



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2017년01월20일  
 (11) 등록번호 10-1698707  
 (24) 등록일자 2017년01월16일

- (51) 국제특허분류(Int. Cl.)  
*G06F 13/40* (2006.01) *G06F 13/42* (2006.01)
- (52) CPC특허분류  
*G06F 13/4068* (2013.01)  
*G06F 13/40* (2013.01)
- (21) 출원번호 10-2015-7020690
- (22) 출원일자(국제) 2013년03월14일  
 심사청구일자 2015년07월29일
- (85) 번역문제출일자 2015년07월29일
- (65) 공개번호 10-2015-0103373
- (43) 공개일자 2015년09월10일
- (86) 국제출원번호 PCT/CN2013/072583
- (87) 국제공개번호 WO 2014/139115  
 국제공개일자 2014년09월18일
- (56) 선행기술조사문헌  
 CN 101676894 B (2011.08.17)

- (73) 특허권자  
**인텔 코퍼레이션**  
 미합중국 캘리포니아 95054 산타클라라 미션 칼리지 블러바드 2200
- (72) 발명자  
**양, 웨이**  
 중국 200241 상하이 민항 지싱 로드 넘버 880
- 수, 차오**  
 중국 200241 상하이 민항 지싱 로드 넘버 880
- 리양, 리**  
 중국 200241 상하이 민항 지싱 로드 넘버 880
- (74) 대리인  
**양영준, 백만기**

전체 청구항 수 : 총 22 항

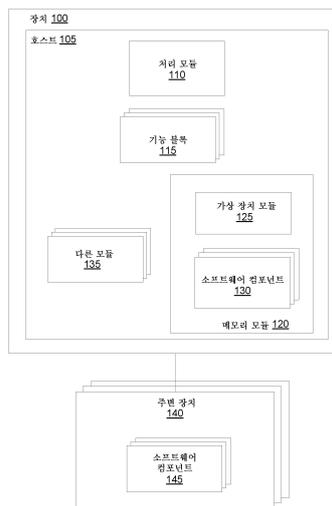
심사관 : 김세영

**(54) 발명의 명칭 가상 PCI 장치 및 가상 MMIO 장치를 형성하기 위한 일반적인 방법**

**(57) 요약**

가상 장치를 가상 PCI(Peripheral Controller Interconnect) 장치 또는 가상 입출력(I/O) 장치 중 적어도 하나로서 형성하기 위한 방법을 구현하기 위한 기술이 개시된다. 본 개시 내용의 방법은 PCI 호환 장치에 대한 요청을 수신하는 단계를 포함한다. 방법은 PCI 호환 장치에 대한 요청에 기초하여 가상 장치를 형성하는 단계를 더 포함하며, 가상 장치는 가상 PCI 장치 또는 가상 I/O 장치 중 적어도 하나로서 형성된다.

**대표도 - 도1**



(52) CPC특허분류  
*G06F 13/42* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

가상 장치를 형성하기 위한 기기(apparatus)로서,

메모리; 및

상기 메모리와 통신 가능하게 결합되는 처리 장치

를 포함하고,

상기 처리 장치는

PCI(Peripheral Controller Interconnect) 호환 장치에 대한 PCI 요청을 수신하고,

상기 PCI 호환 장치의 관련을 결정하고,

상기 관련이 PCI 호환형이 아닌 기능 블록과의 관련이라고 결정하는 것에 응답하여, 상기 PCI 호환 장치에 대한 가상 장치를 형성하고 - 상기 가상 장치는 가상 PCI 장치로서 형성되고, 상기 기능 블록은 상기 처리 장치에 통신가능하게 결합되는 버스와의 상기 처리 장치의 통신들을 핸들링하는 회로를 포함함 - ,

상기 관련이 상기 PCI 호환 장치로부터 부팅될 소프트웨어 컴포넌트와의 관련이라고 결정하는 것에 응답하여, 상기 PCI 호환 장치에 대한 가상 장치를 형성하는 - 상기 가상 장치는 가상 I/O 장치로서 형성됨 - 기기.

#### 청구항 2

제1항에 있어서,

상기 PCI 호환 장치는 소프트웨어 드라이버와 관련되고, 상기 소프트웨어 드라이버는 PCI 호환 액세스 요청을 상기 가상 장치로 전송하고, 상기 가상 장치는 상기 PCI 호환 액세스 요청을 상기 기능 블록에 제공하고, 상기 기능 블록은 상기 PCI 호환 액세스 요청에 기초하여 상기 PCI 호환 장치와 통신하는 기기.

#### 청구항 3

제1항에 있어서,

운영 체제가 상기 가상 장치를 이용하여 상기 PCI 호환 장치로부터 부팅되는 기기.

#### 청구항 4

삭제

#### 청구항 5

제1항에 있어서,

가상 PCI 장치로서 상기 가상 장치를 형성하는 것은,

상기 가상 장치에 대한 판매자 식별자를 결정하고;

상기 가상 장치에 대한 장치 정보를 결정하고;

상기 가상 장치에 대한 주소 정보를 결정하는 것

을 포함하는 기기.

#### 청구항 6

제5항에 있어서,

상기 처리 장치는 또한 상기 PCI 호환 장치에 대한 상기 요청에 응답하여 상기 가상 장치에 대한 상기 판매자

식별자, 상기 가상 장치에 대한 상기 장치 정보 및 상기 장치에 대한 상기 주소 정보를 전송하는 기기.

**청구항 7**

제1항에 있어서,  
 가상 I/O 장치로서 상기 가상 장치를 형성하는 것은,  
 상기 PCI 호환 장치에 대한 상기 PCI 요청을 폐기하고;  
 상기 가상 장치에 대한 I/O 주소 범위를 결정하는 것  
 을 포함하는 기기.

**청구항 8**

제1항에 있어서,  
 상기 처리 장치는 또한  
 I/O 주소에 대한 메모리 액세스를 수신하고;  
 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련되는지를 결정하고;  
 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련된다는 결정 시에 상기 I/O 주소에 대응하는 상기 PCI 장치와 관련된 PCI 주소를 결정하는 기기.

**청구항 9**

제1항에 있어서,  
 상기 가상 I/O 장치는 가상 메모리-맵핑 입출력(MMIO) 장치인 기기.

**청구항 10**

가상 장치를 형성하기 위한 방법으로서,  
 처리 장치에 의해, PCI(Peripheral Controller Interconnect) 호환 장치에 대한 PCI 요청을 수신하는 단계;  
 상기 처리 장치에 의해, 상기 PCI 호환 장치의 관련을 결정하는 단계;  
 상기 관련이 PCI 호환형이 아닌 기능 블록과의 관련이라고 결정하는 것에 응답하여, 상기 처리 장치에 의해, 상기 PCI 호환 장치에 대한 가상 장치를 형성하는 단계 - 상기 가상 장치는 가상 PCI 장치로서 형성되고, 상기 기능 블록은 상기 처리 장치에 통신가능하게 결합되는 버스와의 상기 처리 장치의 통신들을 핸들링하는 회로를 포함함 - ; 및  
 상기 관련이 상기 PCI 호환 장치로부터 부팅될 소프트웨어 컴포넌트와의 관련이라고 결정하는 것에 응답하여, 상기 처리 장치에 의해, 상기 PCI 호환 장치에 대한 가상 장치를 형성하는 단계 - 상기 가상 장치는 가상 I/O 장치로서 형성됨 -  
 를 포함하는 방법.

**청구항 11**

제10항에 있어서,  
 상기 PCI 호환 장치는 소프트웨어 드라이버와 관련되고, 상기 소프트웨어 드라이버는 PCI 호환 액세스 요청을 상기 가상 장치로 전송하고, 상기 가상 장치는 상기 PCI 호환 액세스 요청을 상기 기능 블록에 제공하고, 상기 기능 블록은 상기 PCI 호환 액세스 요청에 기초하여 상기 PCI 호환 장치와 통신하는 방법.

**청구항 12**

제10항에 있어서,  
 운영 체제가 상기 가상 장치를 이용하여 상기 PCI 호환 장치로부터 부팅되는 방법.

**청구항 13**

삭제

**청구항 14**

제10항에 있어서,  
 가상 PCI 장치로서 상기 가상 장치를 형성하는 단계는  
 상기 가상 장치에 대한 판매자 식별자를 결정하는 단계;  
 상기 가상 장치에 대한 장치 정보를 결정하는 단계; 및  
 상기 가상 장치에 대한 주소 정보를 결정하는 단계  
 를 포함하는 방법.

**청구항 15**

제14항에 있어서,  
 상기 PCI 호환 장치에 대한 상기 요청에 응답하여 상기 가상 장치에 대한 상기 판매자 식별자, 상기 가상 장치에 대한 상기 장치 정보 및 상기 장치에 대한 상기 주소 정보를 전송하는 단계를 더 포함하는 방법.

**청구항 16**

제10항에 있어서,  
 가상 I/O 장치로서 상기 가상 장치를 형성하는 단계는  
 상기 PCI 호환 장치에 대한 상기 PCI 요청을 폐기하는 단계; 및  
 상기 가상 장치에 대한 I/O 주소 범위를 결정하는 단계  
 를 포함하는 방법.

**청구항 17**

제10항에 있어서,  
 I/O 주소에 대한 메모리 액세스를 수신하는 단계;  
 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련되는지를 결정하는 단계; 및  
 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련된다는 결정 시에 상기 I/O 주소에 대응하는 상기 PCI 장치와 관련된 PCI 주소를 결정하는 단계  
 를 더 포함하는 방법.

**청구항 18**

제10항에 있어서,  
 상기 요청은 상기 PCI 호환 장치에 대한 PCI 열거 요청인 방법.

**청구항 19**

명령어들을 포함하는 비일시적인 머신 판독가능한 저장 매체로서,  
 상기 명령어들은 컴퓨팅 시스템에 의해 실행될 때, 상기 컴퓨팅 시스템으로 하여금,  
 PCI(Peripheral Controller Interconnect) 호환 장치에 대한 PCI 요청을 수신하는 동작;  
 처리 장치에 의해, 상기 PCI 호환 장치의 관련을 결정하는 동작;

상기 관련이 PCI 호환형이 아닌 기능 블록과의 관련이라고 결정하는 것에 응답하여, 상기 처리 장치에 의해, 상기 PCI 호환 장치에 대한 가상 장치를 형성하는 동작 - 상기 가상 장치는 가상 PCI 장치로서 형성되고, 상기 기능 블록은 상기 처리 장치에 통신가능하게 결합되는 버스와의 상기 처리 장치의 통신들을 핸들링하는 회로를 포함함 - ; 및

상기 관련이 상기 PCI 호환 장치로부터 부팅될 소프트웨어 컴포넌트와의 관련이라고 결정하는 것에 응답하여, 상기 처리 장치에 의해, 상기 PCI 호환 장치에 대한 가상 장치를 형성하는 동작 - 상기 가상 장치는 가상 I/O 장치로서 형성됨 -

을 포함하는 동작들을 수행하게 하는 비밀시적인 머신 관독가능한 저장 매체.

**청구항 20**

제19항에 있어서,

상기 PCI 호환 장치는 소프트웨어 드라이버와 관련되고, 상기 소프트웨어 드라이버는 PCI 호환 액세스 요청을 상기 가상 장치로 전송하고, 상기 가상 장치는 상기 PCI 호환 액세스 요청을 상기 기능 블록에 제공하고, 상기 기능 블록은 상기 PCI 호환 액세스 요청에 기초하여 상기 PCI 호환 장치와 통신하는 비밀시적인 머신 관독가능한 저장 매체.

**청구항 21**

제19항에 있어서,

운영 체제가 상기 가상 장치를 이용하여 상기 PCI 호환 장치로부터 부팅되는 비밀시적인 머신 관독가능한 저장 매체.

**청구항 22**

삭제

**청구항 23**

제19항에 있어서,

가상 PCI 장치로서 상기 가상 장치를 형성하는 동작은,  
 상기 가상 장치에 대한 판매자 식별자를 결정하는 동작;  
 상기 가상 장치에 대한 장치 정보를 결정하는 동작; 및  
 상기 가상 장치에 대한 주소 정보를 결정하는 동작  
 을 포함하는 비밀시적인 머신 관독가능한 저장 매체.

**청구항 24**

제19항에 있어서,

가상 I/O 장치로서 상기 가상 장치를 형성하는 동작은,  
 상기 PCI 호환 장치에 대한 상기 PCI 요청을 폐기하는 동작; 및  
 상기 가상 장치에 대한 I/O 주소 범위를 결정하는 동작  
 을 포함하는 비밀시적인 머신 관독가능한 저장 매체.

**청구항 25**

제19항에 있어서,

상기 동작들은  
 I/O 주소에 대한 메모리 액세스를 수신하는 동작;

상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련되는지를 결정하는 동작;

상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련된다는 결정 시에 상기 I/O 주소에 대응하는 상기 PCI 장치와 관련된 PCI 주소를 결정하는 동작

을 더 포함하는 비밀시적인 머신 판독가능한 저장 매체.

**발명의 설명**

**기술 분야**

[0001] 본 명세서에서 설명되는 실시예들은 일반적으로 처리 장치에 관한 것으로서, 구체적으로는 가상 PCI 장치 및 가상 MMIO 장치를 형성하기 위한 일반적인 방법에 관한 것이다.

