 변환 페이지 테이블들을 복원하는 수단

을 포함하는, 컴퓨팅 디바이스.

청구항 30

컴퓨팅 디바이스로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 수단;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 수단;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 수단;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 수단; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 수단

을 포함하고,

상기 하이퍼바이저를 인에이블하는 수단은 :

PL0 및 PL1 제 2 스테이지 메모리 관리 유닛들을 인에이블하는 수단;

인터럽트 요청들을 하이퍼바이저 모드에서 취해지도록 구성하는 수단; 및

모든 활성 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 (nested) 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 수단

을 포함하는, 컴퓨팅 디바이스.

청구항 31

제 30 항에 있어서,

디지털 신호 프로세서와 프로세서간 통신을 시작하는 수단을 더 포함하는, 컴퓨팅 디바이스.

청구항 32

제 30 항에 있어서,

SMMU 결함들을 핸들링하는 수단을 더 포함하는, 컴퓨팅 디바이스.

청구항 33

컴퓨팅 디바이스로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 수단;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 수단;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 수단;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 수단; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 수단

을 포함하고,

상기 액세스 제어를 구현하는 수단은 제 2 스테이지 변환들을 구현하는 수단을 포함하고 :

하드웨어 인터럽트들에의 상기 HLOS 의 액세스들을 제한하는 것을 재개하는 수단;

하드웨어 타이머들에의 상기 HLOS 의 액세스들을 제한하는 것을 재개하는 수단; 및

상기 HLOS 의 I/O 액세스들을 제한하는 것을 재개하는 수단

중 적어도 하나를 더 포함하는, 컴퓨팅 디바이스.

청구항 34

컴퓨팅 디바이스로서,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 수단;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 수단;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 수단;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 수단; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 수단

을 포함하고,

상기 액세스 제어를 구현하는 수단은 제 2 스테이지 변환들을 구현하는 수단을 포함하고;

상기 제 2 스테이지 변환들을 구현하는 수단은 :

상기 HLOS 가 메모리를 할당하려는 시도를 모니터링하는 수단; 및

상기 HLOS 가 메모리를 할당하려고 시도할 때, 상기 HLOS 에 액세스가능한 물리 어드레스 공간에서의 하나 이상의 물리 어드레스들을 상기 HLOS 에 제공하는 수단

을 포함하는, 컴퓨팅 디바이스.

청구항 35

제 34 항에 있어서,

상기 HLOS 가 메모리를 할당하려고 시도하지 않을 때, 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있는지 여부를 결정하는 수단; 및

상기 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있다고 결정하는 것에 응답하여, 상기 물리 어드레스 공간에서의 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 수단

을 더 포함하는, 컴퓨팅 디바이스.

청구항 36

제 35 항에 있어서,

상기 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 수단은 :

상기 HLOS 에 액세스가능한 상기 물리 어드레스 공간에서의 상기 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 제공될 상기 물리 어드레스들을 제거하는 수단; 및

상기 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 상기 물리 어드레스들을 제공하는 수단

을 포함하는, 컴퓨팅 디바이스.

청구항 37

프로세서로 하여금, 컴퓨팅 디바이스 상에서 메모리를 관리하는 동작들을 수행하게 하도록 구성된 프로세서 실행가능한 소프트웨어 명령들을 저장하고 있는 비일시적 프로세서 판독가능 저장 매체로서,

상기 동작들은,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것

을 포함하고,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금, 상기 하이퍼바이저를 디스에이블하는 것이 :

제 2 스테이지 변환을 바이패스하도록 모든 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 구성하는 것; 및

상기 HLOS 에 대한 제 2 스테이지 변환들을 턴오프하는 것

을 포함하도록 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 38

제 37 항에 있어서,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금, 상기 하이퍼바이저를 디스에이블하는 것이 :

하드웨어 인터럽트들에의 상기 HLOS 의 액세스들을 제한하는 것을 일시정지하는 것;

하드웨어 타이머들에의 상기 HLOS 의 액세스들을 제한하는 것을 일시정지하는 것; 및

상기 HLOS 의 I/O 액세스들을 제한하는 것을 일시정지하는 것

중 적어도 하나를 더 포함하도록 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 39

프로세서로 하여금, 컴퓨팅 디바이스 상에서 메모리를 관리하는 동작들을 수행하게 하도록 구성된 프로세서 실행가능한 소프트웨어 명령들을 저장하고 있는 비일시적 프로세서 판독가능 저장 매체로서,

상기 동작들은,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것;

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것;

상기 샌드박스 세션이 종료되었는지 여부를 결정하는 것;

상기 샌드박스 세션이 종료되었다고 결정하는 것에 응답하여 샌드박스 세션 해제 프로시저를 수행하는 것; 및

상기 샌드박스 세션 해제 프로시저를 수행한 후에 상기 하이퍼바이저를 디스에이블하는 것

을 포함하는, 비일시적 프로세서 판독가능 저장 매체.

청구항 40

제 39 항에 있어서,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금, 상기 샌드박스 세션이 종료되었는지 여부를 결정하는 것이 상기 샌드박스 세션이 종료되었다는 것을 표시하는 또 다른 신호를 수신하는 것을 포함하도록 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 41

제 39 항에 있어서,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금, 상기 샌드박스 세션 해제 프로시저를 수행하는 것이 :

샌드박스화된 컴포넌트에 대한 모든 버퍼들을 해방 (freeing) 하는 것; 및

모든 단편화들을 제거하기 위해 제 2 스테이지 변환 페이지 테이블들을 복원하는 것

을 포함하도록 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 42

프로세서로 하여금, 컴퓨팅 디바이스 상에서 메모리를 관리하는 동작들을 수행하게 하도록 구성된 프로세서 실행가능한 소프트웨어 명령들을 저장하고 있는 비일시적 프로세서 판독가능 저장 매체로서,

상기 동작들은,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것

을 포함하고,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금, 상기 하이퍼바이저를 인에이블하는 것이 :

PL0 및 PL1 제 2 스테이지 메모리 관리 유닛들을 인에이블하는 것;

인터럽트 요청들을 하이퍼바이저 모드에서 취해지도록 구성하는 것; 및

모든 활성 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 (nested) 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 것

을 포함하도록 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 43

제 42 항에 있어서,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금, 디지털 신호 프로세서와 프로세서 간 통신을 시작하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 44

제 42 항에 있어서,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금, SMMU 결함들을 핸들링하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 45

프로세서로 하여금, 컴퓨팅 디바이스 상에서 메모리를 관리하는 동작들을 수행하게 하도록 구성된 프로세서 실행가능한 소프트웨어 명령들을 저장하고 있는 비일시적 프로세서 판독가능 저장 매체로서,

상기 동작들은,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것; 및

상기 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것

을 포함하고,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금, 상기 액세스 제어를 구현하는 것이 제 2 스테이지 변환들을 구현하는 것을 포함하고 :

하드웨어 인터럽트들에의 상기 HLOS 의 액세스들을 제한하는 것을 재개하는 것;

하드웨어 타이머들에의 상기 HLOS 의 액세스들을 제한하는 것을 재개하는 것; 및

상기 HLOS 의 I/O 액세스들을 제한하는 것을 재개하는 것

중 적어도 하나를 더 포함하도록 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 46

프로세서로 하여금, 컴퓨팅 디바이스 상에서 메모리를 관리하는 동작들을 수행하게 하도록 구성된 프로세서 실행가능한 소프트웨어 명령들을 저장하고 있는 비일시적 프로세서 판독가능 저장 매체로서,

상기 동작들은,

하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것;

초기화 이후 상기 하이퍼바이저를 디스에이블하는 것;

샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 신호를 모니터링하는 것;

상기 샌드박스 세션을 시작하기 위한 상기 보안 모니터로부터의 상기 신호를 수신하는 것에 응답하여 상기 하이퍼바이저를 인에이블하는 것; 및

상기 하이퍼바이저가 인에이블되는 액세스 제어를 수행하는 것

을 포함하고,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금 :

상기 액세스 제어를 구현하는 것이 제 2 스테이지 변환들을 구현하는 것을 포함하고;

상기 제 2 스테이지 변환들을 구현하는 것이 :

상기 HLOS 가 메모리를 할당하려는 시도를 모니터링하는 것; 및

상기 HLOS 가 메모리를 할당하려고 시도할 때, 상기 HLOS 에 액세스가능한 물리 어드레스 공간에서의 하나 이상의 물리 어드레스들을 상기 HLOS 에 제공하는 것

을 포함하도록

동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 47

제 46 항에 있어서,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금 :

상기 HLOS 가 메모리를 할당하려고 시도하지 않을 때, 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있는지 여부를 결정하는 것; 및

상기 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있다고 결정하는 것에 응답하여, 상기 물리 어드레스 공간에서의 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것

을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 48

제 47 항에 있어서,

저장된 상기 프로세서 실행가능한 소프트웨어 명령들은, 프로세서로 하여금, 상기 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것이 :

상기 HLOS 에 액세스가능한 상기 물리 어드레스 공간에서의 상기 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 제공될 상기 물리 어드레스들을 제거하는 것; 및

상기 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 상기 샌드박스화된 컴포넌트에 상기 물리 어드레스들을 제공하는 것

을 포함하도록 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독가능 저장 매체.

청구항 49

삭제

청구항 50

삭제

청구항 51

삭제

청구항 52

삭제

청구항 53

삭제

청구항 54

삭제

청구항 55

삭제

청구항 56

삭제

청구항 57

삭제

청구항 58

삭제

청구항 59

삭제

청구항 60

삭제

청구항 61

삭제

청구항 62

삭제

청구항 63

삭제

청구항 64

삭제

청구항 65

삭제

청구항 66

삭제

청구항 67

삭제

청구항 68

삭제

청구항 69

삭제

청구항 70

삭제

청구항 71

삭제

청구항 72

삭제

청구항 73

삭제

청구항 74

삭제

청구항 75

삭제

청구항 76

삭제

발명의 설명

배 경 기 술

[0001] 일반적으로, 가상화 기술은 운영 시스템과 하드웨어 사이에 소프트웨어 제어 프로그램 (예컨대, 가상 머신 모니터 "VMM" 또는 하이퍼바이저) 을 배치함으로써 컴퓨팅 리소스들의 추상화 (또는 가상화) 를 가능하게 한다. 하이퍼바이저는 특권 모드에서 실행하며, 다수의 운영 시스템들 (게스트 운영 시스템들로 지칭됨) 을 호스트할 수도 있다. 각각의 게스트 운영 시스템은 물리적인 하드웨어와 통신하는 방법과 동일한 방법으로 하이퍼바이저와 통신하므로, 하이퍼바이저와 하드웨어의 조합을 단일, 가상 머신으로서 간주한다. 이것은 각각의 게스트 운영 시스템으로 하여금, 프로세서들, 주변장치들, 메모리 및 I/O 에 배타적으로 액세스한다는 착각 하에서 동작가능하게 한다.

[0002] 운영 시스템들은 다수의 프로세스들에 걸쳐서 물리 메모리를 파티셔닝하는 것을 담당한다. 가상 머신 상에서 실행하는 게스트 운영 시스템을 포함하는 시스템들에서, 게스트 운영 시스템에 의해 할당되는 메모리는 실제 물리 메모리가 아닌, 중간 물리 메모리이다. 이러한 시스템들 상에서, 하이퍼바이저는 물리 메모리의 실제 할당을 담당한다.

[0003] 대부분의 프로세서들은 단지 하나의 메모리 어드레스 공간 변환의 스테이지를 지원하며, 하이퍼바이저는 가상 어드레스들 (VA), 중간 물리 어드레스들 (IPA), 및 물리 어드레스들 (PA) 사이의 관계를 관리한다. 이것은 게스트 운영 시스템의 변환 테이블들의 각각을 해석함으로써 유도되는 그 자신의 변환 테이블들 (쉐도우 변환 테이블들로 불림) 을 유지하는 하이퍼바이저에 의해 일반적으로 달성된다. 구체적으로 설명하면, 하이퍼바이저는 게스트 운영 시스템의 변환 테이블들에 대한 모든 변화들이 쉐도우 구조들에 반영되도록 보장할 뿐만 아니라, 액세스 결함 (fault) 들을 적합한 스테이지로 리다이렉트 (redirect) 하는 것 및 보호를 시행한다.

[0004] 위에서 설명된 단일 스테이지 프로세서들과는 달리, ARM 프로세서 시스템들은 (예컨대, 시스템 메모리 관리 유닛 "SMMU" 와 같은 ARM 가상화 확장들을 통해서) 메모리 변환의 스테이지들 양쪽에 하드웨어 지원을 제공한다. 예를 들어, ARM 프로세서들은 제 1 스테이지 (즉, 제 1 스테이지 변환) 에서 가상 어드레스들 (VA) 이 중간 물리 어드레스들 (IPA) 로 변환되고 제 2 스테이지 (즉, 제 2 스테이지 변환) 에서 중간 물리 어드레스들 (IPA) 이 물리 어드레스들로 변환되는 2개의 스테이지 변환을 가능하게 하는 가상화 확장들을 포함한다. 이것은 하이퍼바이저와 연관되는 오버헤드들을 감소시킨다.

발명의 내용

과제의 해결 수단

[0005] 여러 양태들은 요구될 때 데이터 및/또는 소프트웨어를 보호하는 액세스 제어를 효율적으로 시행하도록 하이퍼바이저를 선택적으로 구현하는 컴퓨팅 디바이스들 및 방법들을 포함한다. 여러 양태들에서, 하이퍼바이저는 보통은 디스에이블되며, 하이퍼바이저에 액세스 제어 (즉, "샌드박스 세션") 를 구현하도록 요구하는 컨디션이 검출될 때 인에이블된다. 하이-레벨 운영 시스템 (HLOS) 은 하이퍼바이저에 의해 관리되는 가상 머신에서 동작할 수도 있다. HLOS 는 가상 어드레스들을 HLOS 상에서 실행하는 여러 프로세스들 또는 애플리케이션들에 할당할 때에 사용하기 위해 중간 물리 어드레스 페이지 테이블을 유지할 수도 있다. HLOS 는 물리 메모리 어드레스 공간으로부터 직접 메모리를 할당함으로써, 중간 물리 메모리 어드레스들이 물리 메모리 어드레스들과 항상 동일하도록 보장할 수도 있다. 중간 물리 어드레스들이 물리 어드레스들과 언제나 동일하게 HLOS 가 메모리를 할당할 수 있도록 보장함으로써, 하이퍼바이저는 선택적으로, 샌드박스화된 세션에 대한 요구가 존재할 때 인에이블될 수도 있으며, 어떤 현재의 샌드박스화된 세션도 존재하지 않을 때 디스에이블될 수도 있다.

디스에이블되는 동안, 하이퍼바이저는 중간 물리 어드레스들로부터 물리 어드레스들로의 스테이지 2 변환들을 수행하지 않을 수도 있다. 또한, 하이퍼바이저가 디스에이블되는 동안, HLOS 는 전체 물리 메모리 어드

레스 공간으로부터 메모리를 할당할 수도 있다.

[0006] 여러 양태들에서, 하이퍼바이저는 샌드박스 세션의 지속기간 동안 인에이블될 수도 있다. 인에이블되는 동안, 하이퍼바이저는 중간 물리 어드레스들로부터 물리 어드레스들로의 스테이지 2 변환들을 수행하는 것을 재개할 수도 있다. 또한, 인에이블되는 동안, 하이퍼바이저는 물리 메모리 어드레스 공간에의 HLOS 의 액세스를 제한함으로써, HLOS 로 하여금 단지 물리 메모리 어드레스 공간의 일부로부터 메모리를 할당가능하게 하며, 일부 양태들에서는, 하드웨어 인터럽트들 및/또는 하드웨어 타이머들에의 HLOS 의 액세스를 제한할 수도 있다. 특히, 샌드박싱이 요구되지 않는 동안 (즉, 하이퍼바이저가 디스에이블되는 동안) 스테이지 2 변환들을 수행하지 않게 하이퍼바이저를 구성함으로써, 여러 양태들은 컴퓨팅 디바이스의 전체 성능을 향상시키면서, 적합한 경우 필요한 보안을 제공할 수도 있다.

[0007] 여러 양태들은 하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하고; 초기화 이후 하이퍼바이저를 디스에이블하고; 샌드박스 세션을 시작하기 위한 보안 모니터로부터의 신호를 모니터링하고; 샌드박스 세션을 시작하기 위한 신호가 수신될 때 하이퍼바이저를 인에이블하고; 그리고 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현함으로써 컴퓨팅 디바이스 상에서 메모리를 관리하는 방법을 포함한다. 일 양태에서, 보안 모니터는 ARM TrustZone® 일 수도 있다. 또 다른 양태에서, 하이퍼바이저는 집적 회로 경계 및 칩 경계 중 적어도 하나에 걸쳐서 디스에이블되거나 또는 인에이블될 수도 있다. 또 다른 양태에서, 하이퍼바이저를 초기화하는 단계는, HLOS 의 중간 물리 어드레스 공간에서의 각각의 중간 물리 어드레스가 물리 어드레스 공간에서의 대응하는 물리 어드레스와 동일하게 메모리 공간을 할당하도록 HLOS 를 구성하는 단계를 포함할 수도 있다. 여전히, 또 다른 양태에서, 하이퍼바이저를 초기화하는 단계는 또한 보안 모니터로 하이퍼바이저의 코드 및 데이터를 인증하는 단계를 포함할 수도 있다. 또 다른 양태에서, 본 방법은 하이퍼바이저가 인에이블되는 동안 디지털 신호 프로세서 및 디지털 신호 프로세서에 포함되는 CPU 중 적어도 하나에 액세스불가능하도록 하이퍼바이저의 코드 및 데이터를 구성하는 단계를 포함할 수도 있다.

[0008] 또 다른 양태에서, 하이퍼바이저를 디스에이블하는 단계는 제 2 스테이지 변환을 바이패스하도록 모든 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 구성하는 단계 및 HLOS 에 대한 제 2 스테이지 변환들을 턴오프하는 단계를 포함할 수도 있다. 일 양태에서, 하이퍼바이저를 디스에이블하는 단계는 하드웨어 인터럽트들에의 HLOS 의 액세스들을 제한하는 것을 일시정지하는 단계, 하드웨어 타이머들에의 HLOS 의 액세스들을 제한하는 것을 일시정지하는 단계, 및 HLOS 의 I/O 액세스들을 제한하는 것을 일시정지하는 단계 중 적어도 하나를 포함할 수도 있다.

[0009] 또 다른 양태에서, 본 방법은 샌드박스 세션이 종료되었는지 여부를 결정하는 단계, 샌드박스 세션이 종료되었다고 결정될 때 샌드박스 세션 해제 (tear-down) 프로시저를 수행하는 단계, 및 샌드박스 세션 해제 프로시저를 수행한 후 하이퍼바이저를 디스에이블하는 단계를 포함할 수도 있다. 또 다른 양태에서, 샌드박스 세션이 종료되었는지 여부를 결정하는 단계는 샌드박스 세션이 종료되었다는 것을 표시하는 또 다른 신호를 수신하는 단계를 포함할 수도 있다. 여전히, 또 다른 양태에서, 샌드박스 세션 해제 프로시저를 수행하는 단계는 샌드박스화된 컴포넌트에 대한 모든 버퍼들을 해방 (freeing) 하는 단계 및 모든 단편화들을 제거하기 위해 제 2 스테이지 변환 페이지 테이블들을 복원하는 단계를 포함할 수도 있다.

[0010] 일 양태에서, 하이퍼바이저를 인에이블하는 단계는, PL0 및 PL1 제 2 스테이지 메모리 관리 유닛들을 인에이블하는 단계, 인터럽트 요청들을 하이퍼바이저 모드에서 취해지도록 구성하는 단계, 및 모든 활성 SMMU 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 (nested) 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 단계를 포함할 수도 있다. 또 다른 양태에서, 본 방법은 또한 디지털 신호 프로세서와 프로세서간 통신들을 시작하는 단계를 포함할 수도 있다. 또 다른 양태에서, 본 방법은 SMMU 결함들을 핸들링하는 단계를 더 포함할 수 있다.

[0011] 또 다른 양태에서, 액세스 제어를 구현하는 단계는 제 2 스테이지 변환들을 구현하는 단계를 포함할 수도 있다. 여전히, 또 다른 양태에서, 액세스 제어를 구현하는 단계는 하드웨어 인터럽트들에의 HLOS 의 액세스들을 제한하는 것을 재개하는 단계, 하드웨어 타이머들에의 HLOS 의 액세스들을 제한하는 것을 재개하는 단계, 및 HLOS 의 I/O 액세스들을 제한하는 것을 재개하는 단계 중 적어도 하나를 포함할 수도 있다. 일 양태에서, 제 2 스테이지 변환들을 구현하는 단계는 HLOS 가 메모리를 할당하려는 시도를 모니터링하는 단계, 및 HLOS 가 메모리를 할당하려고 시도할 때, HLOS 에 액세스가능한 물리 어드레스 공간에서의 하나 이상의 물리 어드레스들을 HLOS 에 제공하는 단계를 포함할 수도 있다.

[0012] 또 다른 양태에서, 본 방법은 HLOS 가 메모리를 할당하려고 시도할 때 샌드박스화된 컴포넌트가 메모리를 할당

하려고 시도하고 있는지 여부를 결정하는 단계 및 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있다고 결정될 때 물리 어드레스 공간에서의 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 단계를 포함할 수도 있다. 일 양태에서, 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 단계는 HLOS 에 액세스가능한 물리 어드레스 공간에서의 물리 어드레스들로부터 샌드박스화된 컴포넌트에 제공될 물리 어드레스들을 제거하는 단계 및 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 단계를 포함할 수도 있다.

[0013] 추가 양태들은 메모리, 및 메모리에 커플링된 프로세서를 포함할 수도 있는 컴퓨팅 디바이스를 포함하며, 프로세서는 하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것, 초기화 이후 하이퍼바이저를 디스에이블하는 것, 샌드박스 세션을 시작하기 위한 보안 모니터로부터의 신호를 모니터링하는 것, 샌드박스 세션을 시작하기 위한 신호가 수신될 때 하이퍼바이저를 인에이블하는 것, 및 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것을 포함할 수도 있는 동작을 수행하는 프로세서 실행가능한 명령들로 구성된다. 또 다른 양태에서, 보안 모니터는 ARM TrustZone® 일 수도 있다. 또 다른 양태에서, 프로세서는 하이퍼바이저가 집적 회로 경계 및 칩 경계 중 적어도 하나에 걸쳐 디스에이블되거나 또는 인에이블될 수 있게 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다.

[0014] 여전히, 또 다른 양태에서, 프로세서는 하이퍼바이저를 초기화하는 것이 HLOS 의 중간 물리 어드레스 공간에서의 각각의 중간 물리 어드레스가 물리 어드레스 공간에서의 대응하는 물리 어드레스와 동일하게 메모리 공간을 할당하도록 HLOS 를 구성하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다. 일 양태에서, 프로세서는 하이퍼바이저를 초기화하는 것이 보안 모니터로 하이퍼바이저의 코드 및 데이터를 인증하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다. 또 다른 양태에서, 프로세서는 하이퍼바이저가 인에이블되는 동안 디지털 신호 프로세서 및 디지털 신호 프로세서에 포함되는 CPU 중 적어도 하나에 액세스불가능하도록 하이퍼바이저의 코드 및 데이터를 구성하는 것을 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다.

[0015] 일 양태에서, 프로세서는 하이퍼바이저를 디스에이블하는 것이 제 2 스테이지 변환을 바이패스하도록 모든 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 구성하는 것, 및 HLOS 에 대한 제 2 스테이지 변환들을 턴오프하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다. 또 다른 양태에서, 프로세서는 하이퍼바이저를 디스에이블하는 것이 하드웨어 인터럽트들에의 HLOS 의 액세스들을 제한하는 것을 일시정지하는 것, 하드웨어 타이머들에의 HLOS 의 액세스들을 제한하는 것을 일시정지하는 것, 및 HLOS 의 I/O 액세스들을 제한하는 것을 일시정지하는 것 중 적어도 하나를 더 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다.

[0016] 또 다른 양태에서, 프로세서는 샌드박스 세션이 종료되었는지 여부를 결정하는 것, 샌드박스 세션이 종료되었다고 결정될 때 샌드박스 세션 해제 프로시저를 수행하는 것, 및 샌드박스 세션 해제 프로시저를 수행한 후 하이퍼바이저를 디스에이블하는 것을 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다.

또 다른 양태에서, 프로세서는 샌드박스 세션이 종료되었는지 여부를 결정하는 것이 샌드박스 세션이 종료되었다는 것을 표시하는 또 다른 신호를 수신하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다. 여전히, 또 다른 양태에서, 프로세서는 샌드박스 세션 해제 프로시저를 수행하는 것이 샌드박스화된 컴포넌트에 대한 모든 버퍼들을 해방하는 것 및 모든 단편화들을 제거하기 위해 제 2 스테이지 변환 페이지 테이블들을 복원하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다.

[0017] 일 양태에서, 프로세서는 하이퍼바이저를 인에이블하는 것이 PL0 및 PL1 제 2 스테이지 메모리 관리 유닛들을 인에이블하는 것, 인터럽트 요청들을 하이퍼바이저 모드에서 취해지도록 구성하는 것, 및 모든 활성 SMMU 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다. 또 다른 양태에서, 프로세서는 디지털 신호 프로세서와 프로세서간 통신들을 시작하는 것을 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다. 여전히, 또 다른 양태에서, 프로세서는 SMMU 결함들을 핸들링하는 것을 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다.

[0018] 일 양태에서, 프로세서는 액세스 제어를 구현하는 것이 제 2 스테이지 변환들을 구현하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다. 또 다른 양태에서, 프로세서는 액세스 제어를 구현하는 것이 하드웨어 인터럽트들에의 HLOS 의 액세스들을 제한하는 것을 재개하는 것, 하드웨어 타이머

들에의 HLOS 의 액세스들을 제한하는 것을 재개하는 것, 및 HLOS 의 I/O 액세스들을 제한하는 것을 재개하는 것 중 적어도 하나를 더 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다.