**배경 기술**

[0002] 컴퓨팅 시스템은 처리 장치(예로서, 마이크로컨트롤러, 마이크로프로세서 등), 메모리 블록, 타이밍 소스, 주변 장치, 외부 인터페이스, 아날로그 인터페이스, 전압 조절기, 전력 관리 회로 등과 같은 다수의 컴포넌트를 포함할 수 있다. 메모리 블록들, 외부 인터페이스들 또는 다른 컴포넌트들은 처리 장치와 주변 장치 사이의 인터페이스를 제공하는 IP 블록들 또는 제어기들로도 알려진 기능 블록들일 수 있다.

[0003] 컴퓨팅 시스템 내의 컴포넌트들은 컴퓨팅 시스템에 의해 액세스 가능한 소프트웨어와 그리고 주변 장치들과 상이한 방식들로, 예로서 주변 컴포넌트 상호접속(PCI) 버스를 통해 통신할 수 있다. 소프트웨어는 플래시 메모리와 같은 일부 주변 장치들로부터 부팅된 후에 컴퓨팅 시스템 내의 처리 장치에 의해 실행될 수 있다.

[0004] 그러나, 컴퓨팅 시스템 내의 일부 컴포넌트들, 예로서 기능 블록들은 PCI 호환형이 아닐 수 있으며, 따라서 통신을 위해 PCI 버스를 사용하는 소프트웨어와 통신하지 못할 수 있다. 더욱이, 일부 소프트웨어는 PCI 호환 주변 장치들과 같은 소정의 주변 장치들로부터 부팅되지 못한다.

[0005] 이러한 문제들을 해결하기 위해 다수의 해법이 이용되었다. 기능 블록이 통신을 위해 PCI 버스를 사용하는 소프트웨어와 통신하지 못하는 문제를 해결하기 위한 현재의 접근법들은 하드웨어 접근법 및 소프트웨어 접근법을 포함한다. 현재의 하드웨어 접근법은 기능 블록을 PCI 호환형이 되게 하기 위해 기능 블록의 하드웨어 설계를 변경하는 것을 필요로 한다. 현재의 소프트웨어 접근법은 기능 블록의 비호환성을 해결할 수 있는 기능 블록에 대한 특정 장치 드라이버와 같은 기능 블록에 대한 소프트웨어를 생성하는 것을 필요로 한다. 소프트웨어가 PCI 호환 주변 장치로부터 부팅되지 못하는 문제를 해결하기 위한 현재의 접근법은 PCI 호환 주변 장치의 하드웨어 구성의 변경(예로서, 실리콘 변경)을 필요로 한다. 그러나, 이러한 접근법들은 상당한 양의 시간 및 비용을 필요로 할 수 있다.

**도면의 간단한 설명**

[0006] 본 개시 내용은 아래에 주어지는 상세한 설명으로부터 그리고 본 개시 내용의 다양한 실시예들의 첨부 도면들로부터 더 충분히 이해될 것이다. 그러나, 도면들은 설명 및 이해를 위한 것일 뿐이며, 본 개시 내용을 특정 실시예들로 한정하는 것으로 간주되지 않아야 한다.

도 1은 가상 장치를 가상 PCI 장치 또는 가상 I/O 장치로서 형성하기 위한 일반 방법을 구현하는 처리 장치의 일 실시예의 블록도이다.

도 2는 본 개시 내용의 일 실시예에 따른, 가상 장치를 가상 PCI 장치 또는 가상 I/O 장치로서 형성하기 위한 일반 방법을 구현하기 위한 가상 장치 모듈을 나타내는 블록도이다.

도 3은 본 개시 내용의 일 실시예에 따른, 가상 장치를 PCI 장치 또는 가상 I/O 장치로서 형성하기 위한 방법을 나타내는 흐름도이다.

도 4는 본 개시 내용의 일 실시예에 따른, 가상 장치를 PCI 장치로서 형성하기 위한 방법을 나타내는 흐름도이다.

도 5는 본 개시 내용의 일 실시예에 따른, 가상 장치를 가상 I/O 장치로서 형성하기 위한 방법을 나타내는 흐름도이다.

도 6은 본 개시 내용의 일 실시예에 따른, 가상 장치를 이용하기 위한 방법을 나타내는 흐름도이다.

도 7은 본 개시 내용의 일 실시예에 따른 컴퓨팅 시스템의 블록도이다.

도 8은 본 개시 내용의 다른 실시예에 따른 컴퓨팅 시스템의 블록도이다.

도 9는 본 개시 내용의 다른 실시예에 따른 컴퓨팅 시스템의 블록도이다.

**발명을 실시하기 위한 구체적인 내용**

- [0007] 컴퓨팅 시스템의 운영 체제 또는 운영 체제 상에서 실행되는 애플리케이션과 같은 소프트웨어는 드라이버를 이용하여 컴퓨팅 시스템 내의 하드웨어 장치(예로서, 주변 장치)에 액세스할 수 있다. 드라이버가 하드웨어 장치에 대한 액세스를 요청할 때, 드라이버는 하드웨어 장치와 관련된 기능 블록을 통해 액세스 요청을 전송할 수 있다. 이어서, 기능 블록은 하드웨어 장치에 액세스하고, 하드웨어 장치로부터 응답을 수신하고, 응답을 드라이버로 전송할 수 있다. 따라서, 기능 블록은 드라이버(및 대응하는 소프트웨어)와 하드웨어 장치 사이의 인터페이스로서 역할한다. 기능 블록은 드라이버 및/또는 하드웨어 장치에 투명할 수 있다. 드라이버는 하드웨어 장치에 대한 커맨드들(예로서, 판독 또는 기록과 같은 액세스 커맨드들)을 컴퓨팅 시스템 내의 주변 컴포넌트 상호접속(PCI) 버스에 기록함으로써 하드웨어 장치에 대한 액세스를 요청할 수 있다. 그러나, 하드웨어 장치에 대응하는 기능 블록이 PCI 호환형이 아닌 경우, 기능 블록은 드라이버가 하드웨어 장치에 액세스하려고 시도하고 있다는 것을 알지 못할 수 있거나, 드라이버와 통신하지 못할 수 있다.
- [0008] 더욱이, 운영 체제는 입출력(I/O) 장치들과 같은 일부 하드웨어 장치들로부터만 부팅될 수 있으며, 다른 하드웨어 장치들로부터는 부팅되지 못할 수 있다. 예를 들어, 운영 체제는 운영 체제가 이동식 장치로서 간주하는 주변 장치, 예로서 PCI 호환 장치로부터 부팅되지 못할 수 있다. 그러나, 컴퓨팅 시스템은 하드웨어 장치를 이동식으로 간주하지 않을 수 있으며, 운영 체제가 하드웨어 장치로부터 부팅되기를 원할 수 있다.
- [0009] 본 개시 내용의 실시예들은 가상 장치를 PCI 장치 또는 가상 I/O 장치로서 형성하기 위한 일반 방법을 제공한다. 일 실시예에서, 본 개시 내용의 방법은 PCI 호환 장치에 대한 요청을 수신하는 단계를 포함한다. 이 방법은 PCI 호환 장치에 대한 요청에 기초하여 가상 장치를 형성하는 단계를 더 포함하고, 가상 장치는 PCI 장치 또는 가상 I/O 장치 중 적어도 하나로서 형성된다.
- [0010] 가상 장치는 PCI 호환형이 아닌 기능 블록에 대한 PCI 장치로서 형성될 수 있으며, 가상 장치는 PCI 호환 장치로부터 부팅될 운영 체제와 관련되는 PCI 호환 장치에 대한 가상 I/O 장치로서 형성될 수 있다. 가상 장치를 PCI 호환형이 아닌 기능 블록에 대한 PCI 장치로서 형성함으로써, 기능 블록과 관련된 PCI 호환 장치에 대해 PCI 버스 상에서 소프트웨어에 의해 전송되는 임의의 액세스 액세스들 및/또는 요청들이 가상 장치에 의해 인식될 수 있고, 기능 블록으로 전송될 수 있다. 따라서, 가상 장치는 PCI 호환 액세스들 및/또는 요청들을 수신하기 위해 PCI 호환형이 아닌 기능 블록에 대한 인터페이스를 제공한다. 가상 장치를 PCI 호환 장치로부터 부팅될 운영 체제와 관련되는 PCI 호환 장치에 대한 가상 I/O 장치로서 형성함으로써, 운영 체제는 가상 장치로부터 부팅될 수 있는데, 그 이유는 운영 체제가 그가 부팅되는 장치가 PCI 호환 장치가 아니라 I/O 장치인 것으로 결정할 것이기 때문이다.
- [0011] 도 1은 본 개시 내용의 일 실시예에 따른, PCI 장치 및 가상 MMIO 장치를 형성하기 위한 일반 방법을 구현하는 장치(100)의 블록도이다. 장치(100)의 일부 예들은 셀룰러 핸드셋 또는 스마트폰과 같은 이동 통신 장치, 태블릿 컴퓨터, 넷북, 노트북 컴퓨터, 랩탑 컴퓨터와 같은 이동 컴퓨팅 장치, 데스크탑 컴퓨터, 서버 컴퓨터 등을 포함할 수 있지만 이에 한정되지 않는다.
- [0012] 장치(100)는 예를 들어 장치(100)에 대한 기본 동작들을 처리하기 위한 호스트(105)를 포함할 수 있다. 호스트(105)는 예를 들어 처리 모듈(110), 기능 블록들(115), 메모리 모듈(120) 및 다른 모듈들(135)을 포함할 수 있다. 처리 모듈(110)은 개별 컴포넌트 내에 위치하는 (처리 장치들로도 알려진) 하나 이상의 프로세서, 또는 대안으로서 예를 들어 시스템-온-칩(SOC) 구성으로 배열되는 단일 집적 회로(IC) 내에 구현되는 하나 이상의 처리 코어를 포함할 수 있다.
- [0013] 기능 블록들(115)은 처리 모듈(110)을 지원하도록 구성되는 회로를 포함할 수 있다. 기능 블록들(115)은 인터페이스/브리징 회로를 포함할 수 있다. 일 실시예에서, 각각의 기능 블록(115)은 장치(100) 내의 특정 버스(예로서, PCI, 직렬 AT 부착(SATA), 유니버설 직렬 버스(USB) 등) 또는 인터페이스(예로서, 멀티미디어 카드(MMC) 장치, 내장형 멀티미디어 카드(eMMC) 장치, 보안 디지털(SD) 장치 등)와 관련된 통신들을 처리하도록 구성되는 집적 회로(IC)이다. 예를 들어, 장치(100)가 PCI, SATA, USB, MMC, eMMC 및 SD 장치들에 대한 버스 및/또는

인터페이스를 포함하는 경우, 장치(100)는 PCI 제어기인 기능 블록(115)(예로서, 제어기), SATA 제어기인 기능 블록(115), USB 제어기인 기능 블록(115), MMC 제어기인 기능 블록(115), eMMC 제어기인 기능 블록(115) 및 SD 제어기인 기능 블록(115)을 포함할 것이다. 기능 블록(115)은 하나의 통신 타입/속도를 다른 통신 타입/속도로 변환함으로써 다양한 모듈들 간의 시그널링을 처리할 수 있다. 각각의 기능 블록(115)은 상이한 시스템 구현, 업그레이드 등을 가능하게 하기 위해 다양한 상이한 장치들과 호환될 수도 있다. 기능 블록들(115)의 기능의 일부는 처리 모듈(110), 메모리 모듈(120) 또는 다른 모듈들(135) 내에 통합될 수도 있다.

[0014] 처리 모듈(110)은 명령어들을 실행할 수 있다. 명령어들은 처리 모듈(110)로 하여금 데이터 판독, 데이터 기록, 데이터 처리, 데이터 형성, 데이터 변환, 데이터 변형 등과 같은, 그러나 이에 한정되지 않는 활동들을 수행하게 하기 위한 프로그램 코드를 포함할 수 있다. 명령어들, 데이터 등을 포함하는 정보(도시되지 않음)가 메모리 모듈(120) 내에 저장될 수 있다.

[0015] 메모리 모듈(120)은 고정 또는 이동 포맷의 랜덤 액세스 메모리(RAM) 또는 판독 전용 메모리(ROM)를 포함할 수 있다. RAM은 장치(100)의 동작 동안 정보를 유지하기 위한 메모리, 예로서 정적 RAM(SRAM) 또는 동적 RAM(DRAM)을 포함할 수 있다. ROM은 장치(100)가 작동할 때 명령어를 제공하기 위한 컴퓨팅 장치 BIOS 메모리, 전자 프로그래밍 가능 ROM들(EPROM들)과 같은 프로그래밍 가능 메모리들, 플래시 등과 같은 메모리들을 포함할 수 있다. 다른 고정 및/또는 이동식 메모리는 플로피 디스크, 하드 드라이브 등과 같은 자기 메모리들, 반도체 플래시 메모리(예로서, eMMC 등), 이동식 메모리 카드 또는 스틱(예로서, USB, 마이크로-SD 등)과 같은 전자 메모리들, 콤팩트 디스크 ROM(CD-ROM), 홀로그래픽과 같은 광학 메모리들 등을 포함할 수 있다.

[0016] 다른 모듈들(135)은 장치(100) 내의 다른 기능의 지원과 관련된 모듈들을 포함할 수 있다. 다른 모듈들(135)은 예를 들어 장치(100)에 전력을 공급하기 위한 모듈들, 장치(100) 내의 유선 및/또는 무선 통신들을 지원하기 위한 모듈들, 장치(100) 내의 사용자 인터페이스(UI) 특징들을 제공하기 위한 모듈들, 특수 기능을 지원하기 위한 모듈들 등을 포함할 수 있다. 다른 모듈들(135)의 구성은 예를 들어 폼 팩터, 장치(100)의 구성 용도 등에 따라 바뀔 수 있다.

[0017] 주변 장치들(140)은 PCI 호환 주변 장치, 메모리 맵핑 I/O(MMIO) 주변 장치와 같은 이동식 또는 비이동식 주변 장치들, 플로피 디스크, 하드 드라이브 등과 같은 자기 메모리들, 반도체 플래시 메모리(예로서, eMMC 등), 이동식 메모리 카드 또는 스틱(예로서, USB, 마이크로-SD 등)과 같은 전자 메모리들, 콤팩트 디스크 기반 ROM(CD-ROM), 홀로그래픽 메모리와 같은 광학 메모리들 등을 포함할 수 있다. 주변 장치는 그가 부착되는 버스 번호에 의해 그리고 주변 장치의 타입에 대한 장치 번호에 의해 식별될 수 있다. 주변 장치(140)는 주변 장치(140) 내에 저장된 하나 이상의 소프트웨어 컴포넌트(145)(예로서, 애플리케이션, 운영 체제 등)를 포함할 수 있다.

[0018] 메모리 모듈(120)의 일 실시예는 가상 장치 모듈(125) 및 하나 이상의 소프트웨어 컴포넌트(130)를 포함할 수 있다. 소프트웨어 컴포넌트들(130)은 애플리케이션들, 운영 체제, BIOS, 시스템 관리 인터럽트(SMI) 핸들러 등을 포함할 수 있다. 일 실시예에서, 소프트웨어 컴포넌트(130)는 주변 장치(140)에 대한 요청(예로서, 열거 요청, 액세스 요청 등)을 가상 장치 모듈(125)로 전송한다. 요청은 주변 장치(140)에 대한 정보, 예로서 주변 장치(140)에 대한 판매키 식별자, 장치 식별자 및 주소 정보를 획득하기 위한 요청일 수 있다. 정보가 가상 장치 모듈(125)에 의해 반환되는 경우, 소프트웨어 컴포넌트(130)는 정보를 이용하여 PCI를 통해 주변 장치(140)와 통신할 수 있다.

[0019] 가상 장치 모듈(125)은 주변 장치(140)에 대한 요청을 수신할 수 있다. 가상 장치 모듈(125)은 주변 장치(140)에 대한 요청에 기초하여 가상 장치를 형성할 수 있다. 가상 장치는 가상 PCI 장치 또는 가상 MMIO 장치일 수 있다.

[0020] 가상 장치 모듈(125)은 가상 장치 모듈(125)이 요청 내의 주변 장치(140)가 PCI 호환형이 아닌 기능 블록(115)과 관련된 것으로 결정하는 경우에 가상 PCI 장치인 가상 장치를 형성할 수 있다. 기능 블록(115)은 기능 블록(115)이 PCI 버스에 대해 액세스 요청들을 판독 또는 기록할 수 없는 경우에는 PCI 호환형이 아니다. 일 실시예에서, 가상 장치 모듈(125)은 기능 블록(115)에 대해 액세스 요청을 판독 또는 기록함으로써 기능 블록(115)이 PCI 호환형이 아닌지를 결정한다. 기능 블록(115)이 액세스 요청을 판독 또는 기록할 수 없다는 예러 또는 기타 지시를 기능 블록(115)이 반환하는 경우, 가상 장치 모듈(125)은 기능 블록(115)이 PCI 호환형이 아닌 것으로 결정할 수 있다. 대안 실시예에서, 가상 장치 모듈(125)은 기능 블록(115)으로부터 호환성 정보를 획득하고 기능 블록(115)에 대한 호환성 정보가 PCI를 포함하는지를 결정함으로써 기능 블록(115)이 PCI 호환형이 아닌지를 결정한다. 다른 실시예에서, 가상 장치 모듈(125)은 장치(100)에 대한 하드웨어 사양으로부터 호환성 정보를 획득함으로써 기능 블록(115)이 PCI 호환형이 아닌지를 결정한다. 또 다른 실시예에서, 가상 장치 모듈

(125)은 SMI 핸들러(도시되지 않음), BIOS(도시되지 않음) 등으로부터 호환성 정보를 획득함으로써 기능 블록(115)이 PCI 호환형이 아닌지를 결정한다. 기능 블록(115)에 대한 호환성 정보가 PCI를 포함하지 않는 경우, 기능 블록(115)은 PCI 호환형이 아니다. 일 실시예에서, 가상 장치 모듈(125)은 가상 PCI 장치에 대한 정보(예로서, 판매자 식별자, 장치 식별자 및 주소 정보)를 결정함으로써 가상 장치를 가상 PCI 장치로서 형성한다. 일부 실시예들에서, 가상 장치 모듈(125)은 소프트웨어 컴포넌트(130)로부터의 요청에 응답하여 정보를 소프트웨어 컴포넌트(130)로 전송한다.

[0021] 가상 장치 모듈(125)은 가상 장치 모듈(125)이 요청 내의 주변 장치(140)가 주변 장치(140)로부터 부팅될 소프트웨어 컴포넌트(145)와 관련된 것으로 결정하는 경우에 가상 MMIO 장치인 가상 장치를 형성할 수 있다. 일 실시예에서, 가상 장치 모듈(125)은 주변 장치(140) 상에 저장된 하나 이상의 소프트웨어 컴포넌트(145)에 액세스하고, 소프트웨어 컴포넌트들 중 임의의 소프트웨어 컴포넌트가 사전 정의된 소프트웨어 컴포넌트(예로서, 운영 체제 등)인지를 결정함으로써 주변 장치(140)가 주변 장치로부터 부팅될 소프트웨어 컴포넌트(145)와 관련되는지를 결정한다. 일 실시예에서, 가상 장치 모듈(125)은 소프트웨어 컴포넌트(130)로부터 수신된 요청을 폐기하고, 가상 I/O 장치에 대한 I/O 주소 범위를 결정함으로써 가상 장치를 가상 I/O 장치로서 형성한다.

[0022] 가상 장치가 주변 장치(140)에 대해 형성되면, 가상 장치 모듈(125)은 가상 장치를 메모리 모듈(120)과 같은 메모리 내에 저장할 수 있다. 일부 실시예들에서, 가상 장치가 주변 장치(140)에 대한 가상 PCI 장치인 경우, 가상 장치 모듈(125)은 소프트웨어 컴포넌트들(130) 중 하나 이상(예로서, 주변 장치(140)에 대한 요청을 전송한 소프트웨어 컴포넌트(130))에 생성된 가상 장치에 대한 판매자 식별자, 장치 식별자 및 주소 정보를 제공한다. 일부 실시예들에서, 가상 장치가 주변 장치(140)에 대한 가상 I/O 장치인 경우, 소프트웨어 컴포넌트들(130)은 요청이 전송된 주변 장치(140)에 더 이상 직접 액세스하지 않는 대신에, 주변 장치(140)에 대한 가상 PCI 장치 또는 가상 I/O 장치에 액세스한다. 가상 장치 모듈(125)은 또한 소프트웨어 컴포넌트들(130)과 가상 장치 사이에 인터페이스를 제공할 수 있다. 일부 실시예들에서, 소프트웨어 컴포넌트(130)로부터 I/O 주소에 대한 액세스를 수신할 때, 가상 장치 모듈(125)은 I/O 주소가 가상 I/O 장치인 가상 장치의 범위 내에 있는지를 결정한다. 이러한 실시예들에서, 가상 장치 모듈(125)이 I/O 주소가 가상 I/O 장치인 가상 장치의 범위 내에 있는 것으로 결정하는 경우, 가상 장치 모듈(125)은 가상 I/O 장치와 관련된 주변 장치(140)에 액세스 요청을 제공하기 전에 I/O 주소를 PCI 주소로 변환한다.

[0023] 도 2는 본 개시 내용의 일 실시예에 따른, 가상 PCI 장치 및 가상 MMIO 장치를 형성하기 위한 일반 방법을 구현하기 위한 가상 장치 모듈(200)을 나타낸다. 일 실시예에서, 가상 장치 모듈(200)은 도 1과 관련하여 기술한 가상 장치 모듈(125)과 동일하다. 가상 장치 모듈(200)은 가상 장치 결정 모듈(205), 가상 PCI 장치 생성 모듈(210), 가상 I/O 장치 생성 모듈(215) 및 가상 I/O 장치 주소 변환 모듈(220)을 포함할 수 있다. 일반성의 손실 없이 더 많거나 적은 컴포넌트가 가상 장치 모듈(200) 내에 포함될 수 있다.

[0024] 가상 장치 결정 모듈(205)은 주변 장치에 대한 요청을 수신할 수 있다. 요청은 요청이 액세스 요청, 열거 요청 등인지와 같은 요청에 대한 식별 정보를 포함할 수 있다. 요청은 주변 장치에 대한 주소와 같은 주변 장치에 대한 식별 정보를 더 포함할 수 있다. 가상 장치 모듈(125)은 주변 장치에 대한 요청에 기초하여 가상 장치를 가상 PCI 장치 또는 가상 I/O 장치로서 형성할지를 결정할 수 있다.

[0025] 가상 장치 결정 모듈(205)은 요청 내의 주변 장치가 PCI 호환형이 아닌 기능 블록과 관련된 경우에 가상 장치가 가상 PCI 장치로서 형성되어야 하는 것으로 결정할 수 있다. 일 실시예에서, 가상 장치 결정 모듈(205)은 기능 블록으로부터 호환성 정보를 획득하고, 기능 블록에 대한 호환성 정보가 PCI를 포함하는지를 결정함으로써 요청 내의 주변 장치가 PCI 호환형이 아닌 기능 블록과 관련되는지를 결정한다. 이 실시예에서, 기능 블록에 대한 호환성 정보가 PCI를 포함하지 않는 경우, 가상 장치 결정 모듈(205)은 기능 블록이 PCI 호환형이 아닌 것으로 결정한다. 이 실시예에서, 기능 블록에 대한 호환성 정보가 PCI를 포함하는 경우, 가상 장치 결정 모듈(205)은 기능 블록이 PCI 호환형인 것으로 결정한다. 대안 실시예에서, 가상 장치 결정 모듈(205)은 SMI 핸들러(미도시), 트랩 핸들러(미도시) 또는 인터럽트 핸들러(미도시)로부터 정보를 획득함으로써 요청 내의 주변 장치가 PCI 호환형이 아닌 기능 블록과 관련되는지를 결정한다. 주변 장치가 PCI 호환형이 아닌 기능 블록과 관련되는 경우, 가상 장치 결정 모듈(205)은 가상 PCI 장치를 가상 장치로서 형성하기 위한 요청을 가상 PCI 장치 생성 모듈(210)로 전송할 수 있다.

[0026] 가상 장치 결정 모듈(205)은 요청 내의 주변 장치가 주변 장치로부터 부팅될 소프트웨어(예로서, 운영 체제)와 관련된 경우에 가상 장치가 가상 I/O 장치로서 형성되어야 하는 것으로 결정할 수 있다. 일 실시예에서, 가상 장치 결정 모듈(205)은 주변 장치 상에 저장된 소프트웨어에 액세스하고, 임의의 소프트웨어가 사전 정의된 타

입의 소프트웨어(예로서, 운영 체제 등)인지를 결정함으로써 주변 장치가 주변 장치로부터 부팅될 소프트웨어와 관련되는지를 결정한다. 대안 실시예에서, 가상 장치 결정 모듈(205)은 SMI 핸들러, 트랩 핸들러 또는 인터럽트 핸들러로부터 정보를 획득함으로써 주변 장치가 주변 장치로부터 부팅될 소프트웨어와 관련되는지를 결정한다. 요청 내의 주변 장치가 주변 장치로부터 부팅될 소프트웨어와 관련되는 경우, 가상 장치 결정 모듈(205)은 가상 I/O 장치를 가상 장치로서 생성하기 위한 요청을 가상 I/O 장치 생성 모듈(215)로 전송할 수 있다.

[0027] 가상 PCI 장치 생성 모듈(210)은 가상 장치 결정 모듈(205)로부터 가상 PCI 장치를 생성하기 위한 요청을 수신할 수 있다. 가상 PCI 장치 생성 모듈(210)은 주변 장치와 관련된 기능 블록에 기초하여 가상 PCI 장치에 대한 식별 정보 및 주소 정보를 결정함으로써 가상 PCI 장치를 생성할 수 있다. 식별 정보는 판매자 식별자, 장치 식별자 등을 포함할 수 있다. 주소 정보는 가상 PCI 장치에 액세스하기 위해 소프트웨어(미도시)에 의해 사용될 수 있는 주소 범위를 포함할 수 있다. 일 실시예에서, 가상 PCI 장치 생성 모듈(210)은 SMI 핸들러, 트랩 핸들러 또는 인터럽트 핸들러로부터 식별 정보 및 주소 정보를 획득한다. 가상 PCI 장치 생성 모듈(210)이 가상 PCI 장치를 생성하면, 가상 PCI 장치 생성 모듈(210)은 가상 PCI 장치를 메모리에 저장할 수 있다. 일 실시예에서, 가상 PCI 장치 생성 모듈(210)은 가상 PCI 장치를 메모리 모듈(250) 내의 가상 장치 정보(255) 내에 가상 장치로서 저장한다.