[0019] 일 양태에서, 프로세서는 제 2 스테이지 변환들을 구현하는 것이 HLOS 가 메모리를 할당하려는 시도를 모니터링하는 것 및 HLOS 가 메모리를 할당하려고 시도할 때, HLOS 에 액세스가능한 물리 어드레스 공간에서의 하나 이상의 물리 어드레스들을 HLOS 에 제공하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다. 또 다른 양태에서, 프로세서는 HLOS 가 메모리를 할당하려고 시도할 때 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있는지 여부를 결정하는 것 및 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있다고 결정될 때 물리 어드레스 공간에서의 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것을 더 포함하는 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다. 여전히, 또 다른 양태에서, 프로세서는 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것이 HLOS 에 액세스가능한 물리 어드레스 공간에서의 물리 어드레스들로부터 샌드박스화된 컴포넌트에 제공될 물리 어드레스들을 제거하는 것 및 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것을 포함하도록 동작들을 수행하는 프로세서 실행가능한 명령들로 구성될 수도 있다.

[0020] 추가 양태들은 하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 수단; 초기화 이후 하이퍼바이저를 디스에이블하는 수단; 샌드박스 세션을 시작하기 위한 보안 모니터로부터의 신호를 모니터링하는 수단; 샌드박스 세션을 시작하기 위한 신호가 수신될 때 하이퍼바이저를 인에이블하는 수단; 및 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 수단을 포함하는 컴퓨팅 디바이스를 포함한다. 또 다른 양태에서, 보안 모니터는 ARM TrustZone® 일 수도 있다. 여전히, 또 다른 양태에서, 하이퍼바이저는 집적 회로 경계 및 칩 경계 중 적어도 하나에 걸쳐 디스에이블되거나 또는 인에이블될 수도 있다. 또 다른 양태에서, 하이퍼바이저를 초기화하는 수단은 HLOS 의 중간 물리 어드레스 공간에서의 각각의 중간 물리 어드레스가 물리 어드레스 공간에서의 대응하는 물리 어드레스와 동일하게 메모리 공간을 할당하도록 HLOS 를 구성하는 수단을 포함할 수도 있다. 또 다른 양태에서, 하이퍼바이저를 초기화하는 수단은 보안 모니터로 하이퍼바이저의 코드 및 데이터를 인증하는 수단을 더 포함할 수도 있다. 여전히, 또 다른 양태에서, 컴퓨팅 디바이스는 하이퍼바이저가 인에이블되는 동안 디지털 신호 프로세서 및 디지털 신호 프로세서에 포함되는 CPU 중 적어도 하나에 액세스불가능하도록 하이퍼바이저의 코드 및 데이터를 구성하는 수단을 포함할 수도 있다.

[0021] 일 양태에서, 하이퍼바이저를 디스에이블하는 수단은 제 2 스테이지 변환을 바이패스하도록 모든 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 구성하는 수단 및 HLOS 에 대한 제 2 스테이지 변환들을 턴오프하는 수단을 포함할 수도 있다. 또 다른 양태에서, 하이퍼바이저를 디스에이블하는 수단은 하드웨어 인터럽트들에의 HLOS 의 액세스들을 제한하는 것을 일시정지하는 수단, 하드웨어 타이머들에의 HLOS 의 액세스들을 제한하는 것을 일시정지하는 수단, 및 HLOS 의 I/O 액세스들을 제한하는 것을 일시정지하는 수단 중 적어도 하나를 더 포함할 수도 있다.

[0022] 일 양태에서, 컴퓨팅 디바이스는 샌드박스 세션이 종료되었는지 여부를 결정하는 수단, 샌드박스 세션이 종료되었다고 결정될 때 샌드박스 세션 해제 프로시저를 수행하는 수단, 및 샌드박스 세션 해제 프로시저를 수행한 후 하이퍼바이저를 디스에이블하는 수단을 더 포함할 수 있다. 또 다른 양태에서, 샌드박스 세션이 종료되었는지 여부를 결정하는 수단은 샌드박스 세션이 종료되었다는 것을 표시하는 또 다른 신호를 수신하는 수단을 포함할 수도 있다. 또 다른 양태에서, 샌드박스 세션 해제 프로시저를 수행하는 수단은 샌드박스화된 컴포넌트에 대한 모든 버퍼들을 해방하는 수단 및 모든 단편화들을 제거하기 위해 제 2 스테이지 변환 페이지 테이블들을 복원하는 수단을 포함할 수도 있다.

[0023] 일 양태에서, 하이퍼바이저를 인에이블하는 수단은 PL0 및 PL1 제 2 스테이지 메모리 관리 유닛들을 인에이블하는 수단, 인터럽트 요청들을 하이퍼바이저 모드에서 취해지도록 구성하는 수단, 및 모든 활성 SMMU 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 수단을 포함할 수도 있다. 또 다른 양태에서, 컴퓨팅 디바이스는 디지털 신호 프로세서와 프로세서간 통신들을 시작하는 수단을 포함할 수도 있다. 여전히, 또 다른 양태에서, 컴퓨팅 디바이스는 SMMU 결함들을 핸들링하는 수단을 포함할 수도 있다.

[0024] 일 양태에서, 액세스 제어를 구현하는 수단은 제 2 스테이지 변환들을 구현하는 수단을 포함할 수도 있다. 또 다른 양태에서, 액세스 제어를 구현하는 수단은 하드웨어 인터럽트들에의 HLOS 의 액세스들을 제한하는 것을 재개하는 수단, 하드웨어 타이머들에의 HLOS 의 액세스들을 제한하는 것을 재개하는 수단, 및 HLOS 의 I/O 액세

스들을 제한하는 것을 재개하는 수단 중 적어도 하나를 포함할 수도 있다. 여전히, 또 다른 양태에서, 제 2 스테이지 변환들을 구현하는 수단은 HLOS 가 메모리를 할당하려는 시도를 모니터링하는 수단 및 HLOS 가 메모리를 할당하려고 시도할 때, HLOS 에 액세스가능한 물리 어드레스 공간에서의 하나 이상의 물리 어드레스들을 HLOS 에 제공하는 수단을 포함할 수도 있다. 또 다른 양태에서, 컴퓨팅 디바이스는 또한 HLOS 가 메모리를 할당하려고 시도할 때 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있는지 여부를 결정하는 수단, 및 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있다고 결정될 때 물리 어드레스 공간에서의 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 수단을 포함할 수도 있다. 여전히, 또 다른 양태에서, 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 수단은 HLOS 에 액세스가능한 물리 어드레스 공간에서의 물리 어드레스들로부터 샌드박스화된 컴포넌트에 제공될 물리 어드레스들을 제거하는 수단, 및 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 수단을 포함할 수도 있다.

[0025] 추가 양태들에서, 비일시적 프로세서 판독가능 저장 매체는 프로세서로 하여금, 컴퓨팅 디바이스 상에서 메모리를 관리하는 동작들을 수행하게 하도록 구성된 프로세서 실행가능한 소프트웨어 명령들을 저장하고 있을 수도 있으며, 상기 동작들은 하이퍼바이저, 보안 모니터, 및 하이-레벨 운영 시스템 (HLOS) 을 초기화하는 것; 초기화 이후 하이퍼바이저를 디스에이블하는 것; 샌드박스 세션을 시작하기 위한 보안 모니터로부터의 신호를 모니터링하는 것; 샌드박스 세션을 시작하기 위한 신호가 수신될 때 하이퍼바이저를 인에이블하는 것; 및 하이퍼바이저가 인에이블되는 동안 액세스 제어를 구현하는 것을 포함한다. 또 다른 양태에서, 보안 모니터는 ARM TrustZone® 일 수도 있다. 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 하이퍼바이저가 집적 회로 경계 및 칩 경계 중 적어도 하나에 걸쳐 디스에이블되거나 또는 인에이블될 수 있게 동작들을 수행하게 하도록 구성될 수도 있다.

[0026] 일 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 하이퍼바이저를 초기화하는 것이 HLOS 의 중간 물리 어드레스 공간에서의 각각의 중간 물리 어드레스가 물리 어드레스 공간에서의 대응하는 물리 어드레스와 동일하게 메모리 공간을 할당하도록 HLOS 를 구성하는 것을 포함하도록 동작들을 수행하게 하도록 구성될 수도 있다. 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 하이퍼바이저를 초기화하는 것이 보안 모니터로 하이퍼바이저의 코드 및 데이터를 인증하는 것을 포함하도록 동작들을 수행하게 하도록 구성될 수도 있다. 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 하이퍼바이저가 인에이블되는 동안 디지털 신호 프로세서 및 디지털 신호 프로세서에 포함되는 CPU 중 적어도 하나에 액세스불가능하도록 하이퍼바이저의 코드 및 데이터를 구성하는 것을 포함하는 동작들을 수행하게 하도록 구성될 수도 있다.

[0027] 일 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 하이퍼바이저를 디스에이블하는 것이 제 2 스테이지 변환을 바이패스하도록 모든 시스템 메모리 관리 유닛들 (SMMU) 컨텍스트 뱅크들을 구성하는 것 및 HLOS 에 대한 제 2 스테이지 변환들을 턴오프하는 것을 포함하도록 동작들을 수행하게 하도록 구성될 수도 있다. 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 하이퍼바이저를 디스에이블하는 것이 하드웨어 인터럽트들에의 HLOS 의 액세스들을 제한하는 것을 일시정지하는 것, 하드웨어 타이머들에의 HLOS 의 액세스들을 제한하는 것을 일시정지하는 것, 및 HLOS 의 I/O 액세스들을 제한하는 것을 일시정지하는 것 중 적어도 하나를 더 포함하도록 동작들을 수행하게 하도록 구성될 수도 있다.

[0028] 일 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 샌드박스 세션이 종료되었는지 여부를 결정하는 것, 샌드박스 세션이 종료되었다고 결정될 때 샌드박스 세션 해제 프로시저를 수행하는 것, 및 샌드박스 세션 해제 프로시저를 수행한 후 하이퍼바이저를 디스에이블하는 것을 포함하는 동작들을 수행하게 하도록 구성될 수도 있다. 일 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 샌드박스 세션이 종료되었는지 여부를 결정하는 것이 샌드박스 세션이 종료되었다는 것을 표시하는 또 다른 신호를 수신하는 것을 포함하도록 동작들을 수행하게 하도록 구성될 수도 있다. 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 샌드박스 세션 해제 프로시저를 수행하는 것이 샌드박스화된 컴포넌트에 대한 모든 버퍼들을 해방하는 것 및 모든 단편화들을 제거하기 위해 제 2 스테이지 변환 페이지 테이블들을 복원하는 것을 포함하도록 동작들을 수행하게 하도록 구성될 수도 있다.

[0029] 일 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 하이퍼바이저를 인에이블하는 것이 PL0 및 PL1 제 2 스테이지 메모리 관리 유닛들을 인에이블하는 것, 인터럽트 요청들을 하이퍼바이저 모드에서 취해지도록 구성하는 것, 및 모든 활성 SMMU 컨텍스트 뱅크들을 제 2 스테이지 변환들 내에 네스트된 제 1 스테이지 변환들에 두기 위해 SMMU 드라이버들을 호출하는 것을 포함하도록 동작들을 수행하게 하도록 구성될

수도 있다. 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 디지털 신호 프로세서와 프로세서간 통신들을 시작하는 것을 포함하는 동작들을 수행하게 하도록 구성될 수도 있다.

또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, SMMU 결함들을 핸들링하는 것을 포함하는 동작들을 수행하게 하도록 구성될 수도 있다.

[0030] 일 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 액세스 제어를 구현하는 것이 제 2 스테이지 변환들을 구현하는 것을 포함하도록 동작들을 수행하게 하도록 구성될 수도 있다. 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 액세스 제어를 구현하는 것이 하드웨어 인터럽트들에의 HLOS 의 액세스들을 제한하는 것을 재개하는 것, 하드웨어 타이머들에의 HLOS 의 액세스들을 제한하는 것을 재개하는 것, 및 HLOS 의 I/O 액세스들을 제한하는 것을 재개하는 것 중 적어도 하나를 더 포함하ㄷ도록 동작들을 수행하게 하도록 구성될 수도 있다. 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 제 2 스테이지 변환들을 구현하는 것이 HLOS 가 메모리를 할당하려는 시도를 모니터링하는 것 및 HLOS 가 메모리를 할당하려고 시도할 때, HLOS 에 액세스가능한 물리 어드레스 공간에서의 하나 이상의 물리 어드레스들을 HLOS 에 제공하는 것을 포함하도록 동작들을 수행하게 하도록 구성될 수도 있다. 여전히, 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, HLOS 가 메모리를 할당하려고 시도할 때 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있는지 여부를 결정하는 것 및 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있다고 결정될 때 물리 어드레스 공간에서의 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것을 포함하는 동작들을 수행하게 하도록 구성될 수도 있다. 또 다른 양태에서, 저장된 프로세서 실행가능한 소프트웨어 명령들은 프로세서로 하여금, 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것이 HLOS 에 액세스가능한 물리 어드레스 공간에서의 물리 어드레스들로부터 샌드박스화된 컴포넌트에 제공될 물리 어드레스들을 제거하는 것 및 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공하는 것을 포함하도록 동작들을 수행하게 하도록 구성될 수도 있다.

도면의 간단한 설명

[0031] 본원에 포함되어 본 명세서의 부분을 구성하는, 첨부 도면들은, 본 발명의 예시적인 양태들을 예시하며, 그리고 위에서 주어진 일반적인 설명 및 아래에 주어지는 상세한 설명과 함께, 본 발명의 특징들을 설명하는 것을 돕는다.

도 1 은 양태 컴퓨팅 디바이스의 컴포넌트 블록도이다.

도 2 는 컴퓨팅 디바이스의 모듈들의 기능 블록 다이어그램이다.

도 3 은 양태 컴퓨팅 시스템의 계층화된 (layered) 컴퓨터 아키텍처 다이어그램이다.

도 4 및 도 5 는 가상 머신들에서 양태 논리적 컴포넌트들의 계층화된 컴퓨터 아키텍처 다이어그램들이다.

도 6 은 시스템 가상 머신을 구현하는 컴퓨팅 디바이스에서 2개의 스테이지 메모리 어드레스 맵핑들을 예시하는 기능 블록 및 메모리 맵 다이어그램이다.

도 7 은 하이퍼바이저가 디스에이블되는 동안 시스템 가상 머신을 구현하는 컴퓨팅 디바이스에서 2개의 스테이지 메모리 어드레스 맵핑들을 예시하는 메모리 맵 다이어그램이다.

도 8 은 샌드박스 세션 동안 하이퍼바이저가 인에이블되는 동안 시스템 가상 머신을 구현하는 컴퓨팅 디바이스에서 2개의 스테이지 메모리 어드레스 맵핑들을 예시하는 메모리 맵 다이어그램이다.

도 9 는 샌드박스 세션 동안 하이퍼바이저가 인에이블되는 동안 공유된 가상 메모리로 시스템 가상 머신을 구현하는 컴퓨팅 디바이스에서 2개의 스테이지 메모리 어드레스 맵핑들을 예시하는 메모리 맵 다이어그램이다.

도 10 은 하이퍼바이저를 선택적으로 인에이블하고 디스에이블하는 양태 방법을 예시하는 프로세스 흐름도이다.

도 11 은 컴퓨팅 디바이스 상에서 공유된 가상 메모리 환경을 개시하는 것에 수반되는 시그널링을 예시하는 콜 흐름도이다.

도 12 는 공유된 가상 메모리 세션을 구성하는 양태 방법을 예시하는 프로세스 흐름도이다.

도 13 은 하이퍼바이저를 디스에이블하는 양태 방법을 예시하는 프로세스 흐름도이다.

도 14 는 컴퓨팅 디바이스 상에서 샌드박스 세션을 개시하는데 수반되는 시그널링을 예시하는 콜 흐름도이다.

도 15 는 하이퍼바이저를 인에이블하는 양태 방법을 예시하는 프로세스 흐름도이다.

도 16a 및 도 16b 는 제 2 스테이지 변환들을 구현하는 양태 방법들을 예시하는 프로세스 흐름도들이다.

도 17 은 샌드박스 세션 해제를 수행하는 양태 방법을 예시하는 프로세스 흐름도이다.

도 18 은 여러 양태들을 구현하는데 적합한 컴퓨팅 디바이스의 컴포넌트 블록도이다.

도 19 는 여러 양태들을 구현하는데 적합한 또 다른 컴퓨팅 디바이스의 컴포넌트 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0032] 여러 양태들이 첨부 도면들을 참조하여 자세히 설명된다. 가능한 경우에는 언제나, 동일한 또는 유사한 부재들을 지칭하기 위해서 동일한 참조 번호들이 도면들 전반에 걸쳐서 사용된다. 특정의 예들 및 구현예들에 대한 언급들은 예시적인 목적들을 위한 것이며, 본 발명 또는 청구항들의 범위에 한정하려고 의도되지 않는다.

[0033] 개관에서, 여러 양태들은 하이퍼바이저와의 액세스 제어를 효율적으로 시행하기 위해 컴퓨팅 디바이스 상에서 하이퍼바이저를 선택적으로 구현하는 컴퓨팅 디바이스 및 방법들을 포함한다. 여러 양태들에서, 하이퍼바이저는 보통은 디스에이블되고, 하이퍼바이저에 액세스 제어 (즉, "샌드박스 세션") 을 구현하도록 요구하는 컨디션이 검출될 때 인에이블된다. 디스에이블되는 동안, 하이퍼바이저는 제 2 스테이지 변환들을 수행하지 못할 수도 있지만, 인에이블되는 동안, 하이퍼바이저는 제 2 스테이지 변환들을 재개할 수도 있다. 일부 양태들에서, 인에이블되는 동안, 하이퍼바이저는 또한 입력/출력 (I/O) 액세스들, 하드웨어 인터럽트 액세스들, 및/또는 하드웨어 타이머 액세스들을 제한하는 것을 포함한, 하이퍼바이저가 디스에이블되는 동안 일시정지되는 다른 활동들을 재개할 수도 있다. 따라서, 단지 샌드박싱이 요구되는 동안에만 하이퍼바이저를 선택적으로 인에이블함으로써, 컴퓨팅 디바이스는 보안 동작 환경을 유지하면서 전체 성능 및 사용자 경험을 향상시킬 수도 있다.

[0034] 여러 양태들에서, 컴퓨팅 디바이스는 메모리, 및 그 메모리에 커플링된, 하드웨어 (즉, 베어 메탈 (bare metal) 하이퍼바이저) 로, 소프트웨어 (즉, 종래의 운영 시스템 환경 내에서 동작하는 호스트된 하이퍼바이저) 로, 또는 하드웨어와 소프트웨어의 조합으로 구현되는 하이퍼바이저로 구성된 프로세서를 포함할 수도 있다. 하이퍼바이저는 추가적으로, 여러 애플리케이션들, 운영 시스템들, 프로세스들, 신호들, 등을 위한 하나 이상의 가상 머신들을 생성하고 관리할 수도 있다.

[0035] 용어 "컴퓨팅 디바이스" 는 본원에서, 셀룰러 전화기들, 스마트폰들, 개인 또는 모바일 멀티-미디어 플레이어들, 개인 휴대 정보단말기들 (PDA's), 랩탑 컴퓨터들, 태블릿 컴퓨터들, 스마트북들, 팜탑 컴퓨터들, 무선 전자 메일 수신기들, 멀티미디어 인터넷 이용가능한 셀룰러 전화기들, 무선 게이밍 제어기들, 및 프로그래밍가능 프로세서 및 메모리를 포함하고 전력 절약 방법들이 유익하도록 배터리 전력 하에서 동작하는 유사한 개인 전자 디바이스들 중 임의의 하나 또는 모두를 지칭하기 위해 사용된다. 여러 양태들은 제한된 프로세싱 전력 및 배터리 용량을 가지는, 셀룰러 전화기들과 같은, 모바일 디바이스들에 특히 유용하지만, 양태들은 향상된 프로세서 성능 및 감소된 에너지 소비로부터 이점을 취할 수도 있는 임의의 컴퓨팅 디바이스에 일반적으로 유용하다.

[0036] 용어들 "가상 머신 모니터", "VMM", 및 "하이퍼바이저" 는 가상 머신 관리기를 지칭하기 위해 본원에서 상호교환가능하게 사용된다. 용어 "하이-레벨 운영 시스템" (HLOS) 은 본원에서, 하이퍼바이저가 관리하는 가상 머신에서 동작하는 게스트 운영 시스템을 지칭하기 위해 사용된다. 일 양태에서, 하이퍼바이저는 샌드박스 세션 동안 HLOS 를 분리할 수도 있다.

[0037] 용어들 "스테이지 1 변환" 및 "제 1 스테이지 변환" 은 본원에서, 가상 메모리 어드레스 ("VA") 로부터 중간 메모리 어드레스 ("IPA") 로의 변환 또는 맵핑을 지칭하기 위해 상호교환가능하게 사용된다. 일 예에서, HLOS 는 HLOS 상에서 동작하는 프로세스에 할당되는 가상 어드레스들로부터, HLOS 의 중간 물리 어드레스 공간에 유지되는 중간 물리 어드레스들로의 제 1 스테이지 변환을 수행할 수도 있다.

[0038] 용어들 "스테이지 2 변환" 및 "제 2 스테이지 변환" 은 본원에서, 중간 물리 어드레스로부터 물리 메모리 어드레스 ("PA") 로의 변환 또는 맵핑을 지칭하기 위해 상호교환가능하게 사용된다. 일 예에서, 하이퍼바이저 또는 시스템 메모리 관리 유닛 ("SMMU") 은 HLOS 에 할당되는 중간 물리 어드레스들로부터, 물리 어드레스 공간에서 하이퍼바이저에 의해 유지되는 물리 어드레스들로의 제 2 스테이지 변환을 수행할 수도 있다.

[0039] 용어 "샌드박스 세션" 은 본원에서, 하이퍼바이저가 2개의 이상의 엔티티들 사이에 액세스 제어를 수행하고 있는 어떤 시간 기간을 지칭하기 위해 사용된다. 일 양태에서, 샌드박스 세션은, 하이퍼바이저가 컴퓨팅 디바이스 (예컨대, DRM (digital rights management) 매체들을 플레이하는 보안 비디오 디바이스) 상에서 별개로 프로세싱되어야 하는 보호된 콘텐츠 (예컨대, DRM 기법들에 의해 보호되는 콘텐츠) 에 대해 경보받을 때 시작할 수도 있으며, 그 분리가 더 이상 필요하지 않을 때, 예컨대, 보호된 콘텐츠가 프로세싱되거나 또는 플레이되는 것을 종료하였으며 서비스 또는 애플리케이션이 릴리즈되었을 때 종료할 수도 있다.

[0040] 용어 "샌드박스화된 컴포넌트" 는 본원에서, 하이퍼바이저가 HLOS 로부터 분리하는 (즉, 샌드박스하는) 컴포넌트, 애플리케이션, 프로세스, 등을 지칭하기 위해 사용된다. 일 양태에서, 하이퍼바이저는 HLOS 에 할당되는 물리 어드레스들과 중첩하지 않는 샌드박스화된 컴포넌트에 대해 물리 메모리 어드레스 공간에서의 물리 어드레스들을 할당할 수도 있다.

[0041] 용어 "공유 엔티티" 는 본원에서 가상 메모리를 HLOS 와 공유하는 컴포넌트, 애플리케이션, 프로세스, 등을 지칭하기 위해 사용된다. 일 양태에서, 공유 엔티티 및 HLOS 는 물리 메모리 어드레스 공간에서의 하나 이상의 물리 어드레스들에의 액세스를 공유할 수도 있다.

[0042] 가상 머신들을 생성하고 관리함으로써, 하이퍼바이저는 운영 시스템들, 애플리케이션들, 프로세스들, 등을 포함한, 여러 동작들 또는 데이터 주변에서 보안 분리 (secured separation) 또는 "샌드박스" 를 생성할 수도 있다. 하이퍼바이저는 샌드박스를 이용하여 여러 특징들 (features) 에의 액세스를 제한함으로써, 동작들 또는 데이터에 보안을 제공할 수도 있다. 예를 들어, HLOS 는 하이퍼바이저가 관리하는 가상 머신 내에서 게스트 운영 시스템으로서 동작할 수도 있으며, 하이퍼바이저는 HLOS 가 비디오 신호를 알아챌 수 없게 (즉, 검출하거나 또는 액세스할 수 없게), HLOS 의 가상 머신의 외부에서 프로세싱된 비디오 신호를 관리할 수도 있다.

[0043] 그러나, 액세스 제어를 시행하기 위해 하이퍼바이저를 이용하는 것과 연관되는 성능 비용이 존재한다. 벤치마크 테스팅은 액세스 제어를 시행하는데 하이퍼바이저를 이용하는 것이 벤치마크에 따라서 대략 5% 내지 30% 의 성능 하락을 초래할 수도 있음을 보였다. 여러 양태들은 샌드박스에 의해 제공되는 데이터 및/또는 소프트웨어 보안에 대한 요구가 존재할 때에만 하이퍼바이저를 구현함으로써 이 성능 불이익 (penalty) 을 극복한다.

[0044] 여러 양태들에서, HLOS 는 하이퍼바이저에 의해 관리되는 가상 머신에서 동작할 수도 있다. HLOS 는 HLOS 상에서 실행하는 여러 프로세스들 또는 애플리케이션들에 가상 어드레스를 할당할 때에 이용하기 위해 중간 물리 어드레스 페이지 테이블을 유지할 수도 있다. HLOS 는 물리 메모리 어드레스 공간으로부터 직접 메모리를 할당함으로써, 중간 물리 어드레스들이 물리 어드레스들과 항상 동일하도록 보장할 수도 있다. 다시 말해서, HLOS 의 중간 물리 어드레스 공간에서의 중간 물리 어드레스들은 물리 어드레스 공간에서의 물리 어드레스들과 항상 동일하다. 중간 물리 어드레스들이 물리 어드레스들과 언제나 동일하게 HLOS 가 메모리를 할당할 수 있도록 보장함으로써, 여러 양태들은 하이퍼바이저가 선택적으로 인에이블 및 디스에이블되는 것을 가능하게 함으로써, 하이퍼바이저의 성능 히트 (hit) 가 단지 샌드박스 세션이 요구될 때 일어나므로, 전체 성능을 향상시킨다.

[0045] 여러 양태들에서, HLOS 는 물리 메모리 어드레스 공간으로부터 직접 메모리를 할당할 수도 있다. 그러나, 하이퍼바이저가 인에이블되는 동안, 하이퍼바이저는 물리 메모리 어드레스 공간에의 HLOS 의 액세스를 제한함으로써, HLOS 로 하여금, 단지 물리 메모리 어드레스 공간의 일부로부터 메모리를 할당가능하게 할 수도 있다.

[0046] 일 양태에서, 컴퓨팅 디바이스 프로세서는 샌드박싱이 요구되는 상황들 (즉, 샌드박스 세션에서 프로세싱되어야 하는 데이터 및/또는 소프트웨어) 을 모니터링할 수도 있다. 일 양태에서, 샌드박스 세션은 별개의 프로세스들, 애플리케이션들, 등 사이에 액세스 제어를 시행하도록 샌드박싱이 요구받는 상황 또는 시간 기간일 수도 있다. 예를 들어, 컴퓨팅 디바이스는 보안 신호 (예컨대, 디지털 저작권 관리의 대상인 비디오 신호) 가 수신되었다는 것 또는 HLOS 로부터 분리되어 유지되어야 하는 제 2 운영 시스템이 개시되고 있다는 것을 검출할 수도 있다. 샌드박스 세션에 대한 요구가 검출될 때, 컴퓨팅 디바이스는 하이퍼바이저를 인에이블할 수도 있다. 인에이블되어진 후, 하이퍼바이저는 제 2 스테이지 변환들을 구현하고 단지 물리 메모리 어드레스 공간의 일부로부터 메모리를 할당하는 것에 HLOS 를 제한함으로써, 샌드박스 세션을 확립할 수도 있다. 하이퍼바이저는, 인에이블되는 동안, 또한 I/O 액세스들, 하드웨어 인터럽트 액세스들, 및 하드웨어 타이머 액세스들 중 하나 이상을 제한하는 것과 같은, 다른 액세스 제어 동작들을 재개할 수도 있다.

[0047] 또 다른 양태에서, 하이퍼바이저는 샌드박스 세션의 끝을 모니터링할 수도 있으며, 예컨대, 하이퍼바이저는 보

안 비디오 신호가 더 이상 수신되고 있지 않은지 여부를 결정할 수도 있다. 샌드박스 세션이 종료될 때, 하이퍼바이저는 세션 해제를 수행할 수도 있다. 세션 해제 프로세스에서, 하이퍼바이저는 샌드박스화된 컴포넌트에 할당된 리소스들을 해방할 수도 있다. 예를 들어, 하이퍼바이저는 디지털 신호 프로세서에 할당된 리소스들을 해방하여 보안 비디오 신호를 프로세싱함으로써, HLOS 로 하여금 전체 물리 메모리 어드레스 공간으로부터 메모리를 할당하게 할 수도 있다. 추가 양태에서, 하이퍼바이저는 다음 샌드박싱 상황이 시작할 때까지 디스에이블될 수도 있다.

[0048] 여전히, 또 다른 양태에서, 하이퍼바이저는 HLOS 및 공유 엔티티에 대한 제 2 스테이지 변환들을 인에이블할 수도 있다. 본 양태에서, 공유 엔티티 및 HLOS 는 가상 메모리를 공유하고 물리 메모리 어드레스 공간에서의 물리 어드레스들에의 액세스를 공유할 수도 있다. 추가 양태에서, 하이퍼바이저는 공유되지 않는 메모리 어드레스들에 대한 공유 엔티티와 HLOS 사이의 액세스 제어를 수행할 수도 있다.

[0049] 여러 양태들은 매우 다양한 단일 및 멀티-프로세서 컴퓨터 아키텍처들 상에서 구현될 수도 있으며, 이의 일 예가 도 1 에 예시되어 있다. 컴퓨팅 디바이스 (100) 는 예시된 디지털 신호 프로세서 (DSP) (102), 모뎀 프로세서 (104), 그래픽스 프로세서 (106), 및 애플리케이션 프로세서 (108) 와 같은, 다수의 이종 프로세서들을 포함할 수도 있다. 컴퓨팅 디바이스 (100) 는 또한 프로세서들 (102 내지 108) 중 하나 이상에 접속된 하나 이상의 벡터 코프로세서들 (110) 을 포함할 수도 있다. 각각의 프로세서 (102 내지 110) 는 하나 이상의 코어들을 포함할 수도 있으며, 각각의 프로세서/코어는 다른 프로세서들/코어들과 독립적으로 동작들을 수행할 수도 있다. 각각의 프로세서 (102 내지 110) 는 또한 메모리 (미예시) 및/또는 메모리 관리 시스템 제어기를 포함할 수도 있다. 일 양태에서, 컴퓨팅 디바이스 (100) 컴포넌트들은 단일 기판 상에 로케이트되거나 및/또는 시스템온칩 (SOC) (125) 으로서 함께 밀접하게 커플링될 수도 있다.

[0050] 컴퓨팅 디바이스 (100) 는 센서 데이터, 아날로그-대-디지털 변환들, 무선 데이터 송신들을 관리하고 다른 특수 동작들을 수행하는, 예컨대, 게임들 및 영화들용 인코딩된 오디오 신호들을 프로세싱하는 아날로그 회로부 및 커스텀 회로부 (114) 를 포함할 수도 있다. 컴퓨팅 디바이스 (100) 는 전압 조정기들, 발진기들, 위상-동기 루프들, 주변장치 브릿지들, 데이터 제어기들, 메모리 제어기들, 시스템 제어기들, 액세스 포트들, 타이머들, 및 프로세서들, 메모리들, 및 컴퓨팅 디바이스 상에서 실행하는 클라이언트들을 지원하기 위해 사용되는 다른 유사한 컴포넌트들과 같은, 시스템 컴포넌트들 및 리소스들 (116) 을 더 포함할 수 있다. 시스템 컴포넌트들/리소스들 (116) 의 각각은 메모리 (미예시) 및/또는 메모리 관리 시스템 제어기를 더 포함할 수 있다.

[0051] 여러 양태들에서, 애플리케이션 프로세서 (108) 는 중앙 프로세싱 유닛 (CPU), CPU 의 컴포넌트, 또는 CPU 에 커플링된 프로세싱 유닛일 수도 있다. 일 양태에서, CPU 는 프로세서들 (102 내지 110), 시스템 컴포넌트들/리소스들 (116) 및/또는 주변장치들의 여러 메모리들로부터 정보를 판독하고 그들에 정보를 기록하도록 구성될 수도 있으며, 이것은 각각의 프로세서들 (102 내지 110), 리소스들 (116), 및/또는 주변장치들의 메모리 관리 시스템 제어기들을 통해서 달성될 수도 있다.

[0052] 컴퓨팅 디바이스 (100) 는 클록 (118) 및 전압 조정기 (120) 와 같은, 컴포넌트들과 리소스들 사이의 통신을 위한 입력/출력 모듈 (미예시) 을 더 포함할 수 있다. 프로세서들 (102 내지 108) 은 하나 이상의 메모리 엘리먼트들 (112), 리소스들 (116), 커스텀 회로부 (114), 및 여러 다른 시스템 컴포넌트들에 상호접속/버스 모듈 (122) 을 통해서 상호접속될 수도 있다.

[0053] 위에서 언급한 바와 같이, 컴퓨팅 디바이스 (100) 는 프로세서들 (102 내지 108) 중 하나 이상에 접속된 하나 이상의 벡터 코프로세서들 (110) 을 포함할 수도 있다. 이러한 벡터 코프로세서들 (110) 은 멀티미디어 및 비디오 스트리밍 애플리케이션들과 같은, 고속 및 병렬 실행을 필요로 하는 애플리케이션들을 프로세싱하는데 특히 유용할 수도 있다. 일 양태에서, 벡터 코프로세서 (110) 는 독립적인 하드웨어 레지스터들, 메모리, 및/또는 실행 하드웨어를 포함하는 단일 명령 다중 데이터 (SIMD) 명령 세트 아키텍처 (ISA) 를 구현할 수도 있다. SIMD 벡터 코프로세서는 컴퓨팅 디바이스 (100) 의 메인 프로세서 (예컨대, 애플리케이션 프로세서 (108), CPU, 등) 의 일부이거나, 또는 그에 밀접하게 커플링될 수도 있다.

[0054] 도 2 는 보안 가상 환경 (즉, 샌드박스) 을 유지하는 것이 가능한 양태 컴퓨팅 디바이스 (206) 의 컴포넌트 다이어그램을 예시한다. 비보안 운영 시스템 (208) (즉, HLOS) 은 하이퍼바이저 (212) 와 통신할 수도 있다. 하이퍼바이저는 물리 메모리 (216) 와 통신할 수도 있다. 일 양태에서, 하이퍼바이저는 비보안 운영 시스템 (208) 과 물리 메모리 (216) 또는 다른 하드웨어 (미도시) 사이에 중재자로서 기능할 수도 있다. 또 다른 양태에서, 하이퍼바이저 (212) 는 물리 메모리 (216) 에서의 물리 어드레스들 (PA) 에의, 비보안 운영 시스템 (208) 에 의해 유지되는 중간 물리 어드레스들 (IPA) 의 맵핑을 용이하게 할 수도 있다.

[0055] 일 양태에서, 하이퍼바이저 (212) 는 또한 보안 모니터 (214) (예컨대, ARM TrustZone®) 와 통신할 수도 있다. 보안 모니터 (214) 는 단지 보안 데이터만이 보안 가상 환경 (210) 에 들어가고 나오는 것을 보장하는 게이트키퍼 (gatekeeper) 로서 기능할 수도 있다. 보안 가상 환경 (210) 은 결국, 보안 네트워크 (204) 와 통신할 수도 있다. 보안 가상 환경 (210) 은 보안 네트워크 (204) 에 데이터를 송신하거나 그로부터 데이터를 수신할 수도 있다. 일 예에서, 보안 가상 환경 (210) 은 보안 네트워크 (204) 로부터 민감한 데이터 (sensitive data) 를 수신할 수도 있는 디지털 신호 프로세서 (즉, DSP) 를 포함할 수도 있다. 이 예에서, 민감한 데이터는 디지털 저작권 관리 제한들 (digital rights management limitations) 에 의해 조정되는 비디오 데이터를 포함하는 신호일 수도 있다. 일 양태에서, 보안 모니터 (214) 는 민감한 데이터가 비보안 운영 시스템 (또는 다른 시스템들 또는 프로세스들) 에 액세스불가능한 물리 메모리 (216) 의 일부에 저장되도록 보장하기 위해 하이퍼바이저 (212) 와 통신할 수도 있다. 추가 양태에서, 이 민감한 데이터는 물리 메모리 (216) 내 암호화된 메모리 (미도시) 에 저장될 수도 있다.

[0056] 도 3 은 전형적인 컴퓨터 시스템에서 논리적 컴포넌트들 및 인터페이스들을 나타내는 프로세서의 계층화된 아키텍처를 예시한다. 예시된 컴퓨터 시스템 아키텍처 (300) 는 하드웨어 컴포넌트들 (322) 및 소프트웨어 컴포넌트들 (320) 양쪽을 포함한다. 소프트웨어 컴포넌트들 (320) 은 운영 시스템 (302), 라이브러리 모듈 (304), 및 하나 이상의 애플리케이션 프로그램들 (A_0 내지 A_n) (306) 을 포함할 수도 있다. 하드웨어 컴포넌트들 (322) 는 주변장치들 (308) (예컨대, 하드웨어 가속기들, 입/출력 디바이스들, 등), 중앙 프로세싱 유닛 (CPU) (310), 중앙 프로세싱 유닛 메모리 관리 유닛 (CPU MMU) (316), 하나 이상의 시스템 메모리 관리 유닛들 (본원에서 "시스템 MMU" 또는 "SMMU") (312), 및 하나 이상의 메모리들 (314) 을 포함할 수도 있다.

[0057] 모바일 컴퓨팅 디바이스들용으로 작성된 애플리케이션 소프트웨어는 "애플리케이션들", "앱들", 또는 애플리케이션 프로그램들 (306) 로서 일반적으로 지칭되는 것인 실행가능 코드로 컴파일될 수도 있다. 각각의 애플리케이션 프로그램 (306) 은 단일 프로세스 또는 스레드일 수도 있거나, 또는 복수의 프로세스들 또는 스레드들을 포함할 수도 있다.

[0058] 애플리케이션 프로그램들 (306) 은 고급 언어 (HLL) 라이브러리 콜들을 라이브러리 모듈 (304) 로 애플리케이션 프로그램 인터페이스 (API) 를 통해서 이슈할 수도 있다. 라이브러리 모듈 (304) 은 운영 시스템 (302) 상에서 애플리케이션 2진 인터페이스 (ABI) 를 통해서 서비스들을 (예컨대, 운영 시스템 콜들을 통해) 호출할 수도 있다. 운영 시스템 (302) 은 하드웨어 (322) 에 의해 구현되는 특정의 동작 코드들 (opcode) 및 네이티브 커맨드들의 리스팅인, 특정의 명령 세트 아키텍처 (ISA) 를 이용하여, 하드웨어 컴포넌트들과 통신할 수도 있다. 이러한 방법으로, 명령 세트 아키텍처는 운영 시스템 (302) 에 의해 보여지는 바와 같이 하드웨어 (322) 를 정의할 수도 있다.

[0059] 운영 시스템 (302) 은 다수의 애플리케이션 프로그램들 (A_0 내지 A_n) (306) 에 걸쳐 물리 메모리를 파티셔닝하는 것을 포함할 수도 있는, 애플리케이션 프로그램들 (306) 간 여러 메모리들 (314) 의 할당 및 사용을 조정하고 제어하는 것을 담당할 수도 있다. 일 양태에서, 운영 시스템 (302) 은 여러 애플리케이션 프로그램들 (A_0 내지 A_n) (306) 에 의해 시스템 메모리의 할당 및 사용을 관리하는 하나 이상의 메모리 관리 시스템들 (예컨대, 가상 메모리 관리기, 등) 을 포함할 수도 있다. 메모리 관리 시스템들은 하나의 프로세스에 의해 사용되는 메모리가 또 다른 프로세스에 의해 이미 사용중인 메모리를 방해하지 않게 보장하도록 기능할 수도 있다.

[0060] 일 양태에서, 운영 시스템 (302) 은 운영 시스템 (302) 으로 하여금 특정의 물리 어드레스가 또 다른 어드레스 (즉, 가상 어드레스) 로 보이게 만들 수 있는 "가상 어드레싱" 동작들을 수행하도록 구성되는 가상 메모리 관리기 (OS VMM) 를 포함할 수도 있다. 가상 어드레싱 동작들은 가상 메모리 어드레스를 애플리케이션 프로그램들 (A_0 내지 A_n) (306) 에 할당하는 것을 포함할 수도 있다. 운영 시스템 (302) 내에 가상 메모리 관리기를 포함시키는 것은 다수의 프로세스들 또는 애플리케이션 프로그램들 (A_0 내지 A_n) (306) 간 시스템 메모리의 조정 및 제어를 간단히 할 수도 있다.

[0061] 위에서 설명된 소프트웨어-기반의 메모리 관리 시스템들 (예컨대, OS VMM, 등) 에 더해서, 시스템은 도 3 에 예시된 중앙 프로세싱 유닛 (CPU) 메모리 관리 유닛 (MMU) (316) 및 시스템 MMU (312) 와 같은, 하나 이상의 하드웨어-기반의 메모리 관리 시스템들을 포함할 수도 있다. CPU MMU (316) 및 시스템 MMU (312) 는 물리 어드레스들로의 가상 어드레스들의 변환, 캐시 제어, 버스 중재, 및 메모리 보호와 같은, 여러 메모리 관련된 동작들을 수행하는 것을 담당하는 하나 이상의 하드웨어 컴포넌트들을 각각 포함할 수도 있다. 일 양태에서, CPU MMU (316) 는 메인 CPU (310) 에 어드레스 변환 서비스들 및 보호 기능들을 제공하는 것을 담당할 수도 있

으며, 시스템 MMU (312) 는 다른 하드웨어 컴포넌트들 (예컨대, 디지털 신호 프로세서, 모뎀 프로세서, 그래픽스 프로세서, 등) 에 어드레스 변환 서비스들 및 보호 기능들을 제공하는 것을 담당할 수도 있다.

[0062] 여러 양태들에서, 메모리 관리 시스템들 (예컨대, 시스템 MMU (312), CPU MMU (316), 등) 중 하나 이상은 (예컨대, 가상 어드레스들을 물리 어드레스들로 변환하는 등) 메모리 어드레스 변환들에 사용될 수도 있는 캐시 메모리인, 변환 룩-어사이드 버퍼 (translation look-aside buffer; TLB) 를 포함할 수도 있다. 일 양태에서, 변환 룩-어사이드 버퍼 (TLB) 는 저장된 정보가 키-값 포맷 (예컨대, 해시 테이블) 으로 편성되는 하드웨어 연관 어레이 메모리일 수도 있는 콘텐츠-어드레스가능한 메모리 (CAM) 일 수도 있다. 키들은 가상 어드레스들일 수도 있으며, 값들은 물리 어드레스들일 수도 있다. 여러 양태들에서, 변환 룩-어사이드 버퍼는 하드웨어-관리되거나, 소프트웨어 관리되거나, 또는 하드웨어와 소프트웨어의 조합에 의해 관리될 수도 있다. 하드웨어-관리되는 변환 룩-어사이드 버퍼에 의해, 변환 룩-어사이드 버퍼 엔트리들의 포맷은 소프트웨어에 비가시적일 수도 있으며, 따라서 상이한 유형들의 중앙 프로세서 유닛들에 대해 상이할 수도 있다.

[0063] 일반적으로, 메모리 어드레스 변환 프로세스의 일부로서, 메모리 관리 시스템 (예컨대, OS VMM, 시스템 MMU (312), CPU MMU (316), 등) 은 변환 룩-어사이드 버퍼에 가상 어드레스를 키로서 전송함으로써 변환 룩-어사이드 버퍼로부터 물리 어드레스를 요청하기 위해 콘텐츠-어드레스가능한 메모리 탐색을 수행할 수도 있다. 가상 어드레스 키가 변환 룩-어사이드 버퍼에 대응하는 물리 어드레스 값을 가지면 (즉, "TLB 히트 (hit)" 가 일어나면), 콘텐츠-어드레스가능한 메모리 탐색은 대응하는 물리 어드레스를 취출하여 리턴할 수도 있다. 요청된 어드레스가 변환 룩-어사이드 버퍼에 없으면 (즉, "TLB 미스 (miss)" 가 일어나면), 메모리 어드레스 변환 프로세스는 다수의 메모리 로케이션들의 콘텐츠를 판독하여 물리 어드레스를 계산함으로써, 페이지 워크 (page walk) (예컨대, 소프트웨어 페이지 워크, 하드웨어 페이지 워크, 등) 를 수행할 수도 있다. 물리 어드레스가 페이지 워크에 의해 결정된 후, 가상 어드레스 대 물리 어드레스 맵핑은 변환 룩-어사이드 버퍼에 저장될 수도 있다.

[0064] 소프트웨어-관리되는 변환 룩-어사이드 버퍼를 포함하는 양태들에서, TLB 미스 (miss) 는 운영 시스템으로 하여금, 페이지 테이블들을 워크 (walk) 하여 소프트웨어에서 변환을 수행하게 할 수도 있다. 하드웨어-관리되는 변환 룩-어사이드 버퍼를 포함하는 양태들에서, 메모리 관리 시스템은 하드웨어 테이블 워크를 수행하여, 유효한 페이지 테이블 엔트리가 규정된 가상 어드레스 키에 대해 존재하는지를 결정할 수도 있다.

[0065] 여러 양태들은 가상화 기법들을 이용하는 메모리 관리 시스템들을 제공한다. 가상화 기술들은 컴퓨팅 리소스들의 추상화 (또는 가상화) 를 가능하게 하며, 이것은 운영 시스템과 하드웨어 사이에 제어 프로그램 (예컨대, 가상 머신 모니터 "VMM" 또는 하이퍼바이저) 을 배치함으로써 달성될 수도 있다. 가상화 기법들은 물리적인 하드웨어 머신과 유사한 애플리케이션 프로그램들을 실행하는 소프트웨어 애플리케이션일 수도 있는 가상 머신 (VM) 에서 일반적으로 구현된다. 가상 머신은 애플리케이션 프로그램들과 실행 하드웨어 사이에 인터페이스를 제공함으로써, 특정의 명령 세트 아키텍처에 타이드된 (tied) 애플리케이션 프로그램들로 하여금, 상이한 명령 세트 아키텍처를 구현하는 하드웨어 상에서 실행가능하게 한다.

[0066] 도 4 및 도 5 는 가상 머신을 구현하는 전형적인 컴퓨터 시스템에서 논리적 컴포넌트들을 예시한다. 가상 머신들은 2개의 일반적인 카테고리들, 즉, 시스템 가상 머신들 및 프로세스 가상 머신들로 분류될 수도 있다. 시스템 가상 머신들은 상이한 프로세스들 또는 애플리케이션들 사이에 하부의 물리적인 하드웨어의 공유를 가능하게 한다. 프로세스 가상 머신들은, 한편, 단일 프로세스 또는 애플리케이션을 지원한다.

[0067] 도 4 는 프로세스 가상 머신 (410) 을 구현하는 컴퓨팅 디바이스 (400) 의 논리 계층들을 예시하는 계층화된 아키텍처 다이어그램이다. 컴퓨터 시스템 (400) 은 하드웨어 (408) 및 애플리케이션 프로세스 모듈 (402), 가상화 모듈 (404), 및 운영 시스템 (406) 을 포함하는 소프트웨어 컴포넌트들을 포함할 수도 있다.

[0068] 도 3 을 참조하여 위에서 설명한 바와 같이, 하드웨어 컴포넌트들은 운영 시스템 (302) 을 통해서 단지 애플리케이션 프로그램들 (306) 에 가시적이며, ABI 및 API 는 애플리케이션 프로그램들 (306) 에 이용가능한 하드웨어 특징들을 효과적으로 정의한다. 가상화 소프트웨어 모듈 (404) 은 ABI/API 레벨에서 논리 연산들을 수행할 수도 있고, 및/또는 애플리케이션 프로세스 (402) 가 그렇지 않으면 하드웨어 컴포넌트들과 (즉, 시스템/라이브러리 콜들을 통해서) 통신하는 방법과 동일한 방법으로 가상화 소프트웨어 모듈 (404) 과 통신하도록 운영 시스템 콜들 또는 라이브러리 콜들을 에뮬레이트할 수도 있다. 이러한 방법으로, 애플리케이션 프로세스 (402) 는 가상화 모듈 (404), 운영 시스템 (406), 및 하드웨어 (408) 의 조합을, 도 4 에 예시된 프로세스 가상 머신 (410) 과 같은 단일 머신으로서 간주한다. 이것은 애플리케이션 소프트웨어가 애플리케이션이 궁극적으로 실행할 컴퓨팅 디바이스들의 실제 아키텍처와 관련될 필요가 없기 때문에 애플리케이션 개발자의 작업을

간단하게 한다.

[0069] 프로세스 가상 머신 (410) 은 단독으로 단일 애플리케이션 프로세스 (402) 를 지원하기 위해 존재하며, 따라서 프로세스 (402) 와 함께 생성되며 프로세스 (402) 가 실행을 종료할 때 종료된다. 가상 머신 (410) 상에서 실행하는 프로세스 (402) 는 "게스트" 로 불리며, 하부의 플랫폼은 "호스트" 로 불린다. 프로세스 가상 머신을 구현하는 가상화 소프트웨어 (404) 는 통상 런타임 소프트웨어 (또는 간단히 "런타임") 로 불린다.

[0070] 도 5 는 시스템 가상 머신 (510) 을 구현하는 컴퓨팅 디바이스 (500) 에서 논리 계층들을 예시하는 계층화된 아키텍처 다이어그램이다. 컴퓨터 시스템은 하드웨어 컴포넌트들 (예컨대, 실행 하드웨어, 메모리, I/O 디바이스들, 등) (508), 및 애플리케이션 프로그램들 모듈 (502), 운영 시스템 (504) 및 가상화 모듈 (506) 을 포함하는 소프트웨어 컴포넌트들을 포함할 수도 있다. 가상화 모듈 (506) 상에서 실행하는 소프트웨어는 "게스트" 소프트웨어로서 지칭되며, 가상화 모듈을 지원하는 하부의 플랫폼은 "호스트" 하드웨어로서 지칭된다.

[0071] 프로세스 가상 머신들과는 달리, 시스템 가상 머신 (510) 은 다수의 운영 시스템들 ("게스트 운영 시스템들" 로 불림) 이 공존할 수 있는 완전한 환경을 제공한다. 이와 유사하게, 호스트 하드웨어 플랫폼은 다수의, 분리된 게스트 운영 시스템 환경들을 동시에 지원하도록 구성될 수도 있다. 동시에 실행하는 운영 시스템들 사이의 분리는 그 시스템에 보안의 레벨을 추가한다. 예를 들어, 하나의 게스트 운영 시스템 상의 보안이 침해되거나, 또는 하나의 게스트 운영 시스템이 고장을 일으키면, 다른 게스트 시스템들 상에서 실행하는 소프트웨어는 침해/고장에 의해 영향을 받지 않는다. 호스트 하드웨어 플랫폼은 또한 애플리케이션 소프트웨어가 애플리케이션이 궁극적으로 실행할 컴퓨팅 디바이스들의 실제 아키텍처와 관련될 필요가 없기 때문에 애플리케이션 개발자의 작업을 간단하게 한다.

[0072] 가상화 소프트웨어 모듈 (506) 은 호스트 하드웨어와 게스트 소프트웨어 사이에 논리적으로 위치될 수도 있다. 가상화 소프트웨어는 실제 하드웨어 (네이티브) 상에서 또는 (호스트된) 운영 시스템의 상부에서 실행할 수도 있으며, "하이퍼바이저" 또는 가상 머신 모니터 (VMM) 로서 일반적으로 지칭된다. 네이티브 구성들에서, 가상화 소프트웨어는 실제 하드웨어 상에서 이용가능한 최고 특권 모드에서 실행하며, 게스트 운영 시스템들은 가상화 소프트웨어가 보통 하드웨어 리소스들에 액세스하거나 또는 그들을 조작하는 모든 게스트 운영 시스템 액션들을 인터셉트하여 에뮬레이트할 수 있도록, 축소된 특권들로 실행한다. 호스트되는 구성들에서, 가상화 소프트웨어는 기존 호스트 운영 시스템 상에서 실행하며, 디바이스 드라이버들 및 다른 하부-레벨 서비스들을 제공하기 위해 호스트 운영 시스템에 의존할 수도 있다. 어느 경우에나, 게스트 운영 시스템들 (예컨대, 운영 시스템 (504)) 의 각각은 그들이 물리적인 하드웨어 (508) 와 통신하는 방법과 동일한 방법으로, 가상화 소프트웨어 모듈 (506) 과 통신하므로, 가상화 모듈 (506) 및 하드웨어 (508) 의 조합을 단일, 가상 머신 (510) 으로서 간주한다. 이것은 각각의 게스트 운영 시스템 (예컨대, 운영 시스템 (504)) 으로 하여금, 하드웨어 (508) 에서의 프로세서들, 주변장치들, I/O, MMU들, 및 메모리들에 배타적으로 액세스한다는 착각 하에 동작가능하게 한다.