[0028] 가상 I/O 장치 생성 모듈(215)은 가상 장치 결정 모듈(205)로부터 가상 I/O 장치를 생성하기 위한 요청을 수신할 수 있다. 가상 I/O 장치 생성 모듈(215)은 주변 장치에 대한 요청을 폐기하고, 가상 I/O 장치에 대한 I/O 주소 범위를 결정함으로써 가상 I/O 장치를 생성할 수 있다. 일 실시예에서, 가상 I/O 장치 생성 모듈(215)은 주변 장치에 대한 요청에 응답하지 않으므로써 주변 장치에 대한 요청을 폐기한다. 대안 실시예에서, 가상 I/O 장치 생성 모듈(215)은 요청이 실패했다(예로서, PCI 판독에 실패했다)는 요청에 대한 응답을 전송함으로써 주변 장치에 대한 요청을 폐기한다. 가상 I/O 장치 생성 모듈(215)이 가상 I/O 장치를 생성하면, 가상 I/O 장치 생성 모듈(215)은 가상 I/O 장치를 메모리에 저장할 수 있다. 일 실시예에서, 가상 I/O 장치 생성 모듈(215)은 가상 I/O 장치를 메모리 모듈(250) 내의 가상 장치 정보(255) 내에 저장한다.

[0029] 가상 I/O 장치 주소 변환 모듈(220)은 I/O 주소에 대한 요청 또는 액세스를 수신할 수 있다. 일 실시예에서, 요청 또는 액세스는 소프트웨어로부터 수신된다. 요청 또는 액세스에 응답하여, 가상 I/O 장치 주소 변환 모듈(220)은 I/O 주소가 가상 I/O 장치인 가상 장치의 주소 범위 내에 있는지를 결정할 수 있다. 가상 I/O 장치 주소 변환 모듈(220)은 I/O 주소를 가상 I/O 장치인 각각의 가상 장치에 대한 주소 범위를 비교함으로써 I/O 주소가 가상 I/O 장치의 주소 범위 내에 있는지를 결정할 수 있다. 일 실시예에서, 가상 I/O 장치 주소 변환 모듈(220)은 I/O 주소를 가상 장치 정보(255) 내의 주소 범위들과 비교한다. 가상 I/O 장치 주소 변환 모듈(220)이 I/O 주소가 가상 I/O 장치인 가상 장치의 주소 범위 내에 있는 것으로 결정하는 경우, 가상 I/O 장치 주소 변환 모듈(220)은 I/O 주소를 PCI 주소로 변환할 수 있고, 가상 장치와 관련된 주변 장치의 PCI 주소가 액세스되게 할 수 있다.

[0030] 도 3은 본 개시 내용의 일 실시예에 따른, 가상 PCI 장치 및 가상 MMIO 장치를 형성하기 위한 방법(300)의 흐름도이다. 방법(300)은 하드웨어(예로서, 회로, 전용 논리, 프로그래밍 가능 논리, 마이크로코드 등), 소프트웨어(예로서, 처리 장치 상에서 실행되는 명령어들), 펌웨어 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 일 실시예에서, 방법(300)은 도 1과 관련하여 설명된 장치(100)에 의해 수행된다.

[0031] 블록 305에서, 처리 논리는 PCI 호환 장치에 대한 요청을 수신한다. PCI 호환 장치는 PCI 버스에 부착될 수 있는 주변 장치일 수 있다. PCI 호환 장치에 대한 요청은 요청이 액세스 요청, 열거 요청 등인지와 같은 요청에 대한 식별 정보를 포함할 수 있다. 일 실시예에서, 요청은 처리 장치가 시스템 관리 모드(SMM)에서 방법(300)을 수행하고 있는 동안 SMI 핸들러로부터 수신되는 열거 요청이다.

[0032] SMM은 처리 장치의 모든 정상적인 실행(OS 포함)이 중지되고 특수한 개별 소프트웨어(통상적으로 펌웨어 또는 하드웨어 지원 디버거)가 고특권 모드로 실행되는 동작 모드이다. SMM은 격리된 메모리 및 실행 환경을 제공하며, SMM 코드는 운영 체제(OS)에게는 보이지 않지만, 메모리에 대한 완전한 액세스 및 PCI 호환 장치 등과 같은 주변 장치들에 대한 완전한 제어를 유지한다. SMM이 개시될 때, 처리 장치의 현재 상태가 저장되고, 모든 다른 프로세스들이 중단된다. 디버깅, 하드웨어 관리, 보안 기능, 에뮬레이션 등과 같은 고특권 동작들이 SMM 모드에서 수행될 수 있으며, 이어서 처리 장치는 처리 장치의 저장된 상태에 기초하여 동작을 재개할 수 있다. 시스템 관리 인터럽트(SMI)의 발생시에, 처리 장치는 SMM에 들어가서 SMI 핸들러를 실행할 수 있다. SMI는 처리 장치의 실행이 시작(부팅)될 때, 새로운 주변 장치가 장치에 추가될 때 등에 생성될 수 있다. 예를 들어, 펌웨어 또는 BIOS가 부팅시에 SMI를 생성할 수 있다.

- [0033] SMI의 수신시에, SMI 핸들러는 PCI 호환 장치(들)를 결정하기 위해 PCI 버스에 조회(예로서, 판독하려고 시도)함으로써 처리 장치에 의해 이용 가능한 PCI 호환 (주변) 장치들을 열거할 수 있다. SMI가 부팅에 응답하여 생성된 경우, 장치 내의 모든 PCI 호환 장치들이 열거되지는 않았을 수 있다. SMI가 새로운 PCI 호환 장치의 추가에 응답하여 생성되는 경우, 새로운 PCI 호환 장치는 열거되지 않았을 수 있다. SMI 핸들러는 아직 열거되지 않은 각각의 PCI 호환 장치에 대한 열거 요청을 생성할 수 있다. 열거 요청은 PCI 호환 장치에 대한 버스 번호 및 장치 번호와 같은 PCI 호환 장치에 대한 식별 정보를 포함할 수 있다.
- [0034] 도 3을 참조하면, 블록 310에서, 처리 논리는 요청 내의 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련되는지를 결정한다. 기능 블록은 기능 블록이 PCI 버스에 대해 액세스 요청들을 판독 또는 기록할 수 없는 경우에 PCI 호환형이 아니다.
- [0035] 일 실시예에서, 처리 논리는 PCI 호환 장치와 관련된 기능 블록으로부터 호환성 정보를 획득하고, 기능 블록에 대한 호환성 정보가 PCI 호환성을 포함하는지를 결정함으로써 요청 내의 주변 장치가 PCI 호환형이 아닌 기능 블록과 관련되는지를 결정한다. 기능 블록에 대한 호환성 정보가 PCI 호환성을 포함하지 않는 경우, 처리 논리는 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련되지 않은 것으로 결정한다(즉, PCI 호환 장치는 PCI 호환형인 기능 블록과 관련된다). 기능 블록에 대한 호환성 정보가 PCI 호환성을 포함하지 않는 경우, 처리 논리는 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련되는 것으로 결정한다.
- [0036] 대안 실시예에서, 처리 논리는 SMI 핸들러로부터 PCI 호환 장치와 관련된 기능 블록에 대한 호환성 정보를 획득함으로써 주변 장치가 PCI 호환형이 아닌 기능 블록과 관련되는지를 결정한다. 호환성 정보는 PCI 호환 장치와 관련된 기능 블록이 PCI 호환형인지의 여부를 포함할 수 있다. SMI 핸들러는 처리 장치 상에서 실행되고 있거나 실행될 드라이버들 및/또는 소프트웨어(예로서, 애플리케이션들)에 대한 정보를 수집할 수 있으며, 어느 PCI 호환 장치들이 드라이버들 및/또는 소프트웨어에 의해 지원되는지를 결정할 수 있다. SMI 핸들러는 문서로부터, 드라이버 또는 소프트웨어 소스 코드로부터, 기타 등등으로부터 드라이버들 및/또는 소프트웨어에 대한 정보를 수집할 수 있다. 드라이버들 및/또는 소프트웨어에 의해 지원되는 PCI 호환 장치들의 결정시에, SMI 핸들러는 PCI 호환 장치들 각각에 대한 대응하는 기능 블록(예로서, 제어기)을 결정하고, 대응하는 기능 블록이 PCI 호환형인지를 결정할 수 있다. SMI 핸들러는 기능 블록들 각각에 액세스하여, 기능 블록들 각각의 PCI 호환성을 결정할 수 있다. 예를 들어, 기능 블록들 각각에 대한 장치 타입 정보가 SMI 핸들러에 의해 액세스될 수 있다. 일 실시예에서, 처리 논리는 기능 블록에 대한 호환성 정보에 대한 요청을 전송하고, SMI 핸들러로부터 기능 블록에 대한 호환성 정보를 수신함으로써 SMI 핸들러로부터 PCI 호환 장치와 관련된 기능 블록에 대한 호환성 정보를 획득할 수 있다. 대안 실시예에서, 처리 논리는 SMI 핸들러에 의해 기록된 사전 정의된 메모리 위치에 액세스함으로써 SMI 핸들러로부터 기능 블록에 대한 호환성 정보를 획득할 수 있다. 호환성 정보가 기능 블록이 PCI 호환형이 아니라는 호환성 정보를 포함하는 경우, 처리 논리는 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련되는 것으로 결정한다. 기능 블록에 대한 호환성 정보가 기능 블록이 PCI 호환형이라는 정보를 포함하는 경우, 처리 논리는 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련되지 않은 것으로 결정한다(즉, PCI 호환 장치는 PCI 호환형인 기능 블록과 관련된다).
- [0037] 다른 대안 실시예에서, 처리 논리는 요청 내의 주변 장치에 대해 액세스 요청을 판독 또는 기록함으로써 요청 내의 주변 장치가 PCI 호환형이 아닌지를 결정한다. 요청 내의 주변 장치가 액세스 요청을 판독 또는 기록할 수 없다는 에러 또는 기타 지시를 요청 내의 주변 장치가 반환하는 경우, 처리 논리는 요청 내의 주변 장치가 PCI 호환형이 아닌 것으로 결정할 수 있다.
- [0038] 처리 논리가 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련되지 않은 것으로 결정하는 경우, 방법(300)은 블록 320으로 진행한다. 처리 논리가 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련된 것으로 결정하는 경우, 방법(300)은 블록 315로 진행한다.
- [0039] 블록 315에서, 처리 논리는 가상 장치를 가상 PCI 장치로서 형성한다. 처리 논리는 가상 장치에 대한 PCI 식별 정보 및 주소 정보를 결정함으로써 가상 장치를 가상 PCI 장치로서 형성할 수 있다. 가상 장치를 가상 PCI 장치로서 형성하는 일 구현은 아래에서 도 4와 관련하여 설명된다. 일 실시예에서, 가상 장치의 형성시에, 처리 논리는 옵션으로서 블록 305에서 수신된 요청에 대한 응답을 제공할 수 있다. 요청에 대한 응답은 가상 장치와 관련된 사전 결정된 레지스터들의 성공적인 판독을 포함할 수 있으며, 가상 장치에 대한 판매자 식별자, 장치 식별자, I/O 주소 범위 및 MMIO 주소 범위와 같은 가상 장치에 대한 식별 정보를 더 포함할 수 있다. 예를 들어, PCI 호환형이 아닌 범용 입출력(GPIO) 제어기를 갖는 GPIO 장치에 대한 열거 요청은 성공적인 응답을 수신하며, GPIO 장치와 관련된 GPIO 제어기에 대해 생성된 가상 장치에 대한 판매자 식별자(예로서, 0x8888), 장치

식별자(예로서, 0x9999), I/O 주소 범위(예로서, 0x200-0x20F) 및 MMIO 범위(0xA0000-0xA00FF)를 포함할 것이다.

[0040] 블록 320에서, 처리 논리는 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되는지를 결정한다. PCI 호환 장치로부터 부팅될 소프트웨어는 운영 체제, 소프트웨어 애플리케이션, BIOS 등일 수 있다. 일부 실시예들에서, PCI 호환 장치로부터 부팅될 소프트웨어는 메모리-맵핑 I/O 주변 장치와 같은 I/O 장치로부터의 부팅을 필요로 하는 소프트웨어이다. 이러한 실시예들에서, 소프트웨어는 PCI 호환 장치들(예로서, 내장 멀티미디어 메모리 카드(eMMC) 등)과 같은 이동식 장치들로서 간주되는 주변 장치들로부터 부팅되지 않도록 설계될 수 있다. 그러나, 일부 PCI 호환 장치들은 SOC 내의 PCI 장치들과 같이 이동식이 아니다.

[0041] 일 실시예에서, 처리 논리는 PCI 호환 장치에 저장된 소프트웨어 컴포넌트들에 액세스하고, 소프트웨어 컴포넌트들이 사전 정의된 소프트웨어 컴포넌트인지를 결정함으로써 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되는지를 결정한다. 처리 논리는 소프트웨어 컴포넌트를 이동식이 아닌 주변 장치 상에 위치하는 것으로 결정된 하나 이상의 사전 정의된 소프트웨어 컴포넌트(예로서, eMMC 상의 운영 체제 등)와 비교함으로써 소프트웨어 컴포넌트가 사전 정의된 소프트웨어 컴포넌트인지를 결정할 수 있다. 비교가 PCI 호환 장치에 저장된 소프트웨어 컴포넌트들이 사전 결정된 소프트웨어 컴포넌트들 중 하나 이상을 포함한다는 것을 지시하는 경우, 처리 논리는 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되는 것으로 결정할 수 있다. 비교가 PCI 호환 장치에 저장된 소프트웨어 컴포넌트들이 사전 결정된 소프트웨어 컴포넌트들 중 하나 이상을 포함하지 않는다는 것을 지시하는 경우, 처리 논리는 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되지 않은 것으로 결정할 수 있다.