[0073] 도 3 을 참조하여 위에서 설명한 바와 같이, 운영 시스템은 다수의 프로세스들에 걸쳐서 물리 메모리를 파티셔닝하는 것을 담당할 수도 있다. 이것은 메모리 어드레스 공간 변환 프로세스를 통해서 달성될 수도 있다. 메모리 어드레스 공간 변환 프로세스에서, 운영 시스템은 가상 어드레스들을 각각의 애플리케이션 프로그램에 할당하고, 그후 프로그램의 실행 전에 가상 어드레스들에 기초하여 물리 어드레스들을 할당한다. 그러나, 가상 머신 상에서 실행하는 게스트 운영 시스템을 포함하는 시스템들에서, 게스트 운영 시스템에 의해 할당된 메모리 어드레스들은 실제 물리 어드레스들이 아니라, 중간 물리 어드레스들이다. 이러한 시스템들에서, 물리 메모리의 실제 할당은 일반적으로 하이퍼바이저에 의해 수행되며, 그 하이퍼바이저는 가상 어드레스들, 중간 물리 어드레스들, 및 물리 어드레스들 사이에 관계들을 유지하도록 요구될 수도 있다.

[0074] 대부분의 프로세서 시스템들은 단지 메모리 어드레스 변환 프로세스의 단일 스테이지를 지원하며, 하이퍼바이저로 하여금 가상 어드레스들, 중간 물리 어드레스들, 및 물리 어드레스들 사이의 관계를 관리하게 하는 것을 필요로 한다. 이것은 일반적으로 그 자신의 변환 테이블들 (쉐도우 변환 테이블들로 불림) 을 유지하는 하이퍼바이저에 의해 달성되며, 그 변환 테이블들은 게스트 운영 시스템의 변환 테이블들의 각각을 해석함으로써 유도될 수도 있다. 이러한 시스템들 상에서, 하이퍼바이저는 게스트 운영 시스템의 변환 테이블들에 대한 모든 변화들이 쉐도우 구조들에 반영되도록 보장할 뿐만 아니라, 액세스 결함들을 적합한 스테이지로 리다이렉트하는 것 및 보호들을 시행한다. 위에서 설명한 바와 같이, 이들 동작들은 하이퍼바이저의 복잡성을 증가시키며, 하이퍼바이저를 실행하거나, 유지하거나, 및/또는 관리하는 것에 상당한 오버헤드들을 추가한다.

[0075] 위에서 설명된 단일 스테이지 프로세서들과는 달리, 일부 프로세서 시스템들 (예컨대, ARM v7-A) 은 메모리 변

환의 양쪽의 스테이지들에 하드웨어 지원을 제공한다. 예를 들어, ARM 프로세서들은 게스트 운영 시스템으로 하여금 제 1 스테이지 (즉, 제 1 스테이지 변환들) 에서 가상 어드레스들을 중간 물리 어드레스들로 변환가능하게 하고, 하드웨어로 하여금 제 2 스테이지 (즉, 제 2 스테이지 변환들) 에서 중간 물리 어드레스들을 물리 어드레스들로 변환가능하게 하는 가상화 확장들을 포함할 수도 있다. 이러한 가상화 확장들은 하이퍼바이저를 실행하거나, 유지하거나, 및/또는 관리하는 것과 연관되는 오버헤드들을 감소시키고, 컴퓨팅 디바이스 성능을 향상시킨다.

[0076] 도 6 은 시스템 가상 머신을 구현하는 컴퓨팅 디바이스 (600) 상에서 2개의 스테이지들에서 메모리를 할당하는 것과 연관되는 예시적인 논리적 컴포넌트들 및 어드레스 변환들을 예시한다. 게스트 운영 시스템 (610) (예컨대, HLOS) 의 메모리 관리 시스템은 가상 어드레스 공간 (602, 604) 을 애플리케이션 프로그램들/프로세스들 (A_0, A_n) 의 각각에 할당할 수도 있다. 예를 들어, 가상 어드레스 공간들 (602, 604) 은 가상 메모리 관리기 (예컨대, 게스트 OS VMM) 에 의해 할당될 수도 있다. 각각의 애플리케이션 프로그램/프로세스 (A_0, A_n) 는 그 자신의 가상 어드레스 공간 (602, 604) 을 할당받을 수도 있으며, 각각의 가상 어드레스 공간 (602, 604) 은 하나 이상의 가상 어드레스들 VA_0 (616), VA_n (618) 을 포함할 수도 있다.

[0077] 도 6 에 예시된 예에서, 메모리 어드레스들은 2개의 스테이지들에서 변환된다. 제 1 스테이지 변환 (612) 에서, 게스트 운영 시스템 (610) 의 가상 메모리 관리기 (게스트 OS VMM) 는 가상 어드레스들 VA_0 (616), VA_n (618) 을 중간 물리 어드레스 공간 (606) 에서의 중간 물리 어드레스들 IPA_0 (626), IPA_n (628) 에 맵핑할 수도 있다. 제 2 스테이지 변환 (614) 에서, 하이퍼바이저 및/또는 가상화 확장들은 중간 물리 어드레스들 IPA_0 (626), IPA_n (628) 을 물리 어드레스 공간 (608) 에서의 물리 어드레스들 PA_0 (636), PA_n (638) 에 맵핑할 수도 있다. 제 1 변환 스테이지 (612) 는 제 2 스테이지 변환 (614) 과 독립적으로 수행될 수도 있으며, 기존 시스템들에서, 제 2 스테이지 변환 (614) 을 수행하는 컴포넌트들은 메모리의 특성들에 기초하여 물리 어드레스들을 할당하지 않을 수도 있다.

[0078] 도 7 은 하이퍼바이저가 디스에이블되는 동안 시스템 가상 머신을 구현하는 컴퓨팅 디바이스 (700) 상에서 2개의 스테이지들에서 메모리를 할당하는 것과 연관되는 예시적인 논리적 컴포넌트들 및 어드레스 변환들을 예시한다. 일 양태에서, 디스에이블되는 동안, 하이퍼바이저는 샌드박스 세션이 시작할 때까지 제 2 스테이지 변환들 (708) 에 참여하지 않을 수도 있다. 예를 들어, 샌드박싱은 오직 게스트 운영 시스템 (예컨대, HLOS) 만이 컴퓨팅 디바이스 상에서 실행하고 있을 때에는 필요하지 않을 수도 있다. 따라서, 여러 양태들에서, 하이퍼바이저는 샌드박싱이 필요하지 않은 것으로 결정될 때 제 2 스테이지 변환들에 참여하지 않음으로써, 좀 더 효율적으로 수행할 수도 있다.

[0079] 일 양태에서, 하이퍼바이저가 디스에이블되는 동안, 제 1 스테이지 변환 (706) 에서, HLOS 는 도 6 을 참조하여 위에서 설명된 바와 같이, 가상 어드레스 공간 (710) 에서의 가상 어드레스들을 HLOS 의 중간 물리 어드레스 공간 (720) 에서의 중간 물리 어드레스들에 맵핑할 수도 있다. 예를 들어, HLOS 는 가상 어드레스들 VA_0 (712) 및 VA_n (714) 을 각각 중간 물리 어드레스들 IPA_0 (722) 및 IPA_n (724) 으로 변환하거나/맵핑할 수도 있다. 또 다른 예에서, HLOS (또는 HLOS 상에서 동작하는 MMU) 는 가상 어드레스 공간 (710) 과 중간 물리 어드레스 공간 (720) 사이의 제 1 스테이지 변환 (706) 을 수행함으로써 애플리케이션들 A_0 내지 A_n 에 의한 사용을 위한 가상 메모리의 블록들을 할당할 수도 있다.

[0080] 추가 양태에서, 디스에이블되는 동안, 하이퍼바이저는 제 2 스테이지 변환 (708) 을 통해 중간 물리 어드레스 공간 (720) 으로부터 물리 어드레스 공간 (730) 으로의 변환들을 수행하지 않을 수도 있다. 이 양태에서, HLOS 는 제 2 스테이지 변환들 (708) 을 바이패스할 수도 있다. 따라서, HLOS 가 제 2 스테이지 변환들 (708) 을 바이패스하도록 허용받기 때문에, HLOS 는 물리 어드레스 공간 (730) 으로부터 직접 메모리를 할당할 수도 있다. HLOS 로 하여금 제 2 스테이지 변환들 (708) 을 바이패스가능하게 함으로써, 하이퍼바이저는 중간 물리 어드레스들이 물리 어드레스들과 동일하도록 보장한다. 따라서, 일 예에서, 중간 물리 어드레스 공간 (720) 에서의 IPA_n (724) 은 물리 어드레스 공간 (730) 에서의 PA_n (734) 과 동등하다. 이와 유사하게, 중간 물리 어드레스 공간 (720) 에서의 IPA_0 (722) 은 물리 어드레스 공간 (730) 에서의 PA_0 (732) 과 동일하다.

[0081] 도 8 은 샌드박스 세션 동안 컴퓨팅 디바이스 (800) 상에서 2개의 스테이지들에서 메모리를 할당하는 것과 연관

되는 예시적인 논리적 컴포넌트들 및 어드레스 변환들을 예시한다. 여러 양태들에서, 하이퍼바이저는 샌드박스 세션의 시작을 검출하는 것에 응답하여, 인에이블될 수도 있다.

[0082] 일 양태에서, 샌드박스 세션은 HLOS 가 보호된 콘텐츠로부터 분리되어야 하는 상황일 수도 있다. 보호된 콘텐츠는 보안 애플리케이션, 컴퓨팅 디바이스 상에서 실행하는 제 2 운영 시스템, 또는 별개로 프로세싱되거나 또는 저장될 필요가 있을 수도 있는 그밖의 다른 것을 포함할 수도 있다. 예를 들어, 디지털 신호 프로세서 (즉, "DSP") 는 프로세싱을 위해 보안 비디오 신호 (즉, 보호된 콘텐츠) 를 수신할 수도 있다. 이 예에서, 보안 비디오 신호는 비디오 신호의 무결성 및/또는 보안을 유지하기 위해 HLOS 와는 별개로 프로세싱될 필요가 있을 수도 있다. 추가 양태에서, 보안 비디오 신호가 별개로 프로세싱될 필요가 있다고 결정하는 것에 응답하여, ARM TrustZone® 과 같은, 보안 모니터는 샌드박스 세션이 시작하였다고 하이퍼바이저에 경보할 수도 있다. 경보를 수신하는 것에 응답하여, 하이퍼바이저는 인에이블될 수도 있으며, 중간 물리 어드레스 공간으로부터 물리 어드레스 공간으로의 제 2 스테이지 변환들을 구현하는 것을 재개할 수도 있다.

[0083] 도 8 에 예시된 바와 같이, 일단 샌드박스 세션이 시작되어 하이퍼바이저가 인에이블되면, 하이퍼바이저는 제 2 스테이지 변환들 (808 및 846) 을 재개할 수도 있다. 일 양태에서, 보호된 콘텐츠는 HLOS 가 도 6 을 참조하여 위에서 설명한 바와 같이 메모리를 할당하는 방법과 유사하게, 제 1 스테이지 변환 (844) 및 제 2 스테이지 변환 (846) 양쪽을 이용하여, 프로세싱될 수도 있다. 예를 들어, 보안 환경 (예컨대, 보안 가상 머신 내에서 동작하는 DSP) 은 도 2 를 참조하여 설명한 바와 같이, 보안 네트워크에의 접속을 통해서 보안 비디오 신호를 수신할 수도 있다. 이 예에서, DSP 는 수신된 보안 비디오 신호를 저장하기 위해, 가상 어드레스 공간 (850) 으로부터의 메모리의 하나 이상의 4kb 블록들 (예컨대, VA_{CP} (852)) 을, DSP 상에서 실행하는 비디오-프로세싱 애플리케이션에 할당할 수도 있다. DSP 는 또한 제 1 스테이지 변환 (844) 을 수행함으로써 가상 어드레스 공간 (850) 에서의 VA_{CP} (852) 로부터, 중간 물리 어드레스 공간 (860) 에서의 중간 물리 어드레스 IPA_{CP} (862) 에의 맵핑을 유지할 수도 있다.

[0084] 샌드박스 세션 동안 (즉, 중간 물리 어드레스 공간 (860) 으로부터의 보안 환경의 메모리 할당들 동안), HLOS 는 또한 메모리 할당들을 수행할 수도 있다. 그러나, 하이퍼바이저가 인에이블되고 제 2 스테이지 변환들 (808) 을 재개하였기 때문에, HLOS 는 전체 물리 어드레스 공간 (830) 으로부터 직접 메모리를 할당하는데 제한받지 않는 능력을 더 이상 가지지 않는다.

[0085] 따라서, 일 양태에서, 하이퍼바이저는 HLOS 에 의한 할당에 이용가능한 물리 어드레스 공간 (830) 에서의 물리 어드레스들을 제한할 수도 있다. 다시 말해서, HLOS 는 물리 어드레스 공간 (830) 에의 직접 메모리 할당들을 여전히 수행하지만, 하이퍼바이저는 물리 어드레스 공간 (830) 의 일부 부분들에 액세스하는 HLOS 의 능력을 제한할 수도 있다. 예를 들어, HLOS 는 가상 메모리들 VA_0 (712) 및 VA_n (714) 을 할당할 수도 있으며, 그 가상 메모리들은 중간 물리 어드레스 공간 (720) 에서 제 1 스테이지 변환 (706) 이후, IPA_0 (722) 및 IPA_n (724) 에 각각 맵핑한다. 또, IPA_0 (722) 및 IPA_n (724) 은 도 7 에 예시된 바와 같이 PA_0 (732) 및 PA_n (734) 에 여전히 맵핑될 수도 있지만, 반면 HLOS 는 하이퍼바이저가 디스에이블되는 동안 (즉, HLOS 가 제 2 스테이지 변환들 (808) 을 바이패스하는 것이 가능한 동안) 도 7 을 참조하여 설명된 바와 같이 전체 물리 어드레스 공간 (830) 으로부터 메모리를 할당할 수 있으며, HLOS 는 하이퍼바이저가 인에이블되는 동안 더 작은 물리 어드레스들의 세트에 액세스할 수도 있다.

[0086] 하이퍼바이저가 인에이블되어 제 2 스테이지 변환들 (808, 846) 을 수행하는 동안, 하이퍼바이저는 HLOS 에 이전에 이용가능했던 물리 어드레스들로부터 샌드박스화된 컴포넌트에 메모리를 할당할 수도 있다. 예를 들어, 하이퍼바이저는 IPA_{CP} (862) 를 현재는 HLOS 에 더 이상 이용불가능한 PA_{CP} (872) 에 맵핑할 수도 있다.

따라서, 인에이블되는 동안, 하이퍼바이저는 예를 들어, 샌드박스화된 엔티티에 메모리를 할당함으로써, 하이퍼바이저가 디스에이블되는 동안 HLOS 에 이용가능한 물리 어드레스들을 "펑처링 (puncturing)" 할 수도 있다. HLOS 의 중간 물리 어드레스 공간 (720) 으로부터 물리 어드레스 공간 (830) 으로 제 2 스테이지 변환 (808) 을 수행할 때에, 하이퍼바이저는 물리 어드레스 공간 (830) 에서의 "펑처링된 (punctured)" 물리 어드레스들을 숨김으로써, HLOS 가 "펑처링된" 물리 어드레스들에 액세스하는 것을 방지할 수도 있다. 따라서, 예를 들어, 하이퍼바이저가 PA_{CP} (872) 를 샌드박스화된 컴포넌트에 할당한 후, HLOS 는 그 물리 어드레스에 더 이상 액세스하지 않을 수도 있다.

[0087] HLOS 로 하여금 스테이지 2 변환 (808) 을 바이패스하게 하는 것을 중지함으로써, 하이퍼바이저는 HLOS

(따라서, 샌드박스화된 컴포넌트) 가 궁극적으로 액세스하는 물리 어드레스들을 다시 관리할 수도 있다. 따라서, 하이퍼바이저는 분리되어야 하는 더 많은 다수의 애플리케이션들, 프로세스들, 운영 시스템들, 등이 존재할 때, 특히, 스테이지 2 변환들을 재개함으로써 (즉, 물리 메모리에의 액세스를 직접 관리함으로써) 샌드박싱을 실시할 수도 있다.

[0088] 도 9 는 샌드박스 세션 및 공유된 가상 메모리 프로세스 동안 컴퓨팅 디바이스 (900) 상에서 2개의 스테이지들에서 메모리를 할당하는 것과 연관되는 예시적인 논리적 컴포넌트들 및 어드레스 변환들을 예시한다. 여러 양태들에서, 하이퍼바이저는 공유 엔티티 (예컨대, DSP) 와의 샌드박스 세션의 시작을 검출하는 것에 응답하여 인에이블될 수도 있으며, HLOS 및 공유 엔티티는 물리 어드레스 공간 (930) 에서의 일부 물리 어드레스들을 공유할 수도 있다.

[0089] 도 11 및 도 12 를 참조하여 아래에서 설명되는 양태에서, 컴퓨팅 디바이스는 예컨대, HLOS 와 디지털 신호 프로세서 (DSP) 사이에, 공유된 가상 메모리 세션을, 개시할 수도 있다. 추가 양태에서, HLOS 와 DSP 사이의 공유된 가상 메모리 세션은 물리 어드레스 공간 (930) 에서의 물리 어드레스들의 세트에의 액세스를 공유하도록 HLOS 및 DSP 를 구성하는 것을 포함할 수도 있다. 예를 들어, HLOS 및 DSP 는 HLOS 및 DSP 가 데이터 구조들, 루틴들, 등을 공유해야 할 때 공유된 가상 메모리 세션을 겪을 수도 있다. 직접 메모리 액세스들을 공유함으로써, HLOS 및 DSP 는 물리 어드레스 공간 (930) 에 저장된 정보를 복사하고 송신할 필요 없이, 정보를 효율적으로 공유할 수도 있다.

[0090] 도 9 에 예시된 바와 같이, HLOS 및 DSP 는 각각의 가상 어드레스 공간들 (910, 950) 에서의 가상 메모리들 (즉, VA_{SVM1} (914) 및 VA_{SVMn} (912)) 을 HLOS 및 DSP 상에서 동작하는 애플리케이션들, 프로세스들, 등에 각각 할당하였을 수도 있다. 예를 들어, HLOS 및 DSP 상에서 동작하는 애플리케이션들은 어떤 데이터 구조들, 기능들, 또는 라이브러리들을 공유할 수도 있다. HLOS 및 DSP 는 제 1 스테이지 변환들 (906, 944) 을 각각 수행하여, 가상 어드레스들 VA_{SVM1} (914) 및 VA_{SVMn} (912) 을 공유 엔티티의 중간 물리 어드레스 공간 (960) 및 HLOS 의 중간 물리 어드레스 공간 (920) 의 각각에서의 IPA_{SVM1} (924) 및 IPA_{SVMn} (922) 에 맵핑할 수도 있다.

[0091] 또 다른 양태에서, 하이퍼바이저가 샌드박스 세션의 시작에 응답하여 인에이블되기 때문에, 하이퍼바이저는 HLOS 의 중간 물리 어드레스 공간 (920) 으로부터의 중간 물리 어드레스들을 물리 어드레스 공간 (930) 에 맵핑하는 제 2 스테이지 변환들 (908) 을 활성화할 수도 있다. 하이퍼바이저는 또한 공유 엔티티의 중간 물리 어드레스 공간 (960) 에서의 중간 물리 어드레스들을 물리 어드레스 공간 (930) 에 맵핑하는 제 2 스테이지 변환들 (946) 을 활성화할 수도 있다.

[0092] 일 양태에서, 도 8 을 참조하여 설명되는 바와 같이, HLOS 의 중간 물리 어드레스 공간 (920) 및 물리 어드레스 공간 (930) 으로부터의 제 2 스테이지 변환들 (908) 을 인에이블함으로써, 하이퍼바이저는 HLOS 가 액세스하는 물리 어드레스 공간 (930) 에서의 물리 어드레스들을 제한할 수도 있다 (즉, HLOS 가 물리 어드레스 공간 (930) 에서의 그들 물리 어드레스들에 액세스하지 않도록 일부 물리 어드레스들에의 맵핑들을 제거할 수도 있다). 도 9 에 예시된 바와 같이, 하이퍼바이저는, 중간 물리 어드레스들이 물리 어드레스들과 동일하게 HLOS 의 중간 물리 어드레스 공간 (920) 으로부터 물리 어드레스 공간 (930) 으로의 맵핑들 (940) 이 보장하도록, IPA_{HLOS} (926), IPA_{SVM1} (924), 및 IPA_{SVMn} (922) 으로부터 각각 물리 어드레스들 PA_{HLOS} (936), PA_{SVM1} (934), 및 PA_{SVMn} (932) 으로의 맵핑들 (940) 을 유지할 수도 있다. 이와 유사하게, 하이퍼바이저는 다른 맵핑들 중에서도 특히, 공유 엔티티의 중간 물리 어드레스 공간 (960) 에서의 IPA_{SVM1} (924) 및 IPA_{SVMn} (922) 으로부터 물리 어드레스 공간 (930) 에서의 각각 PA_{SVM1} (934), 및 PA_{SVMn} (932) 에의 공유된 맵핑들 (941) 을 유지할 수도 있다.

[0093] 또 다른 양태에서, 하이퍼바이저는 공유 엔티티의 "부분" 샌드박싱을 구현할 수도 있다. 본 양태에서, 하이퍼바이저는 HLOS 로부터, HLOS 와 공유되지 않는 공유 엔티티에 할당된 물리 어드레스들 (예컨대, $PA_{non-SVM}$ (962)) 로의 맵핑들을 제거할 수도 있다. 공유 엔티티 및 HLOS 는 공유되지 않는 물리 어드레스 공간에서 메모리에의 맵핑들을 각각 유지할 수도 있다. 예를 들어, DSP 는 예를 들어, DSP 의 커널일 수도 있는 $IPA_{non-SVM}$ (923) 과 연관되는 정보를 유지할 수도 있다. 또 다른 예에서, HLOS 는 공유 엔티티와 공유되지 않는 PA_{HLOS} (936) 에 맵핑되는 HLOS 의 중간 물리 어드레스 공간 (920) 에서의 IPA_{HLOS} (926) 에서 메모리를 유지할 수도 있다.

[0094] 그러나, 하이퍼바이저는 HLOS 로부터, 공유 엔티티에 할당되지만 HLOS 와 공유되는 물리 어드레스들 (예컨대,

IPA_{SVM1} (924), 및 IPA_{SVMn} (922)) 로의 맵핑들을 제거하지 않을 수도 있다. 맵핑들을 제거하지 않음으로써, 하이퍼바이저는 HLOS 및 공유 엔티티로 하여금, 데이터 구조들, 라이브러리들, 루틴들, 등에 대한 포인터들과 같은, 이들 물리 어드레스들에 저장된 정보를 공유가능하게 할 수도 있다.

[0095] 따라서, HLOS 의 중간 물리 어드레스 공간 (920) 으로부터, 제 2 스테이지 변환들 (908) 로부터 제거되는 물리 어드레스 공간 (930) 에의 맵핑들을 관리함으로써, 하이퍼바이저는 HLOS 및 공유 엔티티로 하여금, 어떤 물리 어드레스들에 저장된 정보를 공유가능하게 할 수도 있으며, 공유되지 않는 어드레스들 (예컨대, $PA_{non-SVM}$ (962) 및 PA_{HLOS} (936)) 의 액세스 제어를 여전히 시행할 수도 있다.

[0096] 도 10 은 샌드박스 세션 동안 하이퍼바이저를 선택적으로 인에이블하기 위해 컴퓨팅 디바이스 프로세서 (예컨대, CPU) 에서 구현될 수도 있는 양태 방법 (1000) 을 예시한다. 블록 (1002) 에서, 컴퓨팅 디바이스 프로세서는 하이퍼바이저, 보안 모니터, 및 HLOS 를 초기화할 수도 있다. 일 양태에서, 컴퓨팅 디바이스 프로세서는 Linaro ARMv8 보안 부트 플로우 (secure boot flow) 및 Xen-스타일 승인 테이블 (grant table) 을 이용하여, 하이퍼바이저, 보안 모니터, 및 HLOS 에 부팅함으로써, 하이퍼바이저, 보안 모니터, 및 HLOS 를 초기화할 수도 있다. 또 다른 양태에서, 보안 모니터는 ARM TrustZone® 일 수도 있다.

[0097] 추가 양태에서, 하이퍼바이저의 코드 및 데이터는 초기화 동안 보안 모니터에 의해 인증되거나 및/또는 서명될 수도 있다. 초기화 동안, 하이퍼바이저는 또한 그 코드 및 데이터가 디지털 신호 프로세서 (DSP) 또는 DSP 에 포함되는 CPU 와 같은, 외부 프로세서들에 액세스불가능하도록, 구성될 수도 있다. 또 다른 양태에서, 하이퍼바이저를 인증하는 것 및/또는 하이퍼바이저가 인에이블되는 동안 외부 프로세서들이 하이퍼바이저의 코드 및 데이터에 액세스하는 것을 방지하는 것은 미래 샌드박스 세션들이 안전하도록 보장할 수도 있다.

[0098] 옵션적인 결정 블록 (1004) 에서, 컴퓨팅 디바이스 프로세서는 공유된 가상 메모리와의 동시발생 이종적 계산 세션 (concurrent heterogeneous compute session) 이 존재하는지 여부를 결정할 수도 있다. 일 양태에서, HLOS 및 공유 엔티티 (예컨대, DSP) 는 복잡한, 포인터-포함 데이터 구조들을 공유할 수도 있다. 공유된 가상 메모리와의 동시발생 이종적 계산 세션 상황이 존재한다고 컴퓨팅 디바이스 프로세서가 결정하면 (즉, 결정 블록 1004 = "예"), 블록 (1006) 에서 프로세서는 공유된 가상 메모리와의 동시발생 이종적 계산 세션을 셋업할 수도 있다. 일 양태에서, HLOS 및 DSP 는, 예를 들어, 동일한 제 1 스테이지 페이지 테이블을 공유하도록 구성될 수도 있다. 공유된 가상 메모리와의 동시발생 이종적 계산 세션을 셋업하는 것은 도 11 및 도 12 를 참조하여 아래에서 자세히 설명된다. 컴퓨팅 디바이스 프로세서는 블록 (1008) 에서 계속 동작할 수도 있다. 어떤 공유된 가상 메모리와의 동시발생 이종적 계산 세션도 없다고 컴퓨팅 디바이스 프로세서가 결정하면 (즉, 옵션적인 결정 블록 1004 = "아니오"), 프로세서는 또한 블록 (1008) 에서 계속 동작할 수도 있다.

[0099] 블록 (1008) 에서, 컴퓨팅 디바이스 프로세서는 하이퍼바이저를 디스에이블할 수도 있다. 일 양태에서, 하이퍼바이저의 디폴트 컨디션은 디스에이블될 수도 있다. 또 다른 양태에서, 하이퍼바이저를 디스에이블하는 것은 중간 물리 어드레스 공간들로부터 물리 메모리 어드레스 공간으로의 제 2 스테이지 변환들을 디스에이블할 수도 있다. 하이퍼바이저를 디스에이블하는 다른 결과들은 하드웨어 인터럽트들, 하드웨어 타이머들, 및 입/출력에의 HLOS 의 액세스들을 제한하는 것을 중지하는 것을 포함할 수도 있다. 하이퍼바이저를 디스에이블하는 것은 도 13 을 참조하여 아래에서 추가로 설명된다.

[0100] 블록 (1009) 에서, 컴퓨팅 디바이스 프로세서는 샌드박스 세션을 시작하기 위한 하이퍼바이저 상에서 수신된 신호를 모니터링할 수도 있다. 일 양태에서, 보안 모니터 (예컨대, ARM TrustZone®) 는 보호된 콘텐츠를 수신하거나 또는 검출하고 대기해제 (wake-up) 신호를 하이퍼바이저로 전송하여, 샌드박스 세션을 시작할 수도 있다. 예를 들어, 보안 가상 환경 내에서 동작하는 DSP 는 보안 프로세싱을 위해 보안 비디오 신호를 수신할 수도 있다. 이 예에서, DSP 는 비디오 신호를 예를 들어, HLOS 에 액세스불가능한 물리 메모리 어드레스 공간의 일부에 저장하도록 구성될 수도 있다.

[0101] 결정 블록 (1010) 에서, 컴퓨팅 디바이스 프로세서는 하이퍼바이저가 샌드박스 세션을 시작하기 위한 신호를 수신하였는지 여부를 결정할 수도 있다. 하이퍼바이저가 샌드박스 세션을 시작하기 위한 신호를 수신하지 않았다고 컴퓨팅 디바이스 프로세서가 결정하면 (즉, 결정 블록 1010 = "아니오"), 프로세서는 블록 (1009) 에서 계속 동작할 수도 있다. 일 양태에서, 컴퓨팅 디바이스 프로세서는 샌드박스 세션을 시작하기 위한 하이퍼바이저에 대한 신호를 계속 모니터링할 수도 있다.

[0102] 하이퍼바이저가 샌드박스 세션을 시작하기 위한 신호를 수신하였다고 컴퓨팅 디바이스 프로세서가 결정하면 (즉, 결정 블록 1010 = "예"), 프로세서는 블록 (1012) 에서 하이퍼바이저를 인에이블할 수도 있다. 일 양

태에서, 하이퍼바이저를 인에이블하는 것은 제 2 스테이지 변환들을 재개하는 것을 포함할 수도 있다. 하이퍼바이저를 인에이블하는 것은 도 14 및 도 15 를 참조하여 아래에서 좀더 자세히 설명된다.

[0103] 하이퍼바이저는 그후 블록 (1014) 에서 액세스 제어를 구현할 수도 있다. 일 양태에서, 하이퍼바이저는 중간 물리 어드레스들로부터 물리 어드레스들로의 제 2 스테이지 변환들을 수행함으로써 액세스 제어를 구현할 수도 있다. 추가 양태에서, 하이퍼바이저는 I/O, 하드웨어 인터럽트들, 및 하드웨어 타이머들에 대한 액세스들을 제한하는 것을 재개함으로써 액세스 제어를 추가로 구현할 수도 있다. 액세스 제어를 구현하는 프로세스는 도 16a 및 도 16b 를 참조하여 아래에서 좀더 자세히 설명된다.

[0104] 결정 블록 (1016) 에서, 하이퍼바이저는 샌드박스 세션이 종료되었는지 여부를 결정할 수도 있다. 예를 들어, 샌드박스 세션은 HLOS 또는 다른 프로세스들, 애플리케이션들, 또는 컴포넌트들로부터 보호되거나 또는 분리되어야 하는 어떤 콘텐츠도 존재하지 않을 때 종료될 수도 있다. 예를 들어, 위에서 주어진 예에서, DSP 가 보안 비디오 신호를 수신했을 때 개시되는 샌드박스 세션은, DSP 가 보안 비디오 신호를 프로세싱하여 더 이상 비디오 신호의 비디오 버퍼들을 물리 메모리에 저장할 필요가 없어진 이후에 종료될 수도 있다. 또 다른 양태에서, 보안 모니터 또는 보안 가상 환경에서의 또 다른 컴포넌트는 샌드박스 세션이 종료되었음을 하이퍼바이저로 시그널링할 수도 있다.

[0105] 샌드박스 세션이 종료되지 않았으면 (즉, 결정 블록 1016 = "아니오"), 하이퍼바이저는 블록 (1014) 에서 동작들을 계속 수행할 수도 있다. 그렇지 않으면 (즉, 결정 블록 1016 = "예"), 하이퍼바이저는 블록 (1018) 에서 샌드박스 세션을 해제할 수도 있다. 일 양태에서, 하이퍼바이저는 샌드박스 세션 해제 프로시저를 수행하는 것의 결과로서 컴퓨팅 디바이스의 HLOS 및 여러 다른 컴포넌트들을 "디폴트" 상태 또는 구성으로 리턴할 수도 있다. 샌드박싱 세션 해제들은 도 17 을 참조하여 아래에서 좀더 자세히 설명된다. 컴퓨팅 디바이스는 블록 (1008) 에서 동작들을 계속 수행할 수도 있다.

[0106] 도 11 은 동시발생 이종적 계산 세션을 개시하는 컴퓨팅 디바이스의 여러 컴포넌트들 간 양태 신호 및 콜 흐름을 예시한다. 일 양태에서, 동작 (1112) 에서, HLOS (1102) 는 제 1 스테이지 변환 테이블을 생성할 수도 있다. HLOS (1102) 는 또한 동작 (1114) 에서 HLOS 의 가상 머신 식별자 (즉, "HLOS_VMID") 에 애플리케이션 특정 식별자 (ASID) 를 할당할 수도 있다. 또 다른 양태에서, 하이퍼바이저 (1104) 는 HLOS 의 HLOS_VMID 를 이용하여 제 2 스테이지 변환 테이블을 생성하기 위해 신호 (1116) 를 디지털 신호 프로세서 (DSP) 제 2 스테이지 시스템 메모리 관리 유닛 (SMMU) (1106) 으로 전송할 수도 있다. 일 양태에서, DSP 제 2 스테이지 SMMU 는 제 2 스테이지 변환 테이블을 이용하여, HLOS 의 중간 물리 어드레스 공간으로부터 물리 어드레스 공간으로의 제 2 스테이지 변환들을 수행할 수도 있다.

[0107] 일 양태에서, HLOS 는 HLOS 의 HLOS_VMID, 선택된 애플리케이션 특정 식별자, 및 제 1 스테이지 변환 테이블을 이용하여 공유된 가상 메모리 (SVM) 프로세스의 생성을 요청하는 신호 (1118) 를 DSP 의 하이퍼바이저 (1108) 로 전송할 수도 있다. DSP 의 하이퍼바이저 (1108) 은 DSP 의 메모리 관리 유닛 (MMU) 에 대한 제 1 스테이지 변환 테이블을 프로그래밍한 후 DSP 의 공유된 가상 메모리 프로세스 (1110) 를 개시하는 신호 (1120) 로, DSP 의 공유된 가상 메모리 프로세스 (1110) 를 개시할 수도 있다.

[0108] 도 12 는 HLOS 와 DSP 사이의 공유된 가상 메모리 프로세스를 개시하기 위해 컴퓨팅 디바이스에서 구현될 수도 있는 양태 방법 (1006a) 을 예시한다. 컴퓨팅 디바이스 프로세서는 프로세서가 공유된 가상 메모리와의 동시발생 이종적 계산 세션이 있다고 결정할 때 (즉, 결정 블록 1004 = "예") 방법 (1006a) 을 시작할 수도 있다.

블록 (1204) 에서, 컴퓨팅 디바이스는 제 1 스테이지 변환 테이블을 생성하도록 HLOS 를 구성할 수도 있다.

일 양태에서, HLOS 는 제 1 스테이지 변환 테이블을 이용하여, 가상 어드레스를 중간 물리 어드레스들에 맵핑할 수도 있다. 컴퓨팅 디바이스는 또한 블록 (1206) 에서 HLOS 의 가상 머신 식별자 (즉, HLOS_VMID) 에 애플리케이션 특정 식별자를 할당하도록 HLOS 를 구성할 수도 있다.

[0109] 블록 (1208) 에서, 컴퓨팅 디바이스는 HLOS_VMID 에 기초하여 DSP 제 2 스테이지 SMMU 로 제 2 스테이지 구성을 전송하도록 하이퍼바이저를 구성할 수도 있다. 일 양태에서, DSP 제 2 스테이지 SMMU 는 HLOS_VMID 에 기초하여 HLOS 에 대한 제 2 스테이지 변환 테이블을 생성할 수도 있다. SMMU (또는 하이퍼바이저) 는 제 2 스테이지 변환 테이블을 이용하여, HLOS 의 중간 물리 어드레스 공간으로부터 컴퓨팅 디바이스의 물리 메모리 어드레스 공간으로의 제 2 스테이지 변환들을 수행할 수도 있다.

[0110] 블록 (1210) 에서, 컴퓨팅 디바이스는 DSP 의 하이퍼바이저가 HLOS_VMID, 선택된 ASID, 및 HLOS 제 1 스테이지 변환 테이블을 이용하여 공유된 가상 메모리 프로세스를 생성하는 것을 요청하도록 HLOS 를 구성할 수도 있다.

[0111] 컴퓨팅 디바이스는 또한 블록 (1212) 에서 제 1 스테이지 메모리 관리 유닛 (MMU) 을 프로그래밍하고 공유 가상 메모리 프로세스를 개시하도록 DSP 의 하이퍼바이저를 구성할 수도 있다. 일 양태에서, DSP 의 제 1 스테이지 MMU 는 HLOS 의 제 1 스테이지 변환들 테이블과 동일한 제 1 스테이지 변환 테이블을 개시할 수도 있다. 따라서, 이 양태에서, HLOS 및 DSP 는 그들이 동일한 제 1 스테이지 변환 테이블을 공유하기 때문에 가상 메모리를 공유할 수도 있다.

[0112] 공유된 가상 메모리 프로세스들이 블록 (1212) 에서 완료되면, 컴퓨팅 디바이스 프로세서는 블록 (1008) 에서, 샌드박스 세션에 대해 어떤 요구도 존재하지 않을 때 도 10 을 참조하여 위에서 설명한 바와 같이 하이퍼바이저를 디스에이블할 수도 있다.

[0113] 도 13 은 컴퓨팅 디바이스 상에서 하이퍼바이저를 디스에이블하는 양태 방법 (1008a) 을 예시한다.

[0114] 블록 (1304) 에서, 하이퍼바이저는 제 2 스테이지 변환들을 바이패스하도록 모든 SMMU 제 1 스테이지 컨텍스트 뱅크들을 구성할 수도 있다. 하이퍼바이저는 또한 블록 (1306) 에서 HLOS 의 제 2 스테이지 변환들을 턴오프할 수도 있다.

[0115] 일부 양태들에서, 하이퍼바이저는 디스에이블될 때 여러 다른 활동들을 일시정지할 수도 있다. 예를 들어, 하이퍼바이저는 옵션적으로, 옵션적인 블록 (1308) 에서 I/O 액세스들을 제한하는 것을 일시정지할 수도 있다. 하이퍼바이저는 또한 옵션적인 블록 (1310) 에서 하드웨어 인터럽트 액세스들을 제한하는 것을 일시정지할 수도 있다. 게다가, 옵션적인 블록 (1312) 에서, 하이퍼바이저는 하드웨어 타이머 액세스들을 제한하는 것을 일시정지할 수도 있다.

[0116] 하이퍼바이저는 결정 블록 (1010) 에서 하이퍼바이저가 도 10 을 참조하여 위에서 설명한 바와 같이 샌드박스 세션을 시작하기 위한 신호를 수신하였는지 여부를 결정할 수도 있다.

[0117] 일부 양태들에서, 여러 하이퍼바이저 기능들 (예컨대, 액세스 제어, 샌드박싱 메모리, 등) 은 집적 회로 경계 및/또는 칩 경계를 가로질러 디스에이블될 수도 있다. 일 양태에서, 별개의 칩세트들 (예컨대, 모뎀 칩 및 애플리케이션 프로세서 칩) 을 포함하는 융합-형 칩세트 조합에서, 마스터 칩세트 (즉, 마스터 하이퍼바이저) 는 하이퍼바이저 기능성이 디스에이블되었을 때 다른 칩세트들에서 샌드박싱 메모리 또는 다른 액세스 제어들을 일시정지할 수도 있다. 예를 들어, 애플리케이션 프로세서 칩세트에서 마스터 하이퍼바이저는 모뎀 또는 DSP 칩세트에서 중간 물리 어드레스들로부터 물리 어드레스들로의 제 2 스테이지 변환들을 일시정지할 수도 있다. 따라서, 하이퍼바이저 기능들이 디스에이블될 때, 마스터 하이퍼바이저는 다수의 별개의 칩들에서 이들 기능들을 일시정지할 수도 있다.

[0118] 도 14 는 콘텐츠-보호되는 비디오 신호에 대한 샌드박스 세션을 셋업하는 동안 컴퓨팅 디바이스 상에서 동작하는 다수의 컴포넌트들 사이의 콜 흐름들 (1400) 을 예시한다. 여러 양태들에서, 비디오 버퍼에 대한 페이지들은 4kb 페이지들일 수도 있으며, HLOS 의 제 2 스테이지 변환 페이지 테이블들을 단편화할 수도 있다.

[0119] 일 양태에서, 신호 (1402) 는 안드로이드 MM 프레임워크 (1450), OpenMax (OMX) 컴포넌트 (1452), V4L2 비디오 드라이버 (1454), 커널 페이지 할당기 (1456), 하이퍼바이저 (1458), SMMU (1460), 및 코어 (1462) 와 같은, 여러 컴포넌트들로, 보안 인증으로 펌웨어를 초기화하기 위해, 전송될 수도 있다. 일 양태에서, 보안 인증은 ARM TrustZone® 일 수도 있다. 또 다른 양태에서, 비디오가 DSP 상에 있으면, 컴퓨팅 디바이스는 DSP 비디오 애플리케이션 및 코덱을 로드할 수도 있다.

[0120] 일 양태에서, 안드로이드 MM 프레임워크 (1450) 는 OMX 컴포넌트를 초기화하기 위해 신호 (1404) 를 OMX 컴포넌트로 전송할 수도 있다. OMX 컴포넌트 (1452) 는 V4L2 비디오 드라이버 (1454) 상의 코덱을 설정하기 위해 신호 (1406) 를 전송할 수도 있다. OMX 컴포넌트 (1452) 는 또한 버퍼 사이징 (sizing) 쿼리 신호 (1408) 를 V4L2 비디오 드라이버 (1454) 로 전송할 수도 있다. 안드로이드 MM 프레임워크 (1450) 는 또한 휴지 신호 (1410) 를 OMX 컴포넌트 (1452) 로 전송할 수도 있다.

[0121] 또 다른 양태에서, OMX 컴포넌트 (1452) 는 스트림 온 (ON) 입력 신호 (1412) 를 V4L2 비디오 드라이버 (1454) 로 전송할 수도 있다. V4L2 비디오 드라이버 (1454) 는 호스트 펌웨어 인터페이스 (HFI) 세션 초기화 신호 (1414) 를 코어 (1462) 로 전송할 수도 있다. 코어 (1462) 는 "HFI 세션 완료 (done)" 신호 (1416) 를 V4L2 비디오 드라이버 (1454) 로 전송함으로써 HFI 세션 초기화 신호 (1414) 에 응답할 수도 있다.

[0122] 안드로이드 MM 프레임워크는 메모리 할당 신호 (1418) (즉, "loctl ION_IOC_ALLOC") 를 OMX 컴포넌트 (1452) 로 전송할 수도 있다. 또 다른 양태에서, 안드로이드 MM 프레임워크는 페이지 풀 (page pool) 을 관리하기

위해 메모리 할당 신호 (1418) 를 전송할 수도 있다. OMX V4L2 비디오 드라이버 (1454) 는 "Ion_alloc(ION cp heap)" 신호 (1420) 를 커널 페이지 할당기 (1456) 에 전송할 수도 있다.

[0123] 커널 페이지 할당기 (1456) 는 그후 프로세서 제 2 스테이지 맵핑들을 제거하기 위한 VMM_CALL 신호 (1422) 를 하이퍼바이저 (1458) 로 전송할 수도 있다. 일 양태에서, 이 신호 (1422) 는 샌드박스 세션이 시작하였으며 그리고 어떤 물리 메모리 로케이션들이 HLOS 에 액세스가능한 그들 물리 어드레스로부터 제거되었음을 하이퍼바이저에 통지할 수도 있다. 추가 양태에서, 하이퍼바이저 (1458) 는 제 2-스테이지-변환-맵핑 신호 (1424) 를 SMMU (1460) 로 전송할 수도 있다. 일 양태에서, SMMU 는 이들 제 2 스테이지 변환들을 구현할 수도 있다 (즉, SMMU 는 제 2 스테이지 변환들을 재개할 수도 있다). 추가 양태에서, SMMU 는 HLOS 로부터 액세스가능한 물리 어드레스들로의 맵핑들 (즉, HLOS 로부터 숨겨져 있지 않은 물리 어드레스들로의 맵핑들) 을 유지할 수도 있다. 추가 양태에서, 비디오가 DSP 상에 있을 때, 커널 페이지 할당기 (1456) 는 하이퍼바이저 (1458) 에 페이지들을 DSP 제 2 스테이지 변환 맵핑에 맵핑하도록 추가적으로 시그널링할 수도 있다.

[0124] 또 다른 양태에서, V4L2 비디오 드라이버 (1454) 는 호스트 펌웨어 인터페이스 세트 버퍼들 (1426) 을 코어 (1462) 로 전송할 수도 있으며, 모든 버퍼들이 준비되어 있다는 것을 안드로이드 MM 프레임워크에 표시하는 휴지 신호 (1428) 를 전송할 수도 있다. 안드로이드 MM 프레임워크 (1450) 는 그후 실행으로 전환하는 신호 (1430) 를 OMX 컴포넌트로 전송할 수도 있다. 안드로이드 MM 프레임워크 (1450) 는 또한 OMX 컴포넌트 (1452) 에 헤더로 제 1 버퍼를 큐잉하는 신호 (1432) 를 전송할 수도 있다. OMX 컴포넌트 (1452) 는 또한 헤더로 제 1 버퍼를 큐잉하는 신호 (1434) 를 V4L2 비디오 드라이버 (1454) 로 전송할 수도 있다.

[0125] V4L2 비디오 드라이버 (1454) 는 제 1 버퍼를 코어 (1462) 에 맵핑하는 신호 (1436) 를 커널 페이지 할당기 (1456) 로 전송할 수도 있다. 커널 페이지 할당기 (1456) 는 제 1 버퍼를 제 1 스테이지 변환 입력 컨텍스트 뱅크들에 맵핑하는 신호 (1438) 를 SMMU (1460) 로 시그널링할 수도 있다. V4L2 비디오 드라이버 (1454) 는 또한 헤더로 제 1 버퍼를 큐잉하는 신호 (1440) 를 코어 (1462) 로 전송할 수도 있다.

[0126] 도 15 는 하이퍼바이저를 인에이블하기 위해 (즉, 하이퍼바이저가 도 10 을 참조하여 위에서 설명된 블록 (1010) 에서 디스에이블된 후) 컴퓨팅 디바이스 프로세서에서 구현될 수도 있는 양태 방법 (1012a) 을 예시한다. 블록 (1504) 에서, 컴퓨팅 디바이스는 HCR.VM 을 "1" 로 설정함으로써 PL0 및 PL1 (즉, 특권 레벨 (privilege level) 1 및 특권 레벨 2) 제 2 스테이지 MMU 를 인에이블할 수도 있다. 블록 (1506) 에서, 컴퓨팅 디바이스는 인터럽트 요청들 ("IRQ") 이 하이퍼바이저 모드에서 취해지도록 구성할 수도 있다. 일 양태에서, 컴퓨팅 디바이스는 SCR.IMO 를 "1" 로 설정함으로써 이것을 달성할 수도 있다.

[0127] 컴퓨팅 디바이스는 또한 모든 활성 SMMU 컨텍스트 뱅크들을, 제 1 스테이지 변환들이 제 2 스테이지 변환들과 네스트되는 상태에 두기 위해, 블록 (1508) 에서 SMMU 드라이버들을 호출할 수도 있다. 컴퓨팅 디바이스는 SMMU_CBARn.type 엘리먼트를 "0b11" 로 설정함으로써 이 상태를 설정할 수도 있다.

[0128] 일 양태에서, 하이퍼바이저는 일단 이들 단계들이 완수되면, 도 10 을 참조하여 위에서 설명한 바와 같이 블록 (1012) 에서 인에이블될 수도 있다. 블록 (1012) 에서 인에이블되어진 후, 하이퍼바이저는 옵션적으로, 옵션적인 블록 (1510) 에서 DSP 와 프로세서간 통신들 (IPC) 을 시작할 수도 있다. 하이퍼바이저는 또한 옵션적으로, 옵션적인 블록 (1512) 에서 SMMU 결함들을 핸들링할 수도 있다.

[0129] 일부 양태들에서, 하이퍼바이저는 인에이블될 때 여러 다른 활동들을 재개할 수도 있다. 예를 들어, 하이퍼바이저는 옵션적인 블록 (1514) 에서 I/O 액세스들을 제한하는 것을 재개할 수도 있다. 하이퍼바이저는 또한 블록 (1516) 에서 하드웨어 인터럽트 액세스를 제한하는 것을 재개할 수도 있다. 블록 (1518) 에서, 하이퍼바이저는 추가적으로 하드웨어 타이머 액세스들을 제한하는 것을 재개할 수도 있다.

[0130] 컴퓨팅 디바이스 프로세서는 그후 도 10 을 참조하여 위에서 설명한 바와 같이 블록 (1014) 에서 액세스 제어를 구현할 수도 있다.

[0131] 일부 양태들에서, 여러 하이퍼바이저 기능들 (예컨대, 액세스 제어, 샌드박싱 메모리, 등) 은 집적 회로 경계 및/또는 칩 경계를 가로질러 인에이블될 수도 있다. 도 13 과 관련하여 위에서 설명된 바와 같이, 일 양태에서, 마스터 칩세트 (즉, 마스터 하이퍼바이저) 는 예를 들어, 주변장치 컴포넌트 상호접속 익스프레스 인터페이스 (peripheral component interconnect express interface) 를 통해서, 다른 칩세트들에서의 샌드박싱 메모리를 제어할 수도 있다. 예를 들어, 애플리케이션 프로세서 칩세트에서 마스터 하이퍼바이저는 모뎀 또는 DSP 칩세트에서 중간 물리 어드레스들로부터 물리 어드레스들로의 변환들을 제어할 수도 있다. 따라서, 하이퍼바이저 기능들이 인에이블될 때, 마스터 하이퍼바이저는 다수의 별개의 칩들에 걸쳐서 그들 기능들을 수행

할 수도 있다.

[0132] 도 16a 및 도 16b 는 샌드박스 세션들 동안 제 2 스테이지 변환들을 수행하기 위해 컴퓨팅 디바이스 프로세서 상에서 구현될 수도 있는 양태 방법들을 예시한다. 여러 양태들에서, 하이퍼바이저는 여러 컴포넌트들 (예컨대, HLOS 또는 DSP) 이 액세스할 수도 있는 물리 메모리 어드레스 공간에서의 로케이션들을 관리할 수도 있다.

[0133] 도 16a 는 샌드박스 세션 동안 하이퍼바이저가 메모리를 할당하는 양태 방법 (1014a) 을 예시한다. 하이퍼바이저가 도 10 을 참조하여 위에서 설명한 바와 같이 블록 1012 에서 인에이블될 때.

[0134] 하이퍼바이저는 옵션적인 결정 블록 (1604) 에서 공유된 가상 메모리 상황이 현재 존재하는지 여부를 결정할 수도 있다. 일 양태에서, 공유된 가상 메모리 상황은, 예를 들어, HLOS 가 또 다른 컴포넌트와 가상 메모리를 공유하고 있을 때 존재할 수도 있다. 공유된 가상 메모리 상황이 일어나면 (즉, 결정 블록 1604 = "예"), 하이퍼바이저는 도 16b 를 참조하여 아래에서 설명되는 방법 (1014b) 을 실행할 수도 있다. 그렇지 않으면 (즉, 결정 블록 1604 = "아니오"), 하이퍼바이저는 블록 (1608) 에서 메모리를 할당하려는 시도를 모니터링할 수도 있다. 하이퍼바이저는 결정 블록 (1610) 에서 HLOS 가 메모리를 할당하려고 시도하고 있는지 여부를 결정할 수도 있다. 일 양태에서, HLOS 는 HLOS 상에서 현재 동작하고 있는 애플리케이션들 또는 프로세스들에 대해 메모리를 할당하려고 시도할 수도 있다. 예를 들어, HLOS 는 그 애플리케이션에 의해 액세스되는 가상 어드레스 공간을 생성함으로써 애플리케이션에 대해 메모리를 할당할 수도 있다. HLOS 가 메모리를 할당하려고 시도하고 있다고 프로세서가 결정하면 (즉, 결정 블록 1610 = "예"), 하이퍼바이저는 블록 (1612) 에서 HLOS 에 액세스가능한 물리 메모리 어드레스 공간으로부터의 물리 어드레스들을 HLOS 에 제공할 수도 있다.