[0042] 대안 실시예에서, 처리 논리는 SMI 핸들러로부터 소프트웨어 정보를 획득함으로써 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되는지를 결정한다. 하나의 그러한 실시예에서, SMI 핸들러로부터 획득되는 소프트웨어 정보는 PCI 호환 장치 상의 소프트웨어 컴포넌트들을 포함한다. 이 실시예에서, SMI 핸들러는 PCI 호환 장치 상에 저장된 드라이버들 및/또는 소프트웨어 컴포넌트들에 대한 정보를 수집할 수 있다. 이 실시예에서, 처리 논리는 소프트웨어 정보를 획득하고, 소프트웨어 정보를 이동식이 아닌 주변 장치 상에 위치하는 것으로 결정된 하나 이상의 사전 정의된 소프트웨어 컴포넌트(예로서, eMMC 상의 운영 체제 등)와 비교할 수 있다. 비교가 소프트웨어 정보가 사전 결정된 소프트웨어 컴포넌트들 중 하나 이상을 포함한다는 것을 지시하는 경우, 처리 논리는 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되는 것으로 결정할 수 있다. 비교가 소프트웨어 정보가 사전 결정된 소프트웨어 컴포넌트들 중 하나 이상을 포함하지 않는다는 것을 지시하는 경우, 처리 논리는 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되지 않은 것으로 결정할 수 있다. 대안적인 그러한 실시예에서, 소프트웨어 정보는 PCI 호환 장치 상의 임의의 소프트웨어 컴포넌트가 PCI 호환 장치로부터 부팅되는지에 대한 지시자(예로서, 1로 설정된 비트와 같은 긍정 지시자 또는 0로 설정된 비트와 같은 부정 지시자)를 포함한다. 이 실시예에서, 처리 논리는 소프트웨어 정보를 획득하고, 지시자에 기초하여 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되는지를 결정한다. 지시자가 PCI 호환 장치 상의 소프트웨어가 PCI 호환 장치로부터 부팅된다는 것을 지시하는 경우(예로서, 긍정 지시자), 처리 논리는 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되는 것으로 결정할 수 있다. 지시자가 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되지 않는다는 것을 지시하는 경우(예로서, 부정 지시자), 처리 논리는 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되지 않은 것으로 결정할 수 있다.

[0043] 일 실시예에서, 처리 논리는 PCI 호환 장치에 대한 소프트웨어 정보에 대한 요청을 전송하고, SMI 핸들러로부터 PCI 호환 장치에 대한 소프트웨어 정보를 수신함으로써 SMI 핸들러로부터 PCI 호환 장치에 대한 소프트웨어 정보를 획득할 수 있다. 대안 실시예에서, 처리 논리는 SMI 핸들러에 의해 기록된 사전 정의된 메모리 위치에 액세스함으로써 SMI 핸들러로부터 PCI 호환 장치에 대한 소프트웨어 정보를 획득할 수 있다.

[0044] 처리 논리가 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련되지 않은 것으로 결정하는 경우, 방법(300)은 가상 장치를 형성하지 않고 종료된다. 처리 논리가 PCI 호환 장치가 PCI 호환 장치로부터 부팅될 소프트웨어와 관련된 것으로 결정하는 경우, 방법(300)은 블록 325로 진행한다.

[0045] 블록 325에서, 처리 논리는 가상 장치를 가상 I/O 장치로서 형성한다. 일 실시예에서, 가상 I/O 장치는 가상 MMIO 장치이다. 가상 장치를 가상 I/O 장치로서 형성하는 일 구현이 도 5와 관련하여 아래에서 설명된다. 일 실시예에서, 가상 장치의 형성시에, 처리 논리는 옵션으로서 블록 305에서 수신된 요청에 대한 응답을 제공할 수 있다. 요청에 대한 응답은 PCI 호환 장치와 관련된 사전 결정된 레지스터들의 성공하지 못한 판독을 포함할

수 있다.

- [0046] 도 4는 본 개시 내용의 일 실시예에 따른, 가상 장치를 가상 PCI 장치로서 형성하기 위한 방법(400)의 흐름도이다. 방법(400)은 하드웨어(예로서, 회로, 전용 논리, 프로그래밍 가능 논리, 마이크로코드 등), 소프트웨어(예로서, 처리 장치 상에서 실행되는 명령어들), 펌웨어 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 일 실시예에서, 방법(400)은 도 1과 관련하여 설명된 장치(100)에 의해 수행된다.
- [0047] 블록 405에서, 처리 논리는 가상 장치에 대한 판매자 식별자를 결정한다. 일 실시예에서, 판매자 식별자는 SMI 핸들러 또는 BIOS로부터 판매자 식별자를 획득함으로써 결정된다. SMI 핸들러 또는 BIOS로부터의 판매자 식별자는 SMI 핸들러 또는 BIOS에 의해 할당될 수 있거나, 드라이버 또는 소프트웨어 컴포넌트에 의해 요구되는 판매자 식별자와 매칭될 수 있다. 예를 들어, 드라이버가 판매자 식별자 "0x8086"을 요청하는 경우, SMI 핸들러 BIOS는 가상 장치에 대해 판매자 식별자 "0x8086"을 할당할 것이다. 대안 실시예에서, 판매자 식별자는 PCI 호환 장치와 관련된 판매자 식별자를 결정하고 가상 장치에 대한 판매자 식별자를 결정된 판매자 식별자와 동일하도록 구성함으로써 결정된다. 예를 들어, PCI 호환 장치가 0x8888의 판매자 식별자를 갖는 GPIO 장치인 경우, 가상 장치에 대한 대응하는 판매자 식별자도 0x8888일 것이다.
- [0048] 블록 410에서, 처리 논리는 가상 장치에 대한 장치 식별자를 결정한다. 일 실시예에서, 장치 식별자는 SMI 핸들러 또는 BIOS로부터 장치 식별자를 획득함으로써 결정된다. SMI 핸들러 또는 BIOS로부터의 판매자 식별자는 SMI 핸들러 또는 BIOS에 의해 할당될 수 있거나, 드라이버 또는 소프트웨어 컴포넌트에 의해 요구되는 판매자 식별자와 매칭될 수 있다. 예를 들어, 드라이버가 장치 식별자 "0x8086"을 요청하는 경우, SMI 핸들러 BIOS는 가상 장치에 대해 장치 식별자 "0x8086"을 할당할 것이다. 대안 실시예에서, 장치 식별자는 PCI 호환 장치와 관련된 장치 식별자를 결정하고 가상 장치에 대한 장치 식별자를 결정된 장치 식별자와 동일하도록 구성함으로써 결정된다. 예를 들어, PCI 호환 장치가 0x9999의 장치 식별자를 갖는 GPIO 장치인 경우, 가상 장치에 대한 대응하는 장치 식별자도 0x9999일 것이다.
- [0049] 블록 415에서, 처리 논리는 가상 장치에 대한 주소 정보를 결정한다. 가상 장치에 대한 주소 정보는 가상 장치에 대한 주소 I/O 범위 및 MMIO 주소 범위일 수 있다. 일 실시예에서, 주소 정보는 SMI 핸들러, BIOS 또는 하드웨어 사양으로부터 주소 정보를 획득함으로써 결정된다. 대안 실시예에서, 주소 정보는 PCI 호환 장치와 관련된 주소 정보를 결정하고 가상 PCI 장치에 대한 주소 정보를 결정된 주소 정보와 동일하도록 구성함으로써 결정된다. 예를 들어, PCI 호환 장치가 0x200 내지 0x20F의 I/O 범위 및 0xA0000 내지 0xA00FF의 MMIO 주소 범위를 포함하는 주소 정보를 갖는 GPIO 장치인 경우, 가상 장치에 대한 대응하는 주소 정보도 0x200 내지 0x20F의 I/O 범위 및 0xA0000 내지 0xA00FF의 MMIO 주소 범위일 것이다. 주소 정보는 가상 장치에 액세스하기 위해 소프트웨어 컴포넌트(미도시)에 의해 사용될 수 있는 가상 장치에 대한 주소 범위를 포함할 수 있다.
- [0050] 도 5는 본 개시 내용의 일 실시예에 따른, 가상 장치를 가상 I/O 장치로서 형성하기 위한 방법(500)의 흐름도이다. 방법(500)은 하드웨어(예로서, 회로, 전용 논리, 프로그래밍 가능 논리, 마이크로코드 등), 소프트웨어(예로서, 처리 장치 상에서 실행되는 명령어들), 펌웨어 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 일 실시예에서, 방법(500)은 도 1과 관련하여 설명된 장치(100)에 의해 수행된다.
- [0051] 블록 505에서, 처리 논리는 PCI 호환 장치에 대한 요청을 폐기한다. 일 실시예에서, 요청은 PCI 호환 장치와 관련된 열거 요청이다. PCI 호환 장치에 대한 요청을 폐기함으로써, 처리 논리는 요청과 관련된 PCI 호환 장치가 존재하지 않는다는 피드백을 요청을 행한 소프트웨어에 제공할 수 있다. 이것은 소프트웨어로 하여금 소프트웨어가 MMIO 장치 상에 저장되거나 소프트웨어가 MMIO 장치에 액세스하고 있는 것으로 생각하게 할 것이다. 일 실시예에서, 처리 논리는 요청에 응답하지 않음으로써 PCI 호환 장치에 대한 요청을 폐기한다. 대안 실시예에서, 처리 논리는 PCI 호환 요청과 관련된 하나 이상의 사전 결정된 레지스터(예로서, 요청 내에 제공된 판매자 식별자 및 장치 식별자와 관련된 PCI 구성 레지스터들)의 성공하지 못한 판독을 포함하는 요청에 대한 응답을 생성함으로써 PCI 호환 장치에 대한 요청을 폐기한다.
- [0052] 블록 510에서, 처리 논리는 가상 I/O 장치에 대한 I/O 주소 범위를 결정한다. 일 실시예에서, I/O 주소 범위는 MMIO 주소 범위이다. 일 실시예에서, 처리 논리는 SMI 핸들러로부터 I/O 주소 범위를 획득함으로써 가상 I/O 장치에 대한 I/O 주소 범위를 결정한다. 대안 실시예에서, 처리 논리는 ACPI(Advanced Configuration and Power Interface) 테이블로부터 PCI 호환 장치에 대한 I/O 주소 범위를 획득(예로서, 파싱)함으로써 가상 I/O 장치에 대한 I/O 주소 범위를 결정한다. ACPI 사양은 운영 체제에 의한 장치 구성 및 전력 관리에 대한 개방 표준을 제공할 수 있다. ACPI 테이블은 시스템 내에서 이용 가능한 장치들에 대한 장치 자원 정보를 포함할 수 있다. 장치에 대한 장치 자원 정보는 장치 이름, MMIO 주소 범위, I/O 주소 범위, 인터럽트 메커니즘, 장치와

관련된 장치 등을 포함할 수 있다.