일 양태에서, HLOS 는 물리 어드레스들 중 일부가 HLOS 의 제 2 스테이지 변환 맵핑들로부터 제거되기 때문에 물리 어드레스 공간에의 임의의 액세스를 행하지 못할 수도 있으며, 보호된 콘텐츠에 대해 할당될 수도 있다. 예를 들어, HLOS 는 4kb 비디오 버퍼들을 저장하기 위해 DSP 에 할당되는 물리 어드레스에의 맵핑을 행하지 못할 수도 있다. 하이퍼바이저는 그후 도 10 을 참조하여 위에서 설명된 결정 블록 (1016) 에서 샌드박스 세션이 종료되었는지 여부를 결정할 수도 있다.

[0135] HLOS 가 메모리를 할당하려고 시도하고 있지 않으면 (즉, 결정 블록 1610 = "아니오"), 하이퍼바이저는 결정 블록 (1611) 에서, 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있는지 여부를 결정할 수도 있다. 예를 들어, 하이퍼바이저는 보안 비디오 신호를 프로세싱하는 DSP 가 4kb 비디오 버퍼들을 물리 메모리에 저장하려고 시도하고 있는지 여부를 결정할 수도 있다. 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있지 않으면 (즉, 결정 블록 1611 = "아니오"), 하이퍼바이저는 도 10 을 참조하여 위에서 설명된 결정 블록 (1016) 에서, 샌드박스 세션이 종료되었는지 여부를 결정할 수도 있다.

[0136] 샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있다고 하이퍼바이저가 결정하면 (즉, 결정 블록 1611 = "예"), 하이퍼바이저는 블록 (1614) 에서, HLOS 에 액세스가능한 물리 어드레스 공간에서의 물리 어드레스들로부터 샌드박스화된 컴포넌트에 제공될 물리 어드레스들을 제거할 수도 있다. 하이퍼바이저는 또한 블록 (1616) 에서, 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공할 수도 있다. 일 양태에서, 이용가능한 물리 어드레스들은 HLOS 에 할당되지 않은 물리 어드레스 공간에서의 물리 어드레스들일 수도 있다. 다시 말해서, 이용가능한 물리 어드레스들은 "해방 (free)" 메모리 어드레스들이다. 일 양태에서, 일단 하이퍼바이저가 샌드박스화된 컴포넌트에 의한 사용을 위해 물리 어드레스 공간에 메모리를 할당하면, HLOS 는 샌드박스 세션 동안 그 물리 메모리에 더 이상 액세스하지 않을 수도 있다. 예를 들어, 일단 보안 비디오에 대한 4kb 비디오 버퍼가 특정의 물리 어드레스에 저장되면, HLOS 는 그 물리 어드레스에 더 이상 맵핑하지 않을 수도 있다 (즉, HLOS 는 그것을 할당할 그 물리 어드레스들을 더 이상 "인식할 수" 없을 수도 있다). 하이퍼바이저는 그후 도 10 을 참조하여 위에서 설명된 결정 블록 (1016) 에서 샌드박스 세션이 종료되었는지 여부를 결정할 수도 있다.

[0137] 도 16b 는 HLOS 가 또 다른 컴포넌트와 가상 메모리를 공유하고 있는 동안에 (즉, 결정 블록 1604 = "예" 일 때) 샌드박스 세션 동안 메모리를 할당하기 위해 하이퍼바이저에서 구현될 수도 있는 양태 방법 (1014b) 을 예시한다. 결정 블록 (1624) 에서, 하이퍼바이저는 HLOS 가 물리 어드레스들을 할당하려고 시도하고 있는지 여부를 결정할 수도 있다. 예를 들어, HLOS 는 HLOS 상에서 실행하는 애플리케이션들에 의한 사용을 위해 물리 어드레스 공간에서의 어떤 물리 어드레스들을 할당하려고 시도하고 있을 수도 있다. HLOS 가 물리 어드레스들을 할당하려고 시도하고 있다고 하이퍼바이저가 결정하면 (즉, 결정 블록 1624 = "예"), 하이퍼바이저는 블록 (1612) 에서, HLOS 에 액세스가능한 물리 메모리 어드레스 공간에서의 물리 어드레스들로부터 HLOS 에

물리 어드레스들을 제공할 수도 있다. 도 16a 를 참조하여 위에서 설명된 양태에서, 물리 어드레스들은 예를 들어, 하이퍼바이저가 HLOS 로 하여금 그들 물리 어드레스들을 할당하게 할 때 HLOS 에 액세스가능할 수도 있다. 다시 말해서, HLOS 에 액세스가능한 물리 어드레스들은 샌드박스화된 컴포넌트에 할당되어 있지 않고 HLOS 로부터 숨겨져 있지 않을 수도 있다. 물리 어드레스들을 HLOS 에 할당한 후, 하이퍼바이저는 도 10 을 참조하여 위에서 설명된 결정 블록 (1016) 에서 샌드박스 세션이 종료되었는지 여부를 결정할 수도 있다.

[0138] HLOS 가 물리 어드레스들을 할당하려고 시도하고 있지 않다고 하이퍼바이저가 결정하면 (즉, 결정 블록 1624 = "아니오"), 하이퍼바이저는 결정 블록 (1626) 에서, 샌드박스화된 컴포넌트가 물리 어드레스들을 할당하려고 시도하고 있는지 여부를 결정할 수도 있다. 예를 들어, 샌드박스화된 컴포넌트 내에서 동작하는 DSP 는 4kb 비디오 버퍼들을 저장하기 위해 어떤 물리 어드레스에 액세스하려고 시도할 수도 있다. 샌드박스화된 컴포넌트가 물리 어드레스들을 할당하려고 시도하고 있다고 하이퍼바이저가 결정하면 (즉, 결정 블록 1626 = "예"), 하이퍼바이저는 블록 (1628) 에서, HLOS 에 액세스가능한 물리 메모리 어드레스 공간에서의 PA들로부터 샌드박스화된 컴포넌트에 제공될 물리 어드레스들을 제거할 수도 있다. 일 양태에서, 샌드박스화된 컴포넌트에 할당된 물리 어드레스들이 HLOS 로부터 숨겨질 수도 있다. 다시 말해서, 하이퍼바이저는 HLOS 로부터 샌드박스화된 컴포넌트에 할당된 그들 물리 어드레스들로의 제 2 스테이지 맵핑들을 제거할 수도 있다. 하이퍼바이저는 또한 블록 (1632) 에서, 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 샌드박스화된 컴포넌트에 물리 어드레스들을 제공할 수도 있다. 일 양태에서, 이용가능한 물리 어드레스들은 하이퍼바이저가 HLOS 에 의한 사용을 위해 챙겨두지 않은 그들 물리 어드레스들 (즉, 물리 어드레스 공간에서의 "해방" 물리 어드레스들) 을 포함할 수도 있다. 물리 어드레스들을 샌드박스화된 컴포넌트에 할당한 후, 하이퍼바이저는 도 10 을 참조하여 위에서 설명된 결정 블록 (1016) 에서, 샌드박스 세션이 종료되었는지 여부를 결정할 수도 있다.

[0139] 샌드박스화된 컴포넌트가 물리 어드레스들을 할당하려고 시도하고 있지 않다고 하이퍼바이저가 결정하면 (즉, 결정 블록 1626 = "아니오"), 하이퍼바이저는 결정 블록 (1630) 에서 공유 엔티티가 물리 어드레스들을 할당하려고 시도하고 있는지 여부를 결정할 수도 있다. 일 양태에서, 공유 엔티티는 HLOS 와 가상 메모리를 공유하고 있는 컴퓨팅 디바이스 상에서 동작하는 컴포넌트일 수도 있다. 또 다른 양태에서, HLOS 및 또 다른 엔티티는 물리 어드레스들에 대한 포인터들을 공유함으로써 가상 메모리를 공유할 수도 있다. 다시 말해서, HLOS 및 공유 엔티티는 물리 메모리 어드레스 공간에서 동일한 물리 어드레스들을 액세스 또는 할당가능할 수도 있다.

[0140] 공유 엔티티가 물리 어드레스들을 할당하려고 시도하고 있지 않다고 하이퍼바이저가 결정하면 (즉, 결정 블록 1630 = "아니오"), 하이퍼바이저는 도 10 을 참조하여 위에서 설명된 결정 블록 (1016) 에서 샌드박스 세션이 종료되었는지 여부를 결정할 수도 있다. 그렇지 않으면 (즉, 결정 블록 1630 = "예"), 하이퍼바이저는 결정 블록 (1632) 에서 공유 엔티티가 공유된 물리 어드레스들을 할당하려고 시도하고 있는지 여부를 결정할 수도 있다. 다시 말해서, 하이퍼바이저는 공유 엔티티가 HLOS 와 공유된 물리 어드레스를 이용하거나, 변경하거나, 액세스하거나, 할당하거나, 또는 아니면 판독하거나 그에 기록하려고 시도하고 있는지 여부를 결정할 수도 있다.

[0141] 공유 엔티티가 공유된 물리 어드레스들을 할당하려고 시도하고 있다고 하이퍼바이저가 결정하면 (즉, 결정 블록 1632 = "예"), 하이퍼바이저는 블록 (1636) 에서, 공유된 물리 어드레스들을 공유 엔티티에 제공할 수도 있다. 일 양태에서, HLOS 는 또한 공유된 물리 어드레스들을 액세스 및 할당할 수도 있다. 다시 말해서, 하이퍼바이저는 HLOS 로부터 공유 엔티티에 할당된 공유된 물리 어드레스들을 숨기지 않을 수도 있다. 하이퍼바이저는 도 10 을 참조하여 위에서 설명된 결정 블록 (1016) 에서 샌드박스 세션이 종료되었는지 여부를 결정할 수도 있다.

[0142] 그렇지 않고, 공유 엔티티가 공유된 물리 어드레스들을 할당하려고 시도하고 있지 않다고 하이퍼바이저가 결정하면 (즉, 결정 블록 1632 = "아니오"), 하이퍼바이저는 블록 (1634) 에서, HLOS 에 액세스가능한 물리 메모리 어드레스 공간에서의 물리 어드레스들로부터 공유 엔티티에 제공될 물리 어드레스들을 제거할 수도 있다. 블록 (1638) 에서, 하이퍼바이저는 물리 어드레스 공간에서의 이용가능한 물리 어드레스들로부터 공유 엔티티에 물리 어드레스들을 제공할 수도 있다. 일 양태에서, 하이퍼바이저는 공유 엔티티를, 공유 엔티티가 HLOS 와 공유된 물리 어드레스들에 액세스하고 있지 않을 때 위에서 설명한 바와 같은 샌드박스화된 컴포넌트처럼 취급할 수도 있다. 물리 어드레스들에 할당한 후, 하이퍼바이저는 도 10 을 참조하여 위에서 설명된 결정 블록 (1016) 에서 샌드박스 세션이 종료되었는지 여부를 결정할 수도 있다.

[0143] 도 17 은 세션 해제를 수행하기 위해 하이퍼바이저에서 구현될 수도 있는 양태 방법 (1018a) 을 예시한다. 샌드박스 세션이 종료되었다고 하이퍼바이저가 결정할 때 (즉, 결정 블록 1016 = "예"), 하이퍼바이저는 블록 (1704) 에서, 샌드박스화된 컴포넌트의 모든 버퍼들을 해방할 수도 있다. 예를 들어, 하이퍼바이저는 물리 메모리 어드레스 공간에서의 여러 물리 어드레스들에 저장된 4kb 비디오 버퍼들을 해방할 수도 있다. 일 양태에서, 이들 버퍼들을 해방함으로써, 하이퍼바이저는 HLOS 에 액세스가능하게 이들 물리 어드레스들을 준비할 수도 있다.

[0144] 블록 (1706) 에서, 하이퍼바이저는 모든 단편화들을 제거하기 위해 제 2 스테이지 변환 페이지 테이블들을 복원할 수도 있다. 일 양태에서, 하이퍼바이저는 샌드박스화된 컴포넌트에의 메모리 할당들에 의해 펑처링되었을 수도 있는 물리 어드레스 공간에서의 물리 어드레스들을 복원할 수도 있다. 또 다른 양태에서, 하이퍼바이저는 HLOS 로 하여금, 하이퍼바이저가 그들 물리 어드레스들을 샌드박스화된 컴포넌트에 할당한 후에 숨긴 물리 어드레스들에 액세스가능하게 할 수도 있는 제 2 스테이지 맵핑들을 추가할 수도 있다. 하이퍼바이저는 또한 도 10 을 참조하여 위에서 설명한 바와 같이 블록 (1008) 에서 디스에이블될 수도 있다. 따라서, 일 양태에서, 세션 해제 프로시저를 수행한 후, 하이퍼바이저는 전체 물리 메모리 어드레스 공간으로부터 직접 메모리를 할당할 수도 있는 위치에 HLOS 를 다시 둔 후, 디스에이블될 수도 있다.

[0145] 여러 양태들과 함께 사용하기에 적합한 전형적인 컴퓨팅 디바이스들 (1800) 은 도 18 에 예시된 컴포넌트들을 공통으로 가질 것이다. 예를 들어, 전형적인 컴퓨팅 디바이스 (1800) 는 내부 메모리 (1801), 디스플레이 (1803), 및 스피커 (1864) 에 커플링된 프로세서 (1802) 를 포함할 수도 있다. 게다가, 컴퓨팅 디바이스는 프로세서 (1802) 에 커플링된, 전자기 방사선을 전송하고 수신하는 안테나 (1804) 를 가질 수도 있다. 일부 양태들에서, 컴퓨팅 디바이스 (1800) 는 시스템온칩들을 포함할 수도 있는 하나 이상의 특수 또는 범용 프로세서들 (1805, 1824) 을 포함할 수도 있다. 컴퓨팅 디바이스들은 일반적으로 또한 키 패드 또는 소형 키보드 (미도시) 및 사용자 입력들을 수신하기 위한 메뉴 선택 버튼들 (1808a, 1808b) 을 포함한다. 컴퓨팅 디바이스들은 또한 컴퓨팅 디바이스들을 턴온 및 턴오프하는 전원 버튼 (1834) 을 포함할 수도 있다.

[0146] 도 19 에 예시된 랩탑 컴퓨터 (1900) 와 같은, 다른 유형들의 컴퓨팅 디바이스들이 또한 여러 양태들을 구현하고 그들로부터 이점을 취할 수도 있다. 랩탑 컴퓨터 (1900) 와 같은 컴퓨팅 디바이스들은 일반적으로 내부 메모리 (1901) 및 대용량 비휘발성 메모리, 예컨대 디스크 드라이브 (1905) 또는 플래시 메모리에 커플링된 프로세서 (1902), 및 디스플레이 (1909) 를 포함한다. 컴퓨팅 디바이스들은 또한 키보드 (1908) 및 사용자 입력들을 수신하는 선택 버튼들 (1907) 을 포함할 수도 있다.

[0147] 여러 양태들을 구현하는 컴퓨팅 디바이스들에 사용되는 프로세서들 (1802, 1805, 1824, 1902) 은 여러 본원에서 설명하는 양태들의 기능들을 포함한, 다양한 기능들을 수행하는 프로세서 실행가능한 소프트웨어 명령들 (애플리케이션들) 에 의해 구성될 수 있는, 임의의 프로그래밍가능 마이크로프로세서, 마이크로컴퓨터, 또는 다중 프로세서 칩 또는 칩들일 수도 있다. 일반적으로, 소프트웨어 애플리케이션들 및 프로세서 실행가능한 명령들은 그들이 프로세서들 (1802, 1805, 1824, 1902) 에 액세스되어 로드되기 전에 내부 메모리 (1801, 1901) 에 저장될 수도 있다. 일부 컴퓨팅 디바이스들에서, 프로세서들 (1802, 1805, 1824, 1902) 은 애플리케이션 소프트웨어 명령들을 저장하기에 충분한 내부 메모리를 포함할 수도 있다.

[0148] 일부 컴퓨팅 디바이스들에서, 보안 메모리는 프로세서 (1802, 1805, 1824, 1902) 에 커플링된 별개의 메모리 칩에 있을 수도 있다. 많은 컴퓨팅 디바이스들에서, 내부 메모리 (1801, 1901) 는 휘발성 또는 비휘발성 메모리, 예컨대, 플래시 메모리, 또는 양쪽의 혼합물일 수도 있다. 메모리는 당업계에 알려져 있는, 위상 변화 메모리 (PCM), 동적 랜덤-액세스 메모리 (DRAM), 정적 랜덤-액세스 메모리 (SRAM), 비휘발성 랜덤-액세스 메모리 (NVRAM), 의사정적 랜덤-액세스 메모리 (PSRAM), 이중 데이터 레이트 동기식 동적 랜덤-액세스 메모리 (DDR SDRAM), 및 다른 랜덤-액세스 메모리 (RAM) 및 판독 전용 메모리 (ROM) 기술들을 포함한, 임의 개수의 상이한 유형들의 메모리 기술들을 포함할 수도 있다. 이 설명의 목적을 위해, 메모리에 대한 전반적인 언급은, 내부 메모리, 컴퓨팅 디바이스에 플러그인되는 착탈식 메모리, 및 프로세서들 내 메모리를 포함한, 프로세서들 (1802, 1805, 1824, 1902) 에 의해 액세스가능한 모든 메모리들을 지칭한다.

[0149] 상기 방법 설명들 및 프로세스 흐름도들은 단지 예시적인 예들로서 제공되며 여러 양태들의 단계들이 제시된 순서로 수행되어야 한다는 것을 요구하거나 또는 암시하려고 의도된 것이 아니다. 당업자가 주지하고 있는 바와 같이 전술한 양태들에서 단계들의 순서는 임의의 순서로 수행될 수도 있다. "그후에 (thereafter)", "그후 (then)", "다음에 (next)" 등과 같은 단어들은 단계들의 순서를 한정하려고 의도되지 않으며; 이들 단어들은 방법들의 설명을 통해서 독자를 안내하기 위해서 단지 사용된다. 또, 단수형으로, 예를 들어, 한정사 "한",

"하나" 또는 "그" 를 이용한, 청구항 엘리먼트들에 대한 임의의 언급은, 그 엘리먼트를 단수에 한정하는 것으로 해석되어서는 안된다.

[0150] 본원에서 개시한 양태들과 관련하여 설명한 여러가지 예시적인 로직 블록들, 모듈들, 회로들, 및 알고리즘 단계들은 전자 하드웨어, 컴퓨터 소프트웨어, 또는 양쪽의 조합들로서 구현될 수도 있다. 이러한 하드웨어와 소프트웨어의 상호 교환가능성을 명확히 예시하기 위하여, 이상에서는, 여러 예시적인 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 그들의 기능성의 관점에서 일반적으로 설명되었다. 이런 기능성이 하드웨어로 구현되는지 또는 소프트웨어로 구현되는지 여부는 특정의 애플리케이션 및 전체 시스템에 부과되는 설계 제약들에 의존한다. 숙련자들은 각각의 특정의 애플리케이션에 대해 다양한 방식들로 설명된 기능성을 구현할 수도 있지만, 이런 구현 결정들이 본 발명의 범위로부터 일탈을 초래하는 것으로 해석되어서는 안된다.

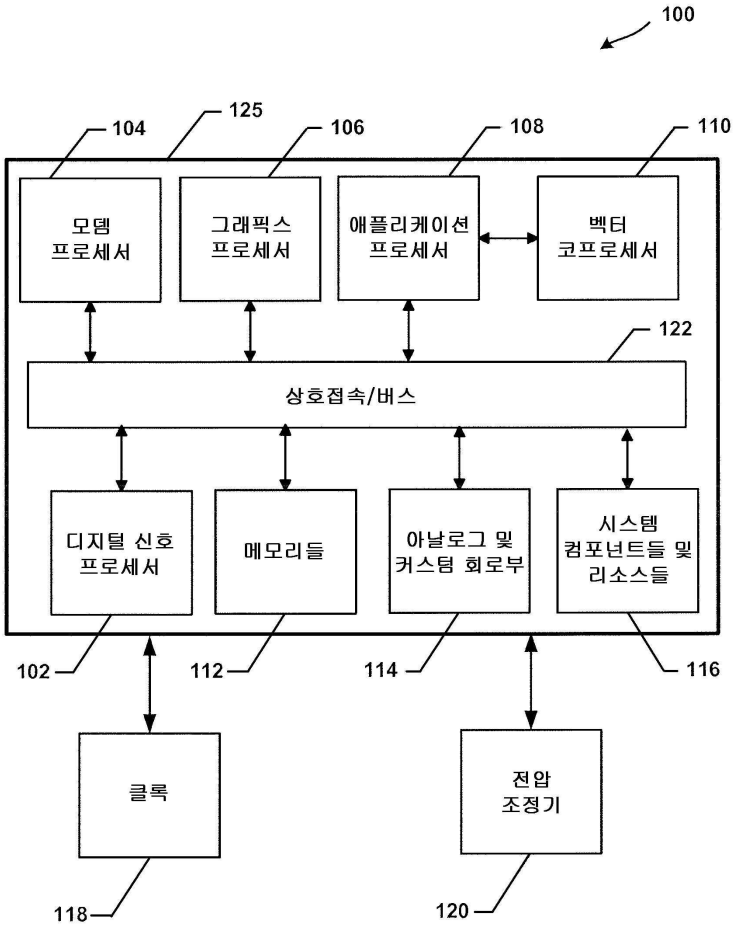
[0151] 본원에서 개시한 양태들과 관련하여 설명한 여러 예시적인 로직들, 로직 블록들, 모듈들, 및 회로들을 구현하기 위해 사용되는 하드웨어는 범용 프로세서, 디지털 신호 프로세서 (DSP), 멀티미디어 브로드캐스트 수신기 칩 내 DSP, 주문형 집적회로 (ASIC), 필드 프로그래밍가능 게이트 어레이 (FPGA) 또는 다른 프로그래밍가능 로직 디바이스, 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 컴포넌트들, 또는 본원에서 설명한 기능들을 수행하도록 설계된 이들의 임의의 조합으로 구현되거나 또는 수행될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 대안적으로는, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한 컴퓨팅 디바이스들의 조합, 예컨대, DSP 와 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 결합된 하나 이상의 마이크로프로세서들, 또는 임의의 다른 이러한 구성으로서 구현될 수도 있다. 이의 대안으로, 일부 단계들 또는 방법들은 주어진 기능에 특유한 회로부에 의해 수행될 수도 있다.

[0152] 하나 이상의 예시적인 양태들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 이 기능들은 비일시적 컴퓨터 판독가능 매체 또는 비일시적 프로세서 판독가능 매체 상에 하나 이상의 명령들 또는 코드로서 저장될 수도 있다. 본원에서 개시한 방법 또는 알고리즘의 단계들은 비일시적 프로세서 판독가능 저장 매체 또는 비일시적 컴퓨터 판독가능 저장 매체 상에 상주할 수도 있는 프로세서 실행가능한 소프트웨어 모듈에서 구현될 수도 있다. 비일시적 컴퓨터 판독가능 또는 프로세서 판독가능 저장 매체들은 컴퓨터 또는 프로세서에 의해 액세스될 수도 있는 임의의 저장 매체들일 수도 있다. 비제한적인 예로서, 이러한 비일시적 컴퓨터 판독가능 또는 프로세서 판독가능 매체들은 RAM, ROM, EEPROM, 플래시 메모리, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기 저장 디바이스들, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수도 있고 컴퓨터에 의해 액세스될 수도 있는 임의의 다른 매체를 포함할 수 있다. 디스크 (disk) 및 디스크 (disc) 는, 본원에서 사용할 때, 컴팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 디스크 (disk) 들은 보통 데이터를 자기적으로 재생하지만, 디스크 (discs) 들은 레이저로 데이터를 광학적으로 재생한다. 앞에서 언급한 것들의 조합들이 또한 비일시적 컴퓨터 판독가능 및 프로세서 판독가능 매체들의 범위 내에 포함된다. 게다가, 방법 또는 알고리즘의 동작들은 컴퓨터 프로그램 제품에 포함될 수도 있는 비일시적 프로세서 판독가능 매체 및/또는 컴퓨터 판독가능 매체 상에, 코드들 및/또는 명령들 중 하나 또는 임의의 조합 또는 세트로서 상주할 수도 있다.

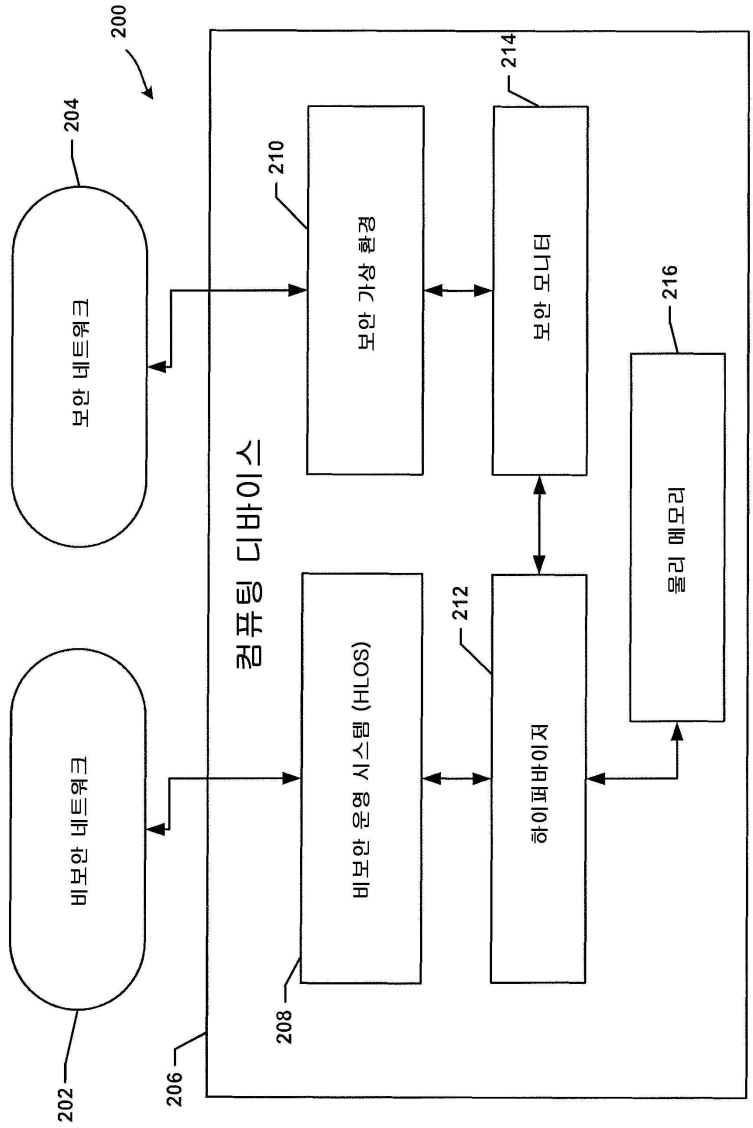
[0153] 개시된 양태들의 선행하는 설명은 임의의 당업자가 본 발명을 실시하고 이용할 수 있도록 제공된다. 이들 양태들에 대한 여러 변경들은 당업자들에게 쉽게 명백할 것이며, 본원에서 정의하는 일반 원리들은 본 발명의 정신 또는 범위로부터 일탈함이 없이 다른 양태들에 적용될 수도 있다. 따라서, 본 발명은 본원에서 나타낸 양태들에 한정시키려는 것이 아니라, 다음 청구범위에 부합하는 최광의 범위 및 본원에서 개시된 원리들 및 신규한 특징들을 부여받게 하려는 것이다.

도면

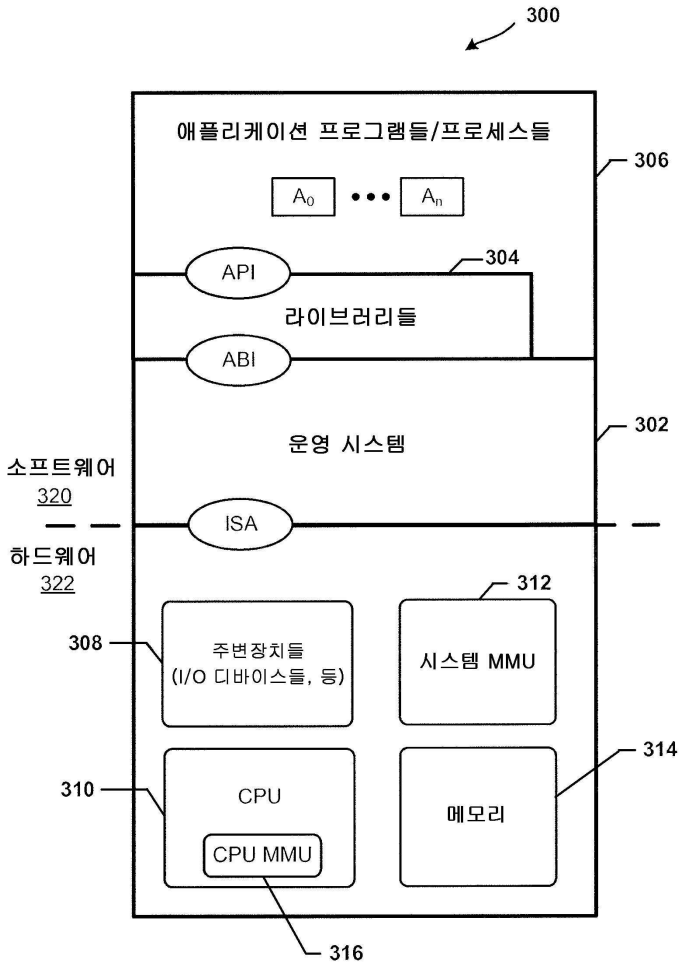
도면1


100
104
125
106
108
110
모뎀 프로세서
그래픽스 프로세서
애플리케이션 프로세서
벡터 코프로세서
122
상호접속/버스
디지털 신호 프로세서
메모리들
아날로그 및 커스텀 회로부
시스템 컴포넌트들 및 리소스들
102
112
114
116
클록
전압 조정기
118
120

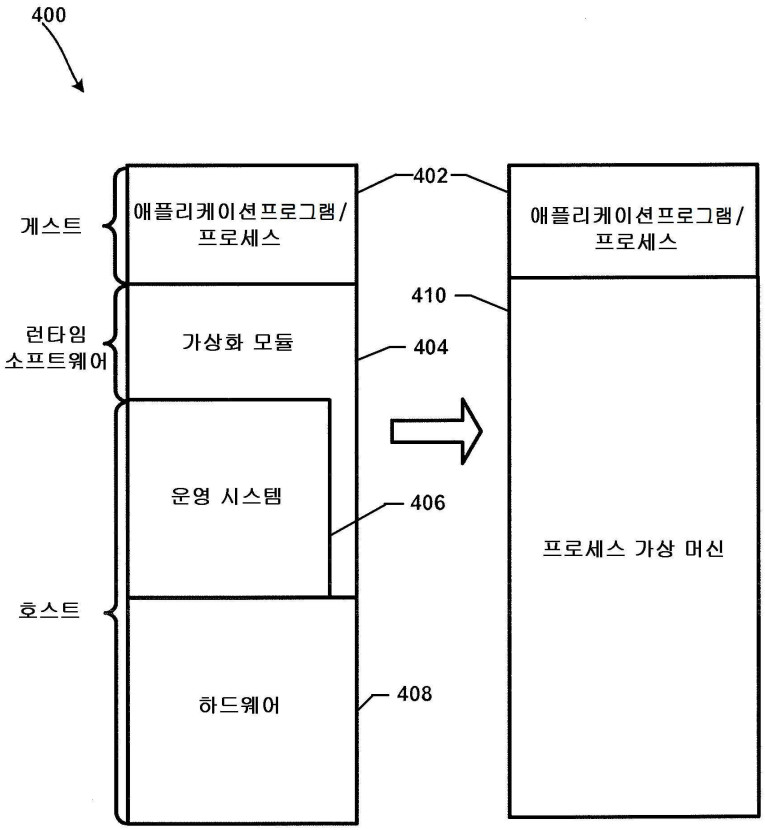
도면2


200
204
보안 네트워크
202
비보안 네트워크
206
컴퓨팅 디바이스
210
보안 가상 환경
208
비보안 운영 시스템 (HLOS)
214
보안 모니터
212
하이퍼바이저
216
물리 메모리

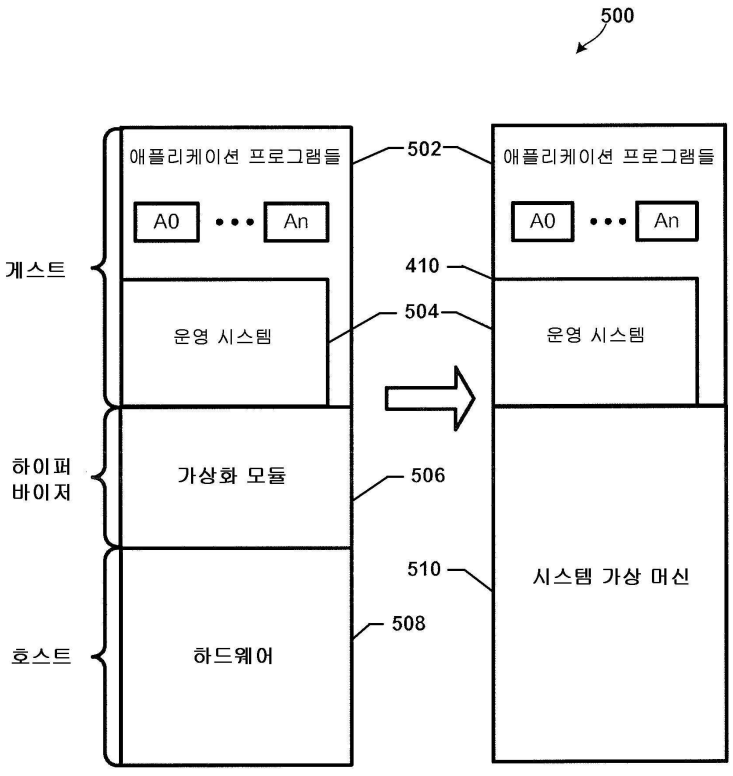
도면3


300
애플리케이션 프로그램들/프로세스들
306
A_0
A_n
API
304
라이브러리들
ABI
운영 시스템
302
소프트웨어
320
ISA
하드웨어
322
312
308
주변장치들
(I/O 디바이스들, 등)
시스템 MMU
314
310
CPU
CPU MMU
메모리
316

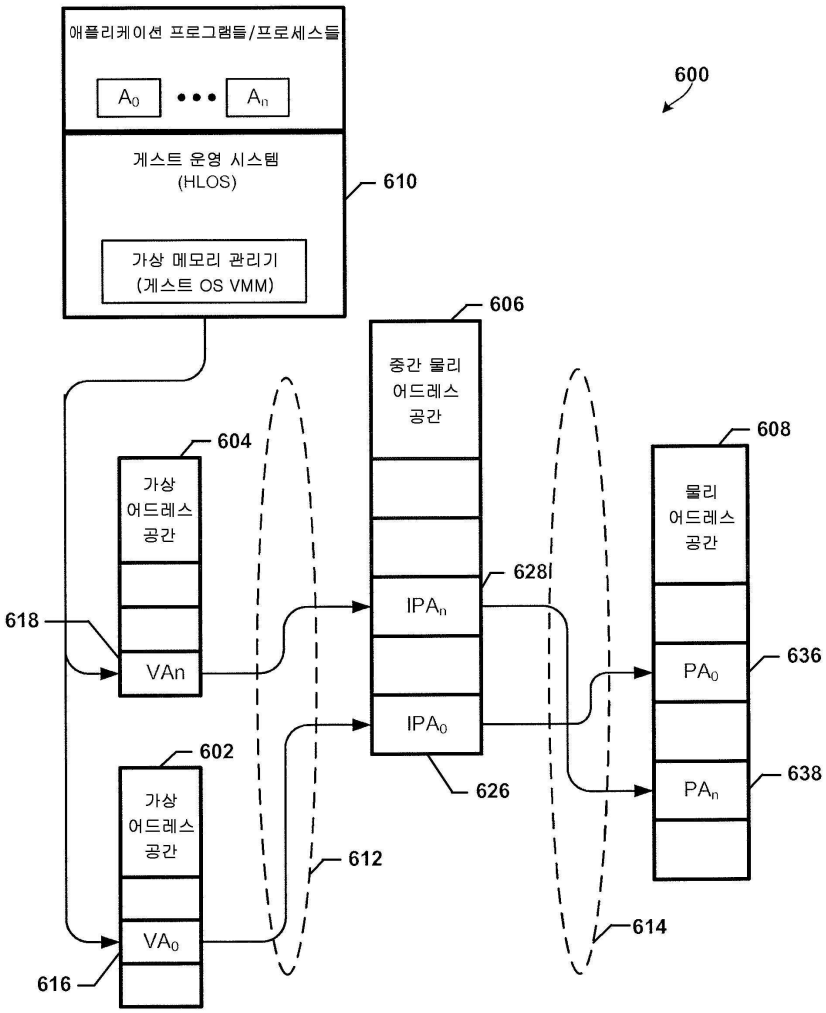
도면4


400
게스트
애플리케이션프로그램/
프로세스
402
애플리케이션프로그램/
프로세스
410
런타임
소프트웨어
가상화 모듈
404
운영 시스템
406
프로세스 가상 머신
호스트
하드웨어
408

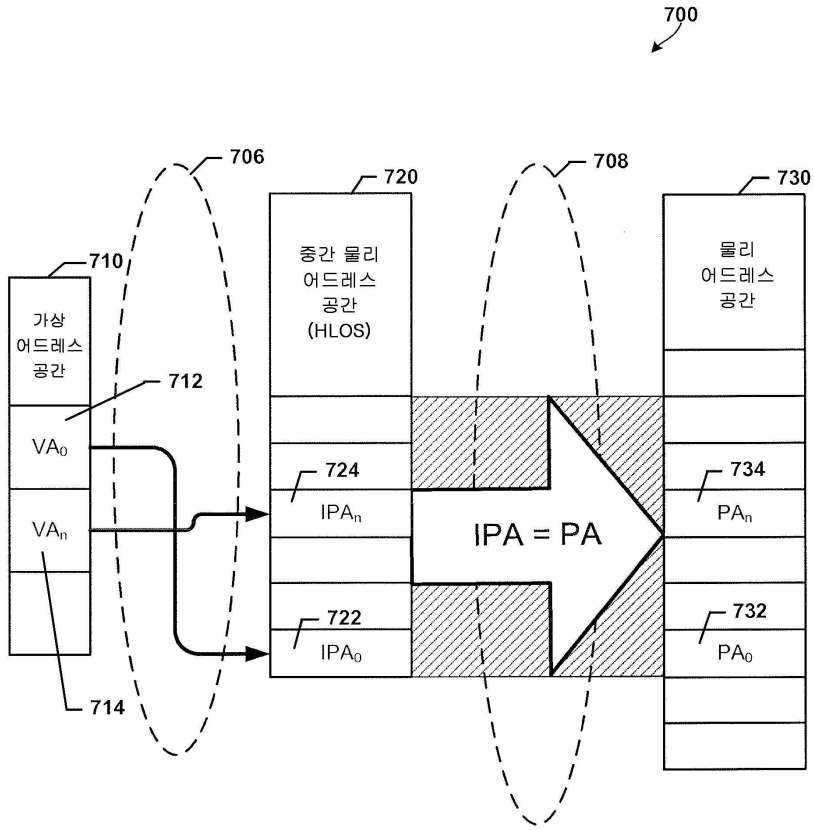
도면5


500
애플리케이션 프로그램들
502
애플리케이션 프로그램들
A0
An
A0
An
게스트
410
운영 시스템
504
운영 시스템
하이퍼
바이저
가상화 모듈
506
510
시스템 가상 머신
호스트
하드웨어
508

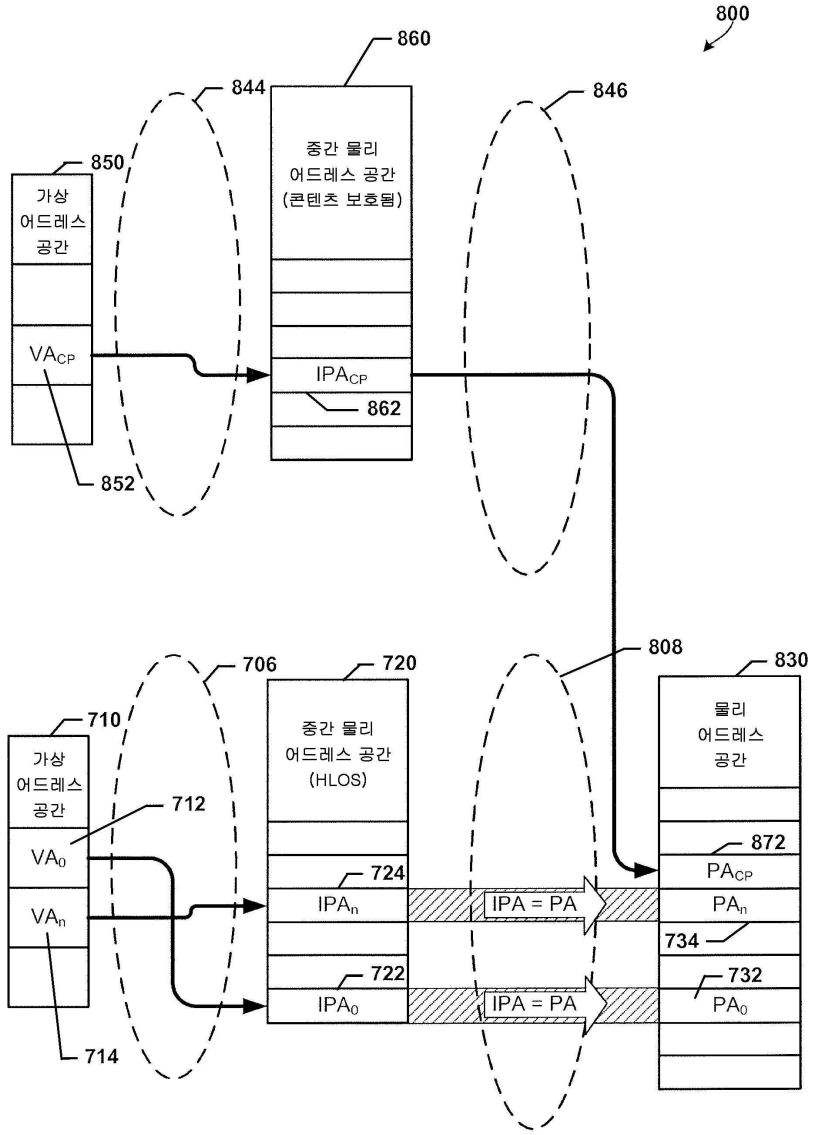
도면6


애플리케이션 프로그램들/프로세스들
A0
An
게스트 운영 시스템
(HLOS)
610
가상 메모리 관리기
(게스트 OS VMM)
600
606
중간 물리
어드레스
공간
604
가상
어드레스
공간
608
물리
어드레스
공간
618
VAn
IPAn
628
PA0
636
IPA0
626
PAn
638
602
가상
어드레스
공간
612
VA0
614
616

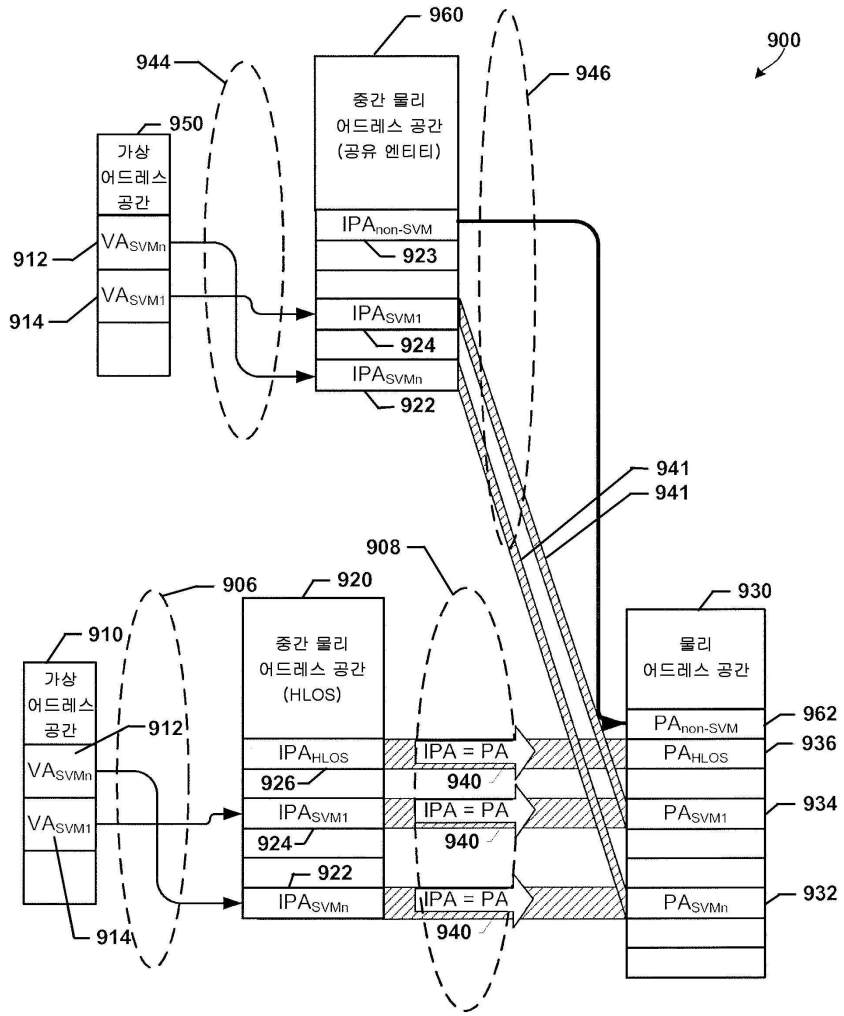
도면7


700
706
708
720
730
710
712
714
724
722
734
732
가상
어드레스
공간
VA0
VAn
중간 물리
어드레스
공간
(HLOS)
IPAn
IPA0
IPA = PA
물리
어드레스
공간
PAn
PA0

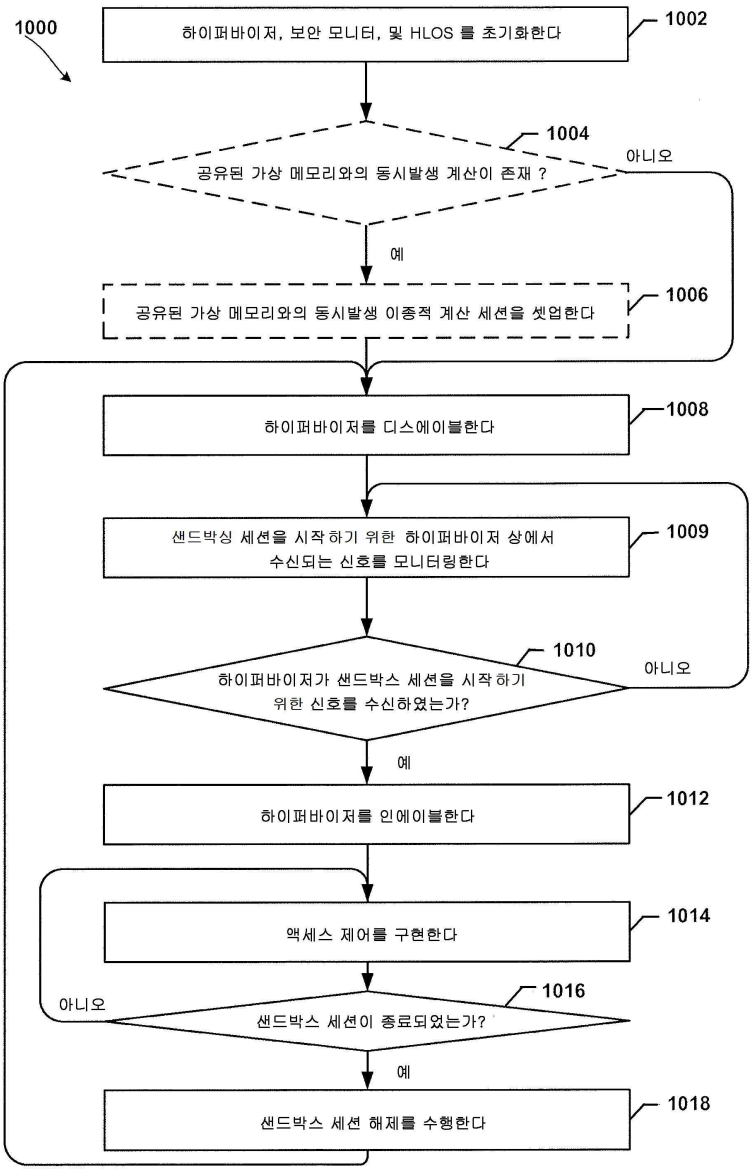
도면8


800
860
844
846
850
가상
어드레스
공간
중간 물리
어드레스 공간
(콘텐츠 보호됨)
VACP
IPACP
862
852
808
706
720
830
710
712
중간 물리
어드레스 공간
(HLOS)
물리
어드레스
공간
872
724
VA0
VAn
IPAn
IPA = PA
PACP
PAn
734
722
732
IPA0
IPA = PA
PA0
714

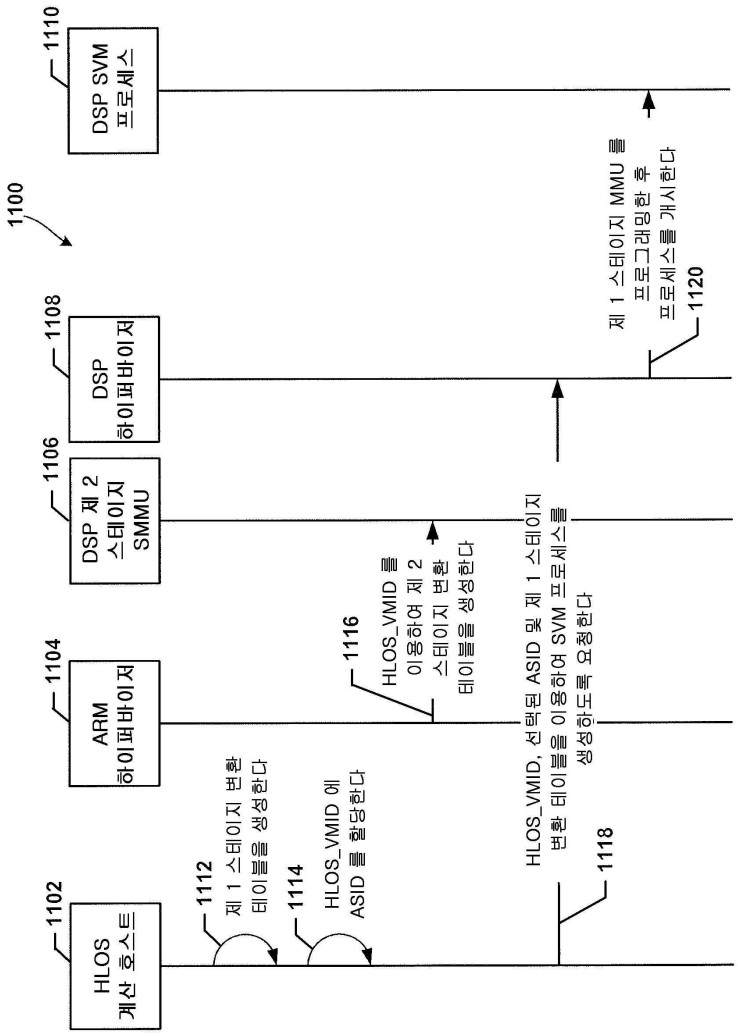
도면9


900
960
944
946
950
가상
어드레스
공간
중간 물리
어드레스 공간
(공유 엔티티)
912
914
VASVMn
VASVM1
IPAnon-SVM
923
IPASVM1
924
IPASVMn
922
941
941
908
906
920
910
930
가상
어드레스
공간
중간 물리
어드레스 공간
(HLOS)
물리
어드레스 공간
912
914
IPAHLOS
926
IPA = PA
940
PAnon-SVM
962
PAHLOS
936
PASVM1
934
PASVMn
932