- [0053] 도 6은 본 개시 내용의 일 실시예에 따른, 가상 장치를 이용하기 위한 방법(600)의 흐름도이다. 방법(600)은 하드웨어(예로서, 회로, 전용 논리, 프로그래밍 가능 논리, 마이크로코드 등), 소프트웨어(예로서, 처리 장치 상에서 실행되는 명령어들), 펌웨어 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 일 실시예에서, 방법(600)은 도 1과 관련하여 설명된 장치(100)에 의해 수행된다.
- [0054] 블록 605에서, 처리 논리는 I/O 주소에 대한 액세스 요청을 수신한다. 일 실시예에서, I/O 주소는 MMIO 주소이다. 일 실시예에서, I/O 주소에 대한 액세스 요청은 소프트웨어로부터 수신된다.
- [0055] 블록 610에서, 처리 논리는 I/O 주소가 가상 I/O 장치인 가상 장치의 주소 범위 내에 있는지를 결정한다. 처리 논리는 I/O 주소를 컴퓨팅 시스템 내의 각각의 가상 I/O 장치에 대한 주소 범위와 비교함으로써 I/O 주소가 가상 I/O 장치인 가상 장치의 주소 범위 내에 있는지를 결정할 수 있다. 처리 논리가 I/O 주소가 가상 I/O 장치인 가상 장치의 주소 범위 내에 있지 않은 것으로 결정하는 경우, 방법(600)이 종료된다. 처리 논리가 I/O 주소가 가상 I/O 장치인 가상 장치의 주소 범위 내에 있는 것으로 결정하는 경우, 방법(600)은 블록 615로 진행한다. 일 실시예에서, 블록 610은 옵션이며, 수행되지 않는다. 이 실시예에서, I/O 주소가 가상 I/O 장치인 가상 장치의 주소 범위 내에 있는지에 대한 결정은 SMI 핸들러에 의해 수행된다.
- [0056] 블록 615에서, 처리 논리는 I/O 주소에 대응하는 PCI 장치의 PCI 주소를 결정한다. 처리 논리는 SMI 핸들러로부터 I/O 주소에 대한 PCI 주소를 획득함으로써 I/O 주소에 대응하는 PCI 장치의 PCI 주소를 결정할 수 있다. 일 실시예에서, 블록 615는 처리 논리가 메모리 공간 트랩을 지원하지 않는 경우에는 옵션이다. 이 실시예에서, 처리 논리는 가상 I/O 장치에 대해 이전에 결정된 가상 I/O 장치에 대한 I/O 주소에 액세스할 것이며, 이는 PCI 장치의 I/O 주소와 동일한 I/O 주소를 노출시킨다.
- [0057] 예를 들어, 시스템이 메모리 공간에서의 트랩을 지원하는 경우, 처리 논리는 가상 I/O 장치인 가상 장치의 모니터링된 주소 범위 내에서 액세스가 발생할 때 SMI를 트리거한다. 이 예에서, SMI 핸들러는 트리거되어, I/O 주소가 가상 장치의 주소 범위 내에 있는지를 결정하고, I/O 주소로부터 PCI 주소로 변환할 것이다.
- [0058] 도 7은 본 개시 내용의 일 실시예에 따른, 가상 PCI 장치 및 가상 MMIO 장치를 형성하기 위한 논리 회로들을 포함하는 SoC(700)의 블록도이다. 점선 박스들은 더 진보된 SoC들 상의 옵션 특징들이다. 도 7에서, 상호접속 유닛(들)(712)은 한 세트의 하나 이상의 코어(702A-N) 및 공유 캐시 유닛(들)(706); 시스템 에이전트 유닛(710); 버스 제어기 유닛(들)(716); 통합 메모리 제어기 유닛(들)(714); 통합 그래픽 논리(708), 정지 및/또는 비디오 카메라 기능을 제공하기 위한 이미지 프로세서(724), 하드웨어 오디오 가속을 제공하기 위한 오디오 프로세서(726) 및 비디오 인코드/디코드 가속을 제공하기 위한 비디오 프로세서(728)를 포함할 수 있는 한 세트 또는 하나 이상의 미디어 프로세서(718); 정적 랜덤 액세스 메모리(SRAM) 유닛(730); 직접 메모리 액세스(DMA) 유닛(732); 및 하나 이상의 외부 디스플레이에 결합하기 위한 디스플레이 유닛(740)을 포함하는 애플리케이션 프로세서(720)에 결합된다.
- [0059] 메모리 계층 구조는 코어들 내의 하나 이상의 캐시 레벨, 한 세트 또는 하나 이상의 공유 캐시 유닛(706), 및 통합 메모리 제어기 유닛(들)(714)의 세트에 결합된 외부 메모리(미도시)를 포함한다. 공유 캐시 유닛(들)(706)의 세트는 하나 이상의 중간 레벨 캐시, 예를 들어 레벨 2(L2), 레벨 3(L3), 레벨 4(L4) 또는 다른 레벨의 캐시, 최종 레벨 캐시(LLC) 및/또는 이들의 조합을 포함할 수 있다.
- [0060] 일부 실시예들에서, 코어들(702A-N) 중 하나 이상은 다중 스레딩이 가능하다.
- [0061] 시스템 에이전트(710)는 코어들(702A-N)을 조정하고 운영하는 컴포넌트들을 포함한다. 시스템 에이전트 유닛(710)은 예를 들어 전력 제어 유닛(PCU) 및 디스플레이 유닛을 포함할 수 있다. PCU는 코어들(702A-N) 및 통합 그래픽 논리(708)의 전력 상태를 조절하는 데 필요한 논리 및 컴포넌트들이거나 이들을 포함할 수 있다. 디스플레이 유닛은 하나 이상의 외부 접속 디스플레이를 구동하기 위한 것이다.
- [0062] 코어들(702A-N)은 아키텍처 및/또는 명령어 세트와 관련하여 동종 또는 이종일 수 있다. 예를 들어, 코어들(702A-N) 중 일부는 순차적일 수 있는 반면, 다른 코어들은 비순차적이다. 다른 예로서, 코어들(702A-N) 중 둘 이상은 동일 명령어 세트를 실행할 수 있는 반면, 다른 코어들은 그러한 명령어 세트의 서브세트만을 또는 상이한 명령어 세트를 실행할 수 있다.
- [0063] 애플리케이션 프로세서(720)는 캘리포니아 산타클라라의 인텔사로부터 입수 가능한 Core(상표) i3, i5, i7, 2 Duo 및 Quad, Xeon(상표), Itanium(상표), XScale(상표) 또는 StrongARM(상표) 프로세서와 같은 범용 프로세서

일 수 있다. 대안으로서, 애플리케이션 프로세서(720)는 ARM Holdings, Ltd., MIPS 등과 같은 다른 회사로부터 입수될 수 있다. 애플리케이션 프로세서(720)는 예를 들어 네트워크 또는 통신 프로세서, 압축 엔진, 그래픽 프로세서, 코-프로세서, 내장 프로세서 등과 같은 특수 목적 프로세서일 수 있다. 애플리케이션 프로세서(720)는 하나 이상의 칩 상에 구현될 수 있다. 애플리케이션 프로세서(720)는 하나 이상의 기관의 일부일 수 있고/있거나, 예를 들어 BiCMOS, CMOS 또는 NMOS와 같은 다수의 프로세스 기술 중 임의의 프로세스 기술을 이용하여 하나 이상의 기관 상에 구현될 수 있다.

[0064] 일 실시예에서, 애플리케이션 프로세서(720)는 본 발명의 실시예들에 따른, 가상 PCI 장치 및 가상 MMIO 장치의 형성을 구현하기 위한 논리도 포함한다. 예를 들어, 애플리케이션 프로세서(720)는 도 1과 관련하여 설명된 가상 장치 모듈(125)과 같은 가상 장치 모듈을 실행하기 위한 논리를 포함할 수 있으며, 가상 장치 모듈은 주변 장치에 대한 요청에 기초하여 가상 장치를 형성할 수 있다. 가상 장치는 가상 PCI 장치 또는 가상 MMIO 장치일 수 있다.

[0065] 도 8은 본 개시 내용에 따른 시스템-온-칩(SOC) 설계의 일 실시예의 블록도이다. 구체적인 설명 예로서, SOC(800)는 사용자 장비(UE) 내에 포함된다. 일 실시예에서, UE는 핸드헬드 전화, 스마트폰, 태블릿, 울트라-씬 노트북, 광대역 어댑터를 갖는 노트북 또는 임의의 다른 유사한 통신 장치와 같이 최종 사용자가 통신에 사용할 수 있는 임의의 장치를 지칭한다. 종종, UE는 사실상 GSM 네트워크 내의 이동국(MS)에 잠재적으로 대응하는 기지국 또는 노드에 접속된다.

[0066] 여기서, SOC(800)는 2개의 코어(806, 807)를 포함한다. 코어들(806, 807)은 Intel(등록상표) Architecture Core(상표) 기반 프로세서, Advanced Micro Devices, Inc.(AMD) 프로세서, MIPS 기반 프로세서, ARM 기반 프로세서 설계 또는 그들의 고객은 물론, 그들의 라이선시들 또는 채택자들과 같은 명령어 세트 아키텍처를 따를 수 있다. 코어들(806, 807)은 시스템(800)의 다른 요소들과 통신하기 위해 버스 인터페이스 유닛(809) 및 L2 캐시(810)와 관련되는 캐시 제어(808)에 결합된다. 상호접속(810)은 설명되는 개시 내용의 하나 이상의 양태를 잠재적으로 구현하는 IOSF, AMBA 또는 전술한 다른 상호접속과 같은 온-칩 상호접속을 포함한다.

[0067] 인터페이스(810)는 가입자 식별 모듈(SIM) 카드와 인터페이스하기 위한 SIM(830), SOC(800)를 초기화 및 부팅하기 위해 코어들(806, 807)에 의해 실행될 부트 코드를 유지하기 위한 부트 ROM(835), 외부 메모리(예로서, DRAM(860))와 인터페이스하기 위한 SDRAM 제어기(840), 비휘발성 메모리(예로서, 플래시(865))와 인터페이스하기 위한 플래시 제어기(845), 주변 장치들과 인터페이스하기 위한 주변 장치 제어(850)(예로서, 직렬 주변 장치 인터페이스), 입력(예로서, 터치 인에이블드 입력)을 표시 및 수신하기 위한 비디오 코덱(820) 및 비디오 인터페이스(825), 그래픽 관련 계산들을 수행하기 위한 GPU(815) 등과 같은 다른 컴포넌트들과의 통신 채널들을 제공한다. 이러한 인터페이스들 중 임의의 인터페이스는 본 명세서에서 설명되는 개시 내용의 양태들을 포함할 수 있다.

[0068] 일 실시예에서, 코어들(806, 807)은 본 발명의 실시예들에 따른, 가상 PCI 장치 및 가상 MMIO 장치의 형성을 구현하기 위한 논리도 포함한다. 예를 들어, 코어들(806, 807)은 도 1과 관련하여 설명된 가상 장치 모듈(125)과 같은 가상 장치 모듈을 실행하기 위한 논리를 포함할 수 있으며, 가상 장치 모듈은 DRAM, (860), 플래시(865) 등과 같은 주변 장치에 대한 요청에 기초하여 가상 장치를 형성할 수 있다. 가상 장치는 가상 PCI 장치 또는 가상 MMIO 장치일 수 있다.

[0069] 게다가, 시스템(800)은 블루투스 모듈(870), 3G 모뎀(875), GPS(880) 및 와이파이(885)와 같은 통신용 주변 장치들을 도시한다. 전술한 바와 같이, UE는 통신용 라디오를 포함한다는 점에 유의한다. 결과적으로, 이러한 주변장치 통신 모듈들은 모두가 필요한 것은 아니다. 그러나, UE에는, 소정 형태의 외부 통신용 라디오가 포함되어야 한다.

[0070] 도 9는 기계가 본 명세서에서 설명되는 방법들 중 어느 하나 이상을 수행하게 하기 위한 명령어들의 세트를 실행할 수 있는 컴퓨터 시스템(900)의 예시적인 형태의 기계의 개략도를 나타낸다. 대안 실시예들에서, 기계는 LAN, 인트라넷, 엑스트라넷 또는 인터넷 내에서 다른 기계들과 접속(예로서, 네트워크)될 수 있다. 기계는 클라이언트-서버 네트워크 환경에서 서버 또는 클라이언트 장치의 용량 내에서 또는 피어 대 피어 (또는 분산) 네트워크 환경에서 피어 기계로서 동작할 수 있다. 기계는 개인용 컴퓨터(PC), 태블릿 PC, 셋톱 박스(STB), 개인 휴대 단말기(PDA), 셀룰러 전화, 웹 설비, 서버, 네트워크 라우터, 스위치 또는 브리지, 또는 그 기계에 의해 취해질 액션들을 지정하는 (순차적인 또는 다른) 명령어들의 세트를 실행할 수 있는 임의의 기계일 수 있다. 또한, 단일 기계만이 도시되지만, 용어 "기계"는 본 명세서에서 설명되는 방법들 중 임의의 하나 이상을 수행하기 위해 명령어들의 세트(또는 다수의 세트)를 개별적으로 또는 공동으로 실행하는 기계들의 임의의 집합을 포

함하는 것으로도 간주되어야 한다.

- [0071] 컴퓨터 시스템(900)은 버스(930)를 통해 서로 통신하는 처리 장치(902), 메인 메모리(904)(예로서, 판독 전용 메모리(ROM), 플래시 메모리, 동적 랜덤 액세스 메모리(DRAM)(예로서, 동기 DRAM(SDRAM) 또는 DRAM(RDRAM)) 등), 정적 메모리(906)(예로서, 플래시 메모리, 정적 랜덤 액세스 메모리(SRAM) 등) 및 데이터 저장 장치(918)를 포함한다.
- [0072] 처리 장치(902)는 마이크로프로세서, 중앙 처리 유닛 등과 같은 하나 이상의 범용 처리 장치를 나타낸다. 더 구체적으로, 처리 장치는 복합 명령어 세트 컴퓨팅(CISC) 마이크로프로세서, 축소 명령어 세트 컴퓨터(RISC) 마이크로프로세서, 매우 긴 명령어 워드(VLIW) 마이크로프로세서, 또는 다른 명령어 세트들을 구현하는 프로세서, 또는 명령어 세트들의 조합을 구현하는 프로세서들일 수 있다. 처리 장치(902)는 주문형 집적 회로(ASIC), 필드 프로그래머블 게이트 어레이(FPGA), 디지털 신호 프로세서(DSP), 네트워크 프로세서 등과 같은 하나 이상의 특수 목적 처리 장치일 수도 있다. 일 실시예에서, 처리 장치(902)는 하나 이상의 처리 코어를 포함할 수 있다. 처리 장치(902)는 본 명세서에서 설명되는 동작들 및 단계들을 수행하기 위해 처리 논리(926)를 실행하도록 구성된다. 일 실시예에서, 처리 장치(902)는 가상 PCI 장치 및 가상 MMIO 장치를 형성하기 위한 일반적인 방법을 구현하는 도 1과 관련하여 설명된 처리 장치(100)와 동일하다. 예를 들어, 처리 장치(902)는 도 1의 가상 장치 모듈(125)과 같은 가상 장치 모듈을 포함할 수 있다.
- [0073] 컴퓨터 시스템(900)은 네트워크(920)에 통신 가능하게 결합되는 네트워크 인터페이스 장치(908)를 더 포함할 수 있다. 컴퓨터 시스템(900)은 비디오 디스플레이 유닛(910)(예로서, 액정 디스플레이(LCD) 또는 음극선관(CRT)), 영숫자 입력 장치(912)(예로서, 키보드), 커서 제어 장치(914)(예로서, 마우스) 및 신호 생성 장치(916)(예로서, 스피커)도 포함할 수 있다. 더구나, 컴퓨터 시스템(900)은 그래픽 처리 유닛(922), 비디오 처리 유닛(928) 및 오디오 처리 유닛(932)을 포함할 수 있다.
- [0074] 데이터 저장 장치(918)는 본 명세서에서 설명되는 기능들의 방법들 중 어느 하나 이상을 구현하는, 예를 들어 전술한 바와 같은 가상 PCI 장치 및 가상 MMIO 장치를 형성하기 위한 일반적인 방법을 구현하는 소프트웨어(926)를 저장하는 기계 판독 가능 저장 매체(924)를 포함할 수 있다. 소프트웨어(926)는 또한 완전히 또는 적어도 부분적으로 메인 메모리(904) 내에 명령어들(926)로서 그리고/또는 컴퓨터 시스템(900)에 의한 그의 실행 동안 처리 장치(902) 내에 처리 논리(926)로서 존재할 수 있으며, 메인 메모리(904) 및 처리 장치(902)는 또한 기계 액세스 가능 저장 매체를 구성할 수 있다.
- [0075] 기계 판독 가능 저장 매체(924)는 또한 도 1의 장치(100) 관련하여 설명된 바와 같은 가상 PCI 장치 및 가상 MMIO 장치를 형성하기 위한 일반적인 방법을 구현하는 명령어들(926) 및/또는 상기 애플리케이션들을 호출하는 방법들을 포함하는 소프트웨어 라이브러리를 저장하는 데 사용될 수 있다. 기계 판독 가능 저장 매체(924)는 일 실시예에서 단일 매체로서 도시되지만, 용어 "기계 액세스 가능 저장 매체"는 명령어들의 하나 이상의 세트를 저장하는 단일 매체 또는 다수의 매체(예로서, 중앙 또는 분산 데이터베이스 및/또는 관련 캐시들 및 서버들)를 포함하는 것으로 간주되어야 한다. 용어 "기계 판독 가능 저장 매체"는 또한, 기계에 의해 실행하기 위한 명령어 세트를 저장, 인코딩 또는 운반할 수 있고, 기계로 하여금 본 발명의 방법들 중 어느 하나 이상을 수행하게 하는 임의의 매체를 포함하는 것으로 간주되어야 한다. 따라서, 용어 "기계 판독 가능 저장 매체"는 반도체 메모리들 및 광학 및 자기 매체들을 포함하지만 이에 한정되지 않는 것으로 간주되어야 한다.
- [0076] 아래의 예들은 추가 실시예들과 관련된다.
- [0077] 예 1은 가상 장치를 형성하기 위한 기기로서, 1) 메모리; 및 2) 상기 메모리와 통신 가능하게 결합되는 처리 장치를 포함하고, 상기 처리 장치는 PCI(Peripheral Controller Interconnect) 호환 장치에 대한 PCI 요청을 수신하고, 상기 PCI 호환 장치에 기초하여 상기 가상 장치를 형성하며, 상기 가상 장치는 가상 PCI 장치 또는 가상 입출력(I/O) 장치 중 적어도 하나로서 형성되는 기기이다.
- [0078] 예 2에서, 예 1의 상기 PCI 호환 장치는 옵션으로서 소프트웨어 드라이버와 관련될 수 있고, 상기 소프트웨어 드라이버는 PCI 호환 액세스 요청을 상기 가상 장치로 전송하고, 상기 가상 장치는 상기 PCI 호환 액세스 요청을 기능 블록에 제공하고, 상기 기능 블록은 상기 PCI 호환 액세스 요청에 기초하여 상기 PCI 호환 장치와 통신한다.
- [0079] 예 3에서, 운영 체제가 옵션으로서 예 1의 상기 가상 장치를 이용하여 상기 PCI 호환 장치로부터 부팅된다.
- [0080] 예 4에서, 상기 PCI 호환 장치에 기초하여 상기 가상 장치를 형성하기 위해, 예 1의 상기 처리 장치는 옵션으로서 상기 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련된다는 결정 시에 상기 가상 장치를 상기 가상

PCI 장치로서 형성하고; 상기 PCI 호환 장치가 상기 PCI 호환 장치로부터 부팅될 운영 체제와 관련된다는 결정 시에 상기 가상 장치를 상기 가상 I/O 장치로서 형성할 수 있다.

- [0081] 예 5에서, 상기 가상 장치를 가상 PCI 장치로서 형성하기 위해, 예 4의 상기 처리 장치는 옵션으로서 상기 가상 장치에 대한 판매자 식별자를 결정하고; 상기 가상 장치에 대한 장치 정보를 결정하고; 상기 가상 장치에 대한 주소 정보를 결정할 수 있다.
- [0082] 예 6에서, 예 5의 상기 처리 장치는 옵션으로서 상기 PCI 호환 장치에 대한 상기 요청에 응답하여 상기 가상 장치에 대한 상기 판매자 식별자, 상기 가상 장치에 대한 상기 장치 정보 및 상기 장치에 대한 상기 주소 정보를 전송할 수 있다.
- [0083] 예 7에서, 상기 가상 장치를 가상 I/O 장치로서 형성하기 위해, 예 4의 상기 처리 장치는 옵션으로서 상기 PCI 호환 장치에 대한 PCI 열거 요청을 폐기하고, 상기 가상 장치에 대한 I/O 주소 범위를 결정할 수 있다.
- [0084] 예 8에서, 예 1의 상기 처리 장치는 옵션으로서 I/O 주소에 대한 메모리 액세스를 수신하고; 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련되는지를 결정하고; 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련된다는 결정 시에 상기 I/O 주소에 대응하는 상기 PCI 장치와 관련된 PCI 주소를 결정한다.
- [0085] 예 9에서, 예 1의 가상 I/O 장치는 옵션으로서 가상 메모리-맵핑 입출력(MMIO) 장치를 포함할 수 있다.
- [0086] 예 10에서, 예 1의 상기 요청은 옵션으로서 상기 PCI 호환 장치에 대한 PCI 열거 요청을 포함할 수 있다.
- [0087] 다양한 실시예들은 전술한 동작 특징들의 상이한 조합들을 가질 수 있다. 예를 들어, 전술한 기기의 모든 옵션 특징들은 본 명세서에서 설명되는 방법 또는 프로세스와 관련하여 구현될 수도 있으며, 예들 내에 상세들은 하나 이상의 실시예에서 어디서나 사용될 수 있다.
- [0088] 예 11은 가상 장치를 형성하기 위한 방법으로서, 1) PCI(Peripheral Controller Interconnect) 호환 장치에 대한 PCI 요청을 수신하는 단계; 및 2) 상기 PCI 호환 장치에 기초하여 상기 가상 장치를 형성하는 단계를 포함하며, 상기 가상 장치는 가상 PCI 장치 또는 가상 입출력(I/O) 장치 중 적어도 하나로서 형성되는 방법이다.
- [0089] 예 12에서, 예 11의 상기 PCI 호환 장치는 소프트웨어 드라이버와 관련되고, 상기 소프트웨어 드라이버는 PCI 호환 액세스 요청을 상기 가상 장치로 전송하고, 상기 가상 장치는 상기 PCI 호환 액세스 요청을 기능 블록에 제공하고, 상기 기능 블록은 상기 PCI 호환 액세스 요청에 기초하여 상기 PCI 호환 장치와 통신한다.
- [0090] 예 13에서, 운영 체제가 옵션으로서 예 11의 상기 가상 장치를 이용하여 상기 PCI 호환 장치로부터 부팅된다.
- [0091] 예 14에서, 예 11의 상기 PCI 호환 장치에 기초하여 상기 가상 장치를 형성하는 단계는 옵션으로서 상기 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련된다는 결정 시에 상기 가상 장치를 상기 가상 PCI 장치로서 형성하는 단계; 및 상기 PCI 호환 장치가 상기 PCI 호환 장치로부터 부팅될 운영 체제와 관련된다는 결정 시에 상기 가상 장치를 상기 가상 I/O 장치로서 형성하는 단계를 포함할 수 있다.
- [0092] 예 15에서, 예 14의 상기 가상 장치를 가상 PCI 장치로서 형성하는 단계는 옵션으로서 상기 가상 장치에 대한 판매자 식별자를 결정하는 단계; 상기 가상 장치에 대한 장치 정보를 결정하는 단계; 및 상기 가상 장치에 대한 주소 정보를 결정하는 단계를 포함할 수 있다.
- [0093] 예 16에서, 예 15의 내용은 옵션으로서 상기 PCI 호환 장치에 대한 상기 요청에 응답하여 상기 가상 장치에 대한 상기 판매자 식별자, 상기 가상 장치에 대한 상기 장치 정보 및 상기 장치에 대한 상기 주소 정보를 전송하는 단계를 포함할 수 있다.
- [0094] 예 17에서, 예 14의 상기 가상 장치를 가상 I/O 장치로서 형성하는 단계는 옵션으로서 상기 PCI 호환 장치에 대한 PCI 열거 요청을 폐기하는 단계; 및 상기 가상 장치에 대한 I/O 주소 범위를 결정하는 단계를 포함할 수 있다.
- [0095] 예 18에서, 예 11의 내용은 옵션으로서 I/O 주소에 대한 메모리 액세스를 수신하는 단계; 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련되는지를 결정하는 단계; 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련된다는 결정 시에 상기 I/O 주소에 대응하는 상기 PCI 장치와 관련된 PCI 주소를 결정하는 단계를 포함할 수 있다.
- [0096] 예 19에서, 예 11의 가상 I/O 장치는 옵션으로서 가상 메모리-맵핑 입출력(MMIO) 장치를 포함할 수 있다.

- [0097] 예 20에서, 예 11의 상기 요청은 옵션으로서 상기 PCI 호환 장치에 대한 PCI 열거 요청을 포함할 수 있다.
- [0098] 다양한 실시예들은 전술한 동작 특징들의 상이한 조합들을 가질 수 있다. 예를 들어, 전술한 방법의 모든 옵션 특징들은 비일시적 컴퓨터 판독 가능 저장 매체와 관련하여 구현될 수도 있다. 예들 내에 상세들은 하나 이상의 실시예에서 어디서나 사용될 수 있다.
- [0099] 예 21은 처리 장치에 의해 액세스될 때 상기 처리 장치로 하여금 예 11 내지 20의 방법을 수행하게 하는 데이터를 포함하는 비일시적 기계 판독 가능 저장 매체이다.
- [0100] 예 22는 가상 장치를 형성하기 위한 기기로서, 1) 메모리; 및 2) 상기 메모리에 결합되는 컴퓨팅 시스템을 포함하고, 상기 컴퓨팅 시스템은 제11항 내지 제20항 중 어느 한 항의 방법을 수행하도록 구성되는 기기이다.
- [0101] 예 23에서, 예 22의 컴퓨팅 시스템은 옵션으로서 PCI(Peripheral Controller Interconnect) 호환 장치에 대한 PCI 요청을 수신하기 위한 인터페이스; 및 상기 인터페이스에 결합되는 가상 장치 처리 블록을 포함할 수 있다.
- [0102] 예 24는 가상 장치를 형성하기 위한 컴퓨팅 시스템으로서, PCI(Peripheral Controller Interconnect) 호환 장치에 대한 PCI 요청을 수신하기 위한 인터페이스; 및 상기 인터페이스에 결합되는 가상 장치 처리 블록을 포함하고, 상기 가상 장치 처리 블록은 상기 PCI 호환 장치에 기초하여 상기 가상 장치를 형성하고, 상기 가상 장치는 가상 PCI 장치 또는 가상 입출력(I/O) 장치 중 적어도 하나로서 형성되는 컴퓨팅 시스템이다.
- [0103] 예 25에서, 예 24의 가상 장치 처리 블록은 옵션으로서 상기 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련되는지를 결정하고, 상기 PCI 호환 장치가 상기 PCI 호환 장치로부터 부팅될 운영 체제와 관련되는지를 결정하기 위한 가상 장치 결정 블록; 상기 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련될 때 상기 가상 장치를 상기 가상 PCI 장치로서 형성하기 위한 가상 PCI 장치 생성 블록; 및 상기 PCI 호환 장치가 상기 PCI 호환 장치로부터 부팅될 운영 체제와 관련될 때 상기 가상 장치를 상기 가상 I/O 장치로서 형성하기 위한 가상 I/O 장치 생성 블록을 포함할 수 있다.
- [0104] 예 26에서, 상기 가상 장치를 가상 PCI 장치로서 형성하기 위해, 예 25의 가상 PCI 장치 생성 블록은 옵션으로서 상기 가상 장치에 대한 판매자 식별자를 결정하고; 상기 가상 장치에 대한 장치 정보를 결정하고; 상기 가상 장치에 대한 주소 정보를 결정한다.
- [0105] 예 27에서, 상기 가상 장치를 가상 I/O 장치로서 형성하기 위해, 예 25의 가상 I/O 장치 생성 블록은 옵션으로서 상기 PCI 호환 장치에 대한 상기 PCI 열거 요청을 폐기하고; 상기 가상 장치에 대한 I/O 주소 범위를 결정한다.
- [0106] 예 28에서, 예 24의 가상 장치 처리 블록은 옵션으로서 상기 PCI 호환 장치에 대한 상기 요청에 응답하여 상기 가상 장치에 대한 상기 판매자 식별자, 상기 가상 장치에 대한 상기 장치 정보 및 상기 장치에 대한 상기 주소 정보를 전송할 수 있다.
- [0107] 예 29에서, 예 24의 가상 장치 처리 블록은 옵션으로서 I/O 주소에 대한 메모리 액세스를 수신하고, 상기 가상 장치가 가상 I/O 장치와 관련될 때 상기 I/O 주소가 상기 가상 장치와 관련되는지를 결정하고, 상기 가상 장치가 가상 I/O 장치와 관련될 때 상기 I/O 주소가 상기 가상 장치와 관련된다는 결정 시에, 상기 I/O 주소에 대응하는 상기 PCI 장치와 관련된 PCI 주소를 결정하기 위한 가상 I/O 장치 주소 변환 블록을 포함할 수 있다.
- [0108] 예 30은 컴퓨팅 시스템에 의해 실행될 때 상기 컴퓨팅 시스템으로 하여금 1) PCI(Peripheral Controller Interconnect) 호환 장치에 대한 PCI 요청을 수신하는 동작; 및 2) 상기 PCI 호환 장치에 기초하여 상기 가상 장치를 형성하는 동작을 포함하는 동작들을 수행하게 하는 명령어들을 포함하고, 상기 가상 장치는 가상 PCI 장치 또는 가상 입출력(I/O) 장치 중 적어도 하나로서 형성되는 비일시적 기계 판독 가능 저장 매체이다.
- [0109] 예 31에서, 예 30의 PCI 호환 장치는 옵션으로서 소프트웨어 드라이버와 관련될 수 있으며, 상기 소프트웨어 드라이버는 옵션으로서 PCI 호환 액세스 요청을 상기 가상 장치로 전송할 수 있고, 예 30의 상기 가상 장치는 옵션으로서 상기 PCI 호환 액세스 요청을 기능 블록에 제공할 수 있고, 상기 기능 블록은 옵션으로서 상기 PCI 호환 액세스 요청에 기초하여 상기 PCI 호환 장치와 통신할 수 있다.
- [0110] 예 32에서, 운영 체제가 예 30의 상기 가상 장치를 이용하여 상기 PCI 호환 장치로부터 부팅된다.
- [0111] 예 33에서, 예 30의 상기 PCI 호환 장치에 기초하여 상기 가상 장치를 형성하는 동작은 옵션으로서 상기 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련된다는 결정 시에, 상기 가상 장치를 상기 가상 PCI 장치로서 형성하는 동작; 및 상기 PCI 호환 장치가 상기 PCI 호환 장치로부터 부팅될 운영 체제와 관련된다는 결정 시에,