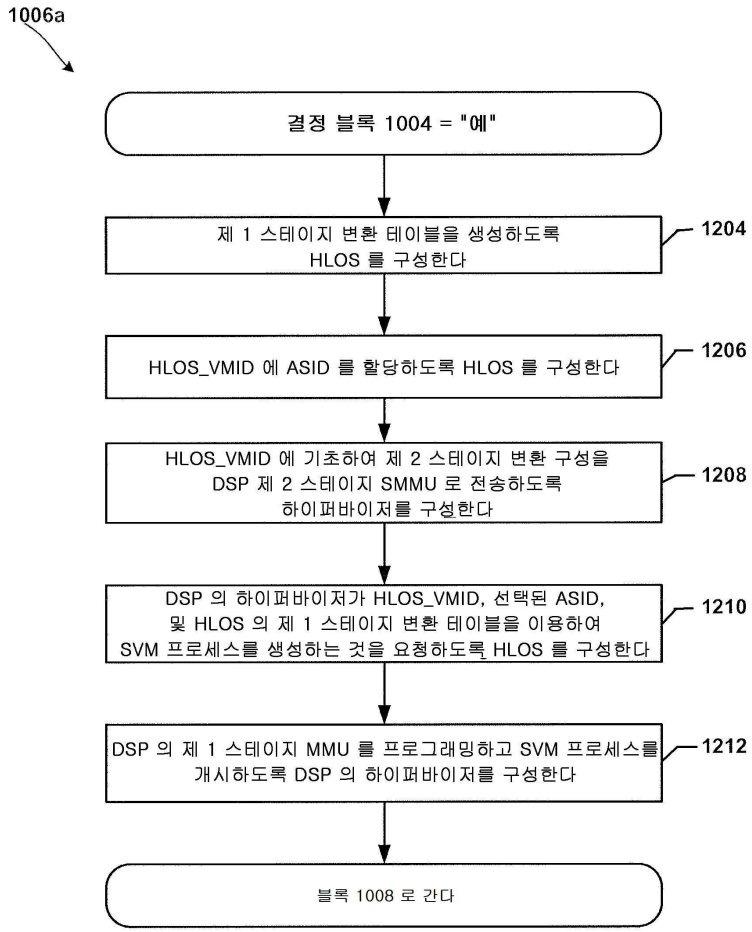
도면10


1000
하이퍼바이저, 보안 모니터, 및 HLOS 를 초기화한다
1002
1004
공유된 가상 메모리와의 동시발생 계산이 존재 ?
아니오
예
공유된 가상 메모리와의 동시발생 이종적 계산 세션을 셋업한다
1006
하이퍼바이저를 디스에이블한다
1008
샌드박싱 세션을 시작하기 위한 하이퍼바이저 상에서 수신되는 신호를 모니터링한다
1009
1010
하이퍼바이저가 샌드박스 세션을 시작하기 위한 신호를 수신하였는가?
아니오
예
하이퍼바이저를 인에이블한다
1012
액세스 제어를 구현한다
1014
1016
아니오
샌드박스 세션이 종료되었는가?
예
샌드박스 세션 해제를 수행한다
1018

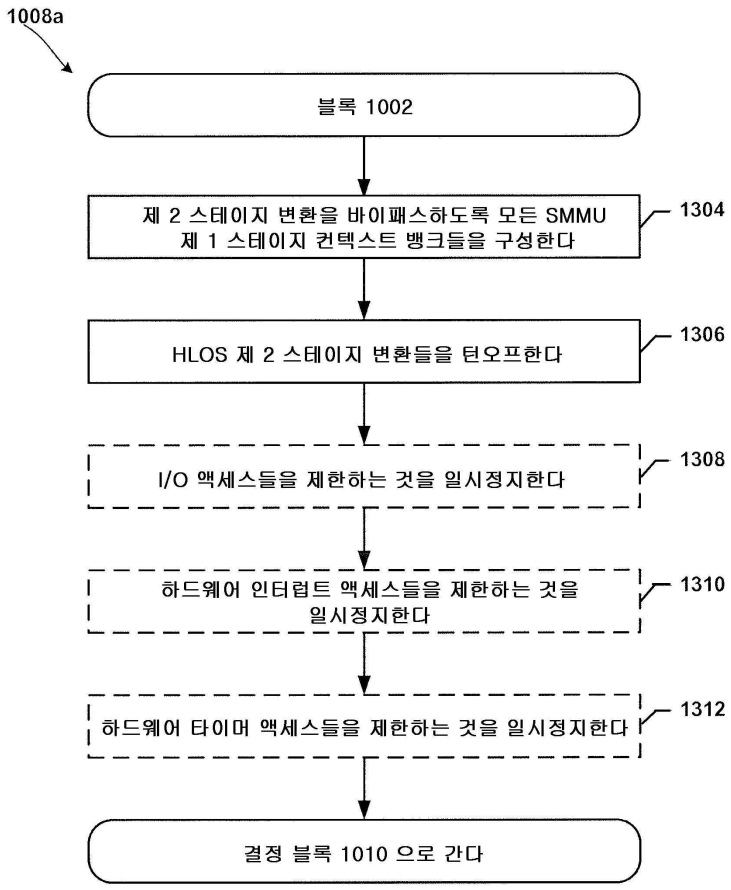
도면11


1100
1102
HLOS 계산 호스트
1104
ARM 하이퍼바이저
1106
DSP 제 2 스테이지 SMMU
1108
DSP 하이퍼바이저
1110
DSP SVM 프로세스
1112
제 1 스테이지 변환 테이블을 생성한다
1114
HLOS_VMID 에 ASID 를 할당한다
1116
HLOS_VMID 를 이용하여 제 2 스테이지 변환 테이블을 생성한다
1118
HLOS_VMID, 선택된 ASID 및 제 1 스테이지 변환 테이블을 이용하여 SVM 프로세스를 생성하도록 요청한다
1120
제 1 스테이지 MMU 를 프로그래밍한 후 프로세스를 개시한다

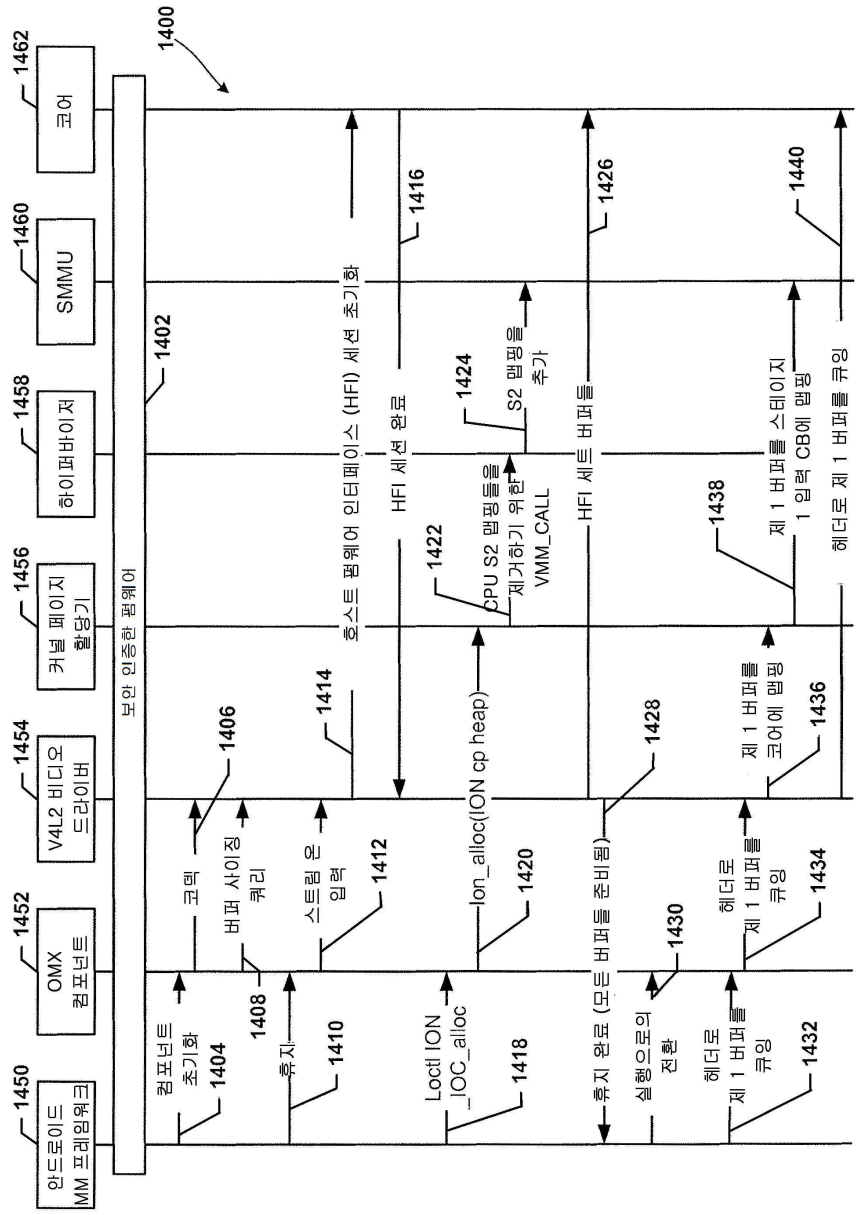
도면12


1006a
결정 블록 1004 = "예"
제 1 스테이지 변환 테이블을 생성하도록 HLOS 를 구성한다
1204
HLOS_VMID 에 ASID 를 할당하도록 HLOS 를 구성한다
1206
HLOS_VMID 에 기초하여 제 2 스테이지 변환 구성을 DSP 제 2 스테이지 SMMU 로 전송하도록 하이퍼바이저를 구성한다
1208
DSP 의 하이퍼바이저가 HLOS_VMID, 선택된 ASID, 및 HLOS 의 제 1 스테이지 변환 테이블을 이용하여 SVM 프로세스를 생성하는 것을 요청하도록 HLOS 를 구성한다
1210
DSP 의 제 1 스테이지 MMU 를 프로그래밍하고 SVM 프로세스를 개시하도록 DSP 의 하이퍼바이저를 구성한다
1212
블록 1008 로 간다

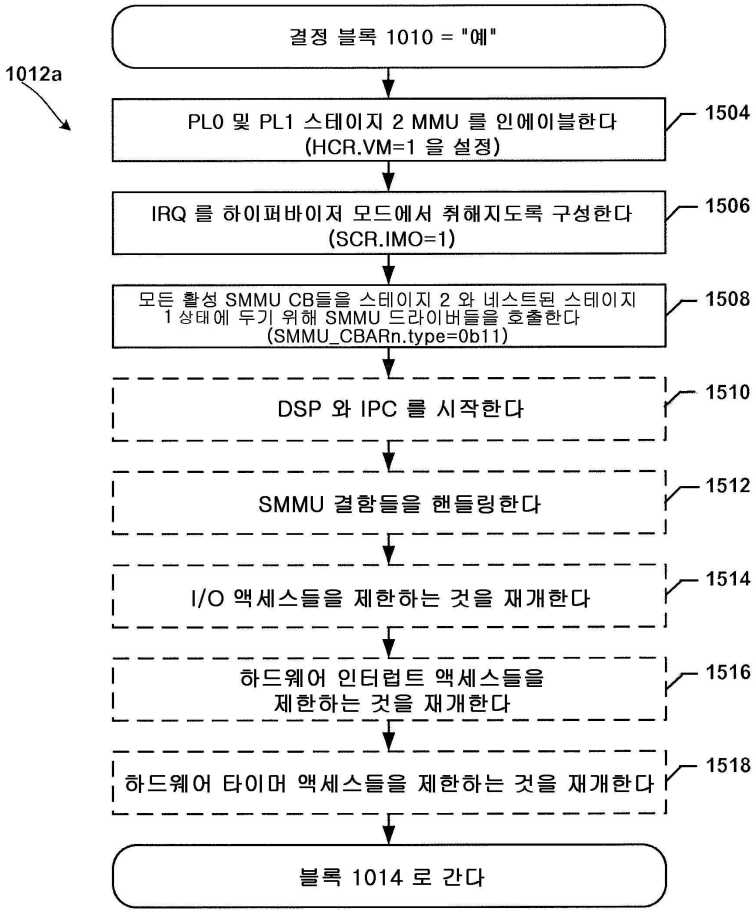
도면13


1008a
블록 1002
제 2 스테이지 변환을 바이패스하도록 모든 SMMU 제 1 스테이지 컨텍스트 뱅크들을 구성한다
1304
HLOS 제 2 스테이지 변환들을 턴오프한다
1306
I/O 액세스들을 제한하는 것을 일시정지한다
1308
하드웨어 인터럽트 액세스들을 제한하는 것을 일시정지한다
1310
하드웨어 타이머 액세스들을 제한하는 것을 일시정지한다
1312
결정 블록 1010 으로 간다

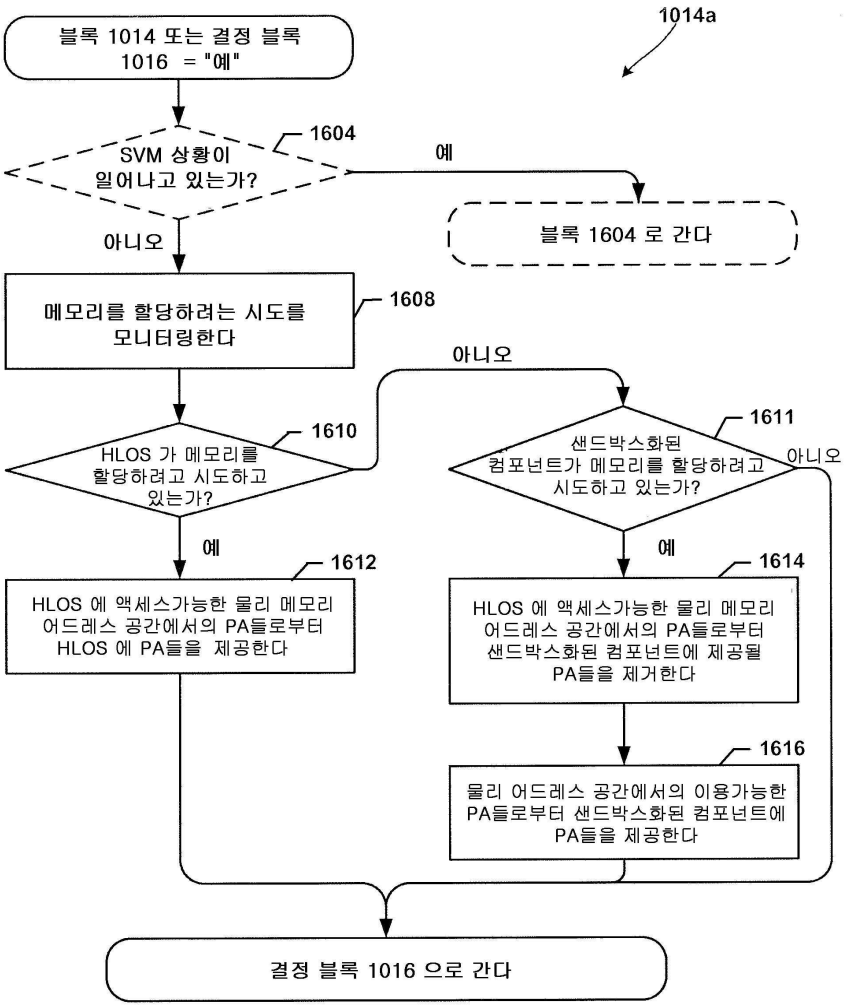
도면14


1400
1462
코어
1460
SMMU
1458
하이퍼바이저
1456
커널 페이지 할당기
1454
V4L2 비디오 드라이버
1452
OMX 컴포넌트
1450
안드로이드 MM 프레임워크
1402
보안 인증한 펌웨어
컴포넌트 초기화
1404
코덱
1406
버퍼 사이징 쿼리
1408
휴지
1410
스트림 온 입력
1412
호스트 펌웨어 인터페이스 (HFI) 세션 초기화
1414
HFI 세션 완료
1416
Loctl ION _IOC_alloc
1418
Ion_alloc(ION cp heap)
1420
CPU S2 맵핑들을 제거하기 위한 VMM_CALL
1422
S2 맵핑을 추가
1424
휴지 완료 (모든 버퍼들 준비됨)
1426
실행으로의 전환
1428
헤더로 제 1 버퍼를 큐잉
1430
HFI 세트 버퍼들
1432
제 1 버퍼를 코어에 맵핑
1434
제 1 버퍼를 스테이지 1 입력 CB에 맵핑
1436
헤더로 제 1 버퍼를 큐잉
1438
1440

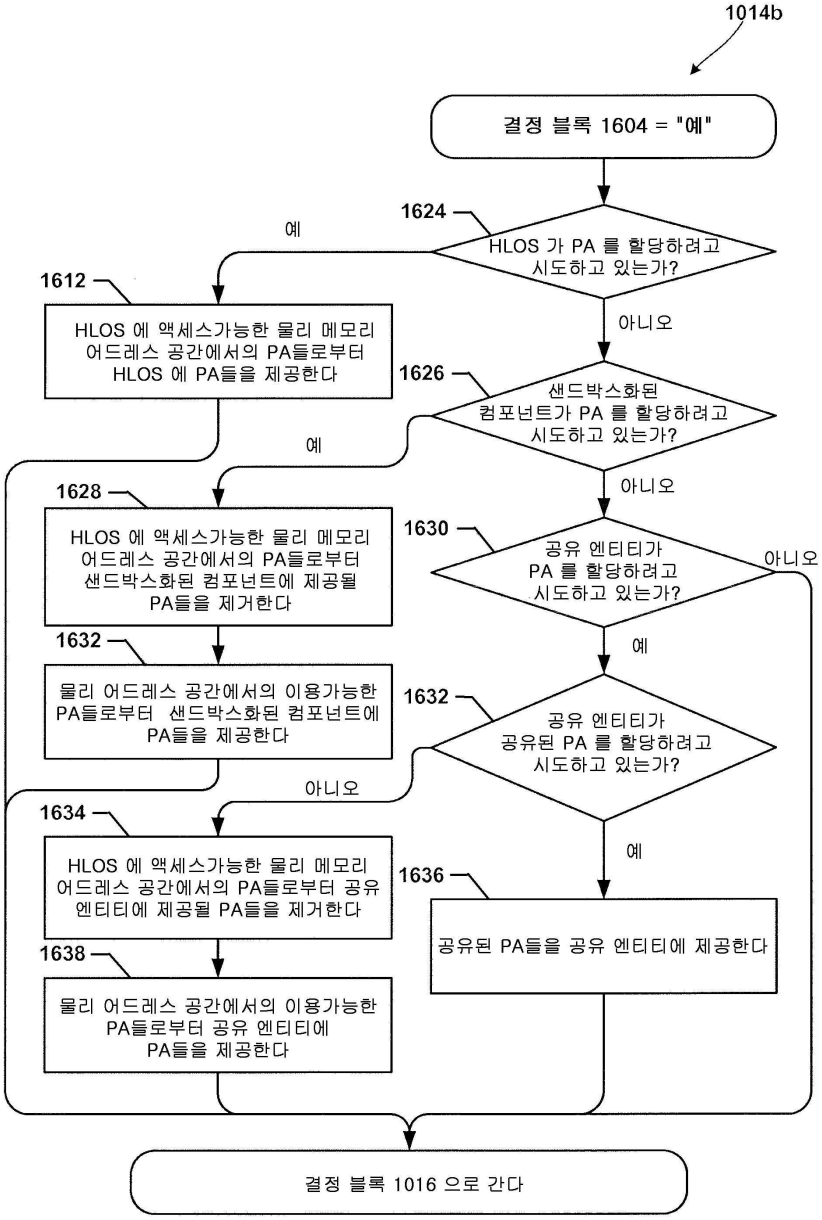
도면15


결정 블록 1010 = "예"
1012a
PL0 및 PL1 스테이지 2 MMU 를 인에이블한다
(HCR.VM=1 을 설정)
1504
IRQ 를 하이퍼바이저 모드에서 취해지도록 구성한다
(SCR.IMO=1)
1506
모든 활성 SMMU CB들을 스테이지 2 와 네스트된 스테이지
1 상태에 두기 위해 SMMU 드라이버들을 호출한다
(SMMU_CBARn.type=0b11)
1508
DSP 와 IPC 를 시작한다
1510
SMMU 결함들을 핸들링한다
1512
I/O 액세스들을 제한하는 것을 재개한다
1514
하드웨어 인터럽트 액세스들을
제한하는 것을 재개한다
1516
하드웨어 타이머 액세스들을 제한하는 것을 재개한다
1518
블록 1014 로 간다

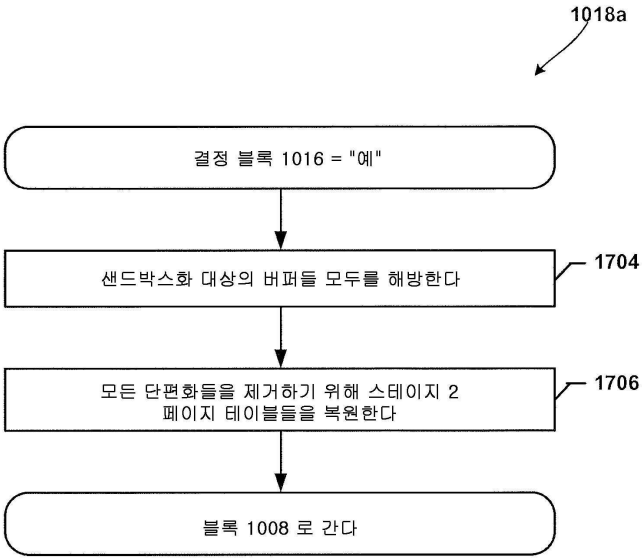
도면16a


1014a
블록 1014 또는 결정 블록 1016 = "예"
1604
SVM 상황이 일어나고 있는가?
예
블록 1604 로 간다
아니오
1608
메모리를 할당하려는 시도를 모니터링한다
아니오
1610
HLOS 가 메모리를 할당하려고 시도하고 있는가?
1611
샌드박스화된 컴포넌트가 메모리를 할당하려고 시도하고 있는가?
아니오
예
예
1612
HLOS 에 액세스가능한 물리 메모리 어드레스 공간에서의 PA들로부터 HLOS 에 PA들을 제공한다
1614
HLOS 에 액세스가능한 물리 메모리 어드레스 공간에서의 PA들로부터 샌드박스화된 컴포넌트에 제공될 PA들을 제거한다
1616
물리 어드레스 공간에서의 이용가능한 PA들로부터 샌드박스화된 컴포넌트에 PA들을 제공한다
결정 블록 1016 으로 간다

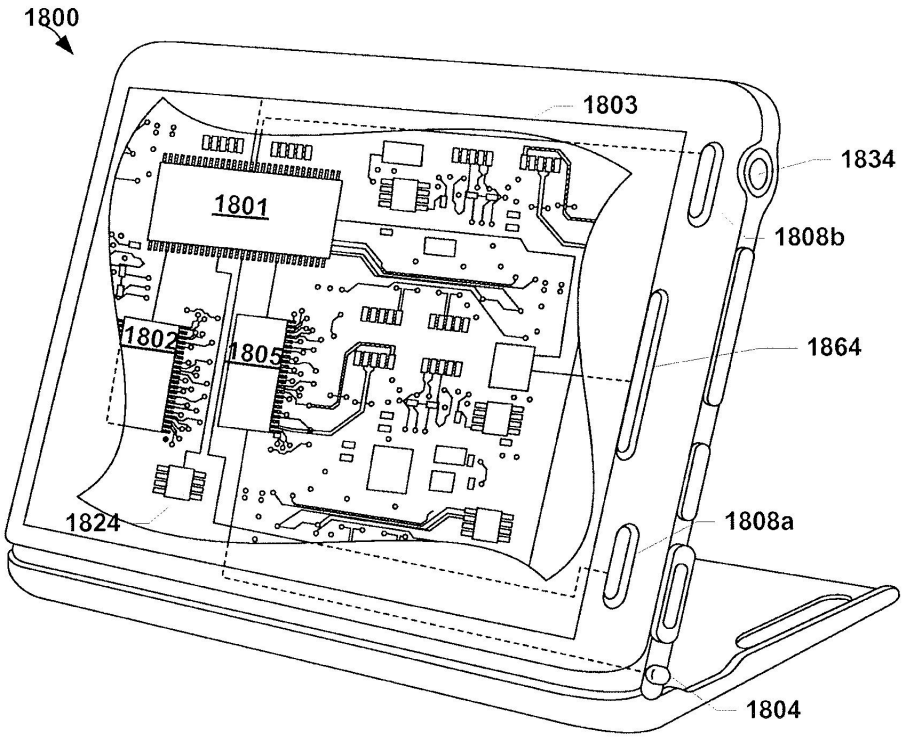
도면16b


1014b
결정 블록 1604 = "예"
1624
HLOS 가 PA 를 할당하려고 시도하고 있는가?
예
아니오
1612
HLOS 에 액세스가능한 물리 메모리 어드레스 공간에서의 PA들로부터 HLOS 에 PA들을 제공한다
1626
샌드박스화된 컴포넌트가 PA 를 할당하려고 시도하고 있는가?
예
아니오
1628
HLOS 에 액세스가능한 물리 메모리 어드레스 공간에서의 PA들로부터 샌드박스화된 컴포넌트에 제공될 PA들을 제거한다
1630
공유 엔티티가 PA 를 할당하려고 시도하고 있는가?
아니오
예
1632
물리 어드레스 공간에서의 이용가능한 PA들로부터 샌드박스화된 컴포넌트에 PA들을 제공한다
1632
공유 엔티티가 공유된 PA 를 할당하려고 시도하고 있는가?
아니오
예
1634
HLOS 에 액세스가능한 물리 메모리 어드레스 공간에서의 PA들로부터 공유 엔티티에 제공될 PA들을 제거한다
1636
공유된 PA들을 공유 엔티티에 제공한다
1638
물리 어드레스 공간에서의 이용가능한 PA들로부터 공유 엔티티에 PA들을 제공한다
결정 블록 1016 으로 간다

도면17


1018a
결정 블록 1016 = "예"
샌드박스화 대상의 버퍼들 모두를 해방한다
1704
모든 단편화들을 제거하기 위해 스테이지 2 페이지 테이블들을 복원한다
1706
블록 1008 로 간다

도면18


1800
1803
1834
1801
1808b
1802
1805
1864
1824
1808a
1804

도면19