상기 가상 장치를 상기 가상 I/O 장치로서 형성하는 동작을 포함할 수 있다.

- [0112] 예 34에서, 예 33의 상기 가상 장치를 가상 PCI 장치로서 형성하는 동작은 옵션으로서 상기 가상 장치에 대한 판매자 식별자를 결정하는 동작; 상기 가상 장치에 대한 장치 정보를 결정하는 동작; 및 상기 가상 장치에 대한 주소 정보를 결정하는 동작을 포함할 수 있다.
- [0113] 예 35에서, 예 34의 상기 가상 장치를 가상 PCI 장치로서 형성하는 동작은 옵션으로서 상기 PCI 호환 장치에 대한 상기 요청에 응답하여 상기 가상 장치에 대한 상기 판매자 식별자, 상기 가상 장치에 대한 상기 장치 정보 및 상기 장치에 대한 상기 주소 정보를 전송하는 동작을 포함할 수 있다.
- [0114] 예 36에서, 예 33의 상기 가상 장치를 가상 I/O 장치로서 형성하는 동작은 옵션으로서 상기 PCI 호환 장치에 대한 상기 PCI 열거 요청을 폐기하는 동작; 및 상기 가상 장치에 대한 I/O 주소 범위를 결정하는 동작을 포함할 수 있다.
- [0115] 예 37에서, 예 30의 내용은 옵션으로서 I/O 주소에 대한 메모리 액세스를 수신하는 동작; 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련되는지를 결정하는 동작; 및 상기 가상 장치가 가상 I/O 장치로서 형성될 때 상기 I/O 주소가 상기 가상 장치와 관련된다는 결정 시에, 상기 I/O 주소에 대응하는 상기 PCI 장치와 관련된 PCI 주소를 결정하는 동작을 포함할 수 있다.
- [0116] 예 38은 가상 장치를 형성하기 위한 기기로서, 1) PCI(Peripheral Controller Interconnect) 호환 장치에 대한 PCI 요청을 수신하기 위한 인터페이스; 및 2) 상기 PCI 호환 장치에 기초하여 상기 가상 장치를 형성하기 위한 수단을 포함하고, 상기 가상 장치는 가상 PCI 장치 또는 가상 입출력(I/O) 장치 중 적어도 하나로서 형성되는 기기이다.
- [0117] 예 39에서, 예 38의 상기 PCI 호환 장치에 기초하여 상기 가상 장치를 형성하기 위한 상기 수단은 옵션으로서 상기 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련되는지를 결정하기 위한 수단; 상기 PCI 호환 장치가 PCI 호환형이 아닌 기능 블록과 관련된다는 결정 시에, 상기 가상 장치를 상기 가상 PCI 장치로서 형성하기 위한 수단; 상기 PCI 호환 장치가 상기 PCI 호환 장치로부터 부팅될 운영 체제와 관련되는지를 결정하기 위한 수단; 및 상기 PCI 호환 장치가 상기 PCI 호환 장치로부터 부팅될 운영 체제와 관련된다는 결정 시에, 상기 가상 장치를 상기 가상 I/O 장치로서 형성하기 위한 수단을 포함할 수 있다.
- [0118] 위의 설명에서 다수의 상세가 설명되었다. 그러나, 본 개시 내용은 이러한 특정 상세 없이도 실시될 수 있다는 것이 이 분야의 기술자에게 명백할 것이다. 일부 예들에서는, 본 개시 내용을 불명확하게 하지 않기 위해, 공지 구조들 및 장치들은 상세하게 도시되지 않고 블록도 형태로 도시된다.
- [0119] 상세한 설명의 일부 부분들은 컴퓨터 메모리 내의 데이터 비트들에 대한 동작들의 알고리즘들 및 심벌 표현들과 관련하여 제공된다. 이러한 알고리즘 설명들 및 표현들은 데이터 처리 분야의 기술자들이 그들의 연구 내용을 그 분야의 다른 기술자들에게 가장 효과적으로 전달하는 데 사용하는 수단이다. 본 명세서에서 그리고 일반적으로 알고리즘은 원하는 결과를 유도하는 동작들의 일관된 시퀀스인 것으로 간주된다. 동작들은 물리량들의 물리적 조작들을 필요로 하는 것들이다. 통상적으로, 반드시 필요한 것은 아니지만, 이러한 양들은 저장, 전송, 결합, 비교, 아니면 다른 방식으로 조작될 수 있는 전기 또는 자기 신호들의 형태를 취한다. 때때로, 주로 통상적인 사용의 이유로, 이러한 신호들을 비트, 값, 요소, 심벌, 문자, 용어, 숫자 등으로 지칭하는 것이 편리한 것으로 입증되었다. 본 명세서에서 설명되는 블록들은 하드웨어, 소프트웨어, 펌웨어 또는 이들의 조합일 수 있다.
- [0120] 그러나, 이들 및 유사한 용어들 모두는 적절한 물리량들과 관련되며, 이러한 양들에 적용되는 편리한 라벨들일 뿐이라는 것을 기억해야 한다. 구체적으로 달리 언급되지 않는 한은, 아래의 설명으로부터 명백하듯이, 본 설명 전반에서 "전송", "수신", "생성(generating)", "결정", "생성(creating)", "변환", "폐기", "비교" 등과 같은 용어들을 이용한 설명들은 컴퓨터 시스템의 레지스터들 및 메모리들 내에 물리(전자) 양들로서 표현되는 데이터를 조작하고, 컴퓨터 시스템 메모리들 또는 레지스터들 또는 다른 그러한 정보 저장, 전송 또는 디스플레이 장치들 내에 물리량들로서 유사하게 표현되는 다른 데이터로 변환하는 컴퓨터 시스템 또는 유사한 전자 컴퓨팅 장치의 액션 및 프로세스들을 지칭한다는 것을 안다.
- [0121] 본 개시 내용은 본 명세서에서의 동작들을 수행하기 위한 기기와도 관련된다. 이러한 기기는 필요한 목적들을 위해 특별히 구성될 수 있거나, 컴퓨터에 저장된 컴퓨터 프로그램에 의해 선택적으로 작동되거나 재구성되는 범용 컴퓨터를 포함할 수 있다. 그러한 컴퓨터 프로그램은 컴퓨터 시스템 버스에 각자 결합되는 플로피 디스크, 광 디스크, CD-ROM 및 광자기 디스크를 포함하는 임의 타입의 디스크, 판독 전용 메모리(ROM), 랜덤 액세스 메

모리(RAM), EPROM, EEPROM, 자기 또는 광학 카드, 또는 전자 명령어들을 저장하는 데 적합한 임의 타입의 매체와 같은, 그러나 이에 한정되지 않는 기계 판독 가능 저장 매체 내에 저장될 수 있다.

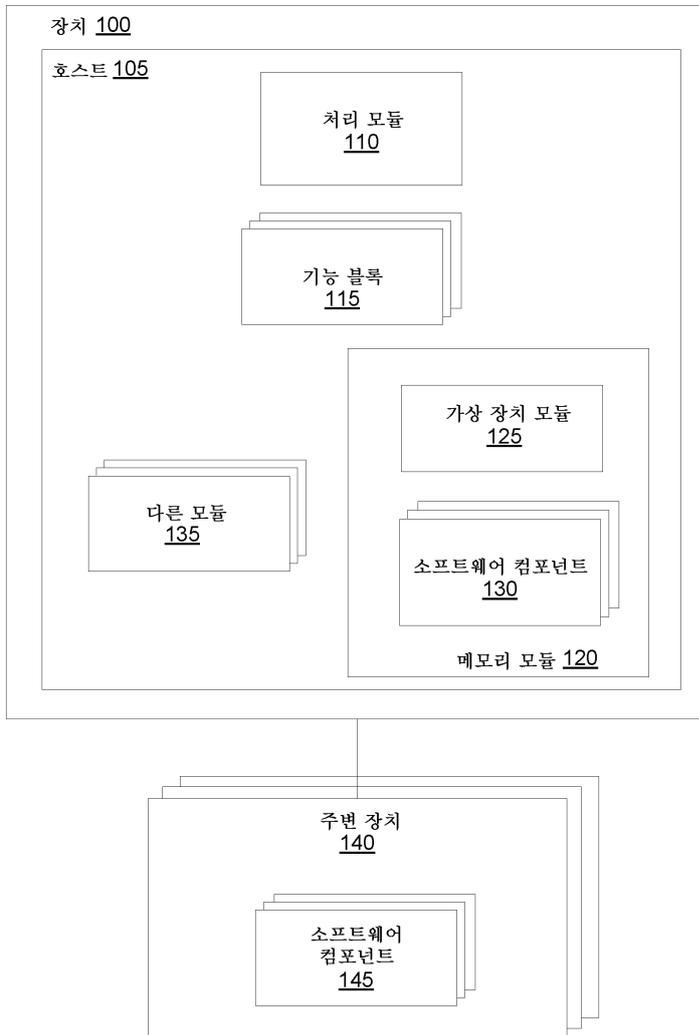
[0122] 본 명세서에서 제공되는 알고리즘들 및 디스플레이들은 임의의 특정 컴퓨터 또는 다른 기기와 본질적으로 관련되지는 않는다. 다양한 범용 시스템들이 본 명세서에서의 가르침에 따른 프로그램들과 함께 사용될 수 있거나, 동작들을 수행하기 위해 더 특수화된 기기를 구성하는 것이 편리한 것으로 입증될 수 있다. 다양한 이러한 시스템들의 필요한 구조는 아래의 설명으로부터 나타날 것이다. 게다가, 본 실시예들은 임의의 특정 프로그래밍 언어와 관련하여 설명되지 않는다. 본 명세서에서 설명되는 바와 같은 실시예들의 가르침들을 구현하기 위해 다양한 프로그래밍 언어들이 사용될 수 있다는 것을 알 것이다.

[0123] 본 개시 내용은 본 개시 내용에 따른 프로세스를 수행하도록 컴퓨터 시스템(또는 다른 전자 장치들)을 프로그래밍하는 데 사용될 수 있는 명령어들을 저장한 기계 판독 가능 매체를 포함할 수 있는 컴퓨터 프로그램 제품 또는 소프트웨어로서 제공될 수 있다. 기계 판독 가능 매체는 정보를 기계(예로서, 컴퓨터)에 의해 판독될 수 있는 형태로 저장하거나 전송하기 위한 임의의 기술을 포함한다. 예를 들어, 기계 판독 가능(예로서, 컴퓨터 판독 가능) 매체는 기계(예로서, 컴퓨터) 판독 가능 저장 매체(예로서, 판독 전용 메모리("ROM"), 랜덤 액세스 메모리("RAM"), 자기 디스크 저장 매체, 광학 저장 매체, 플래시 메모리 장치 등) 등을 포함한다.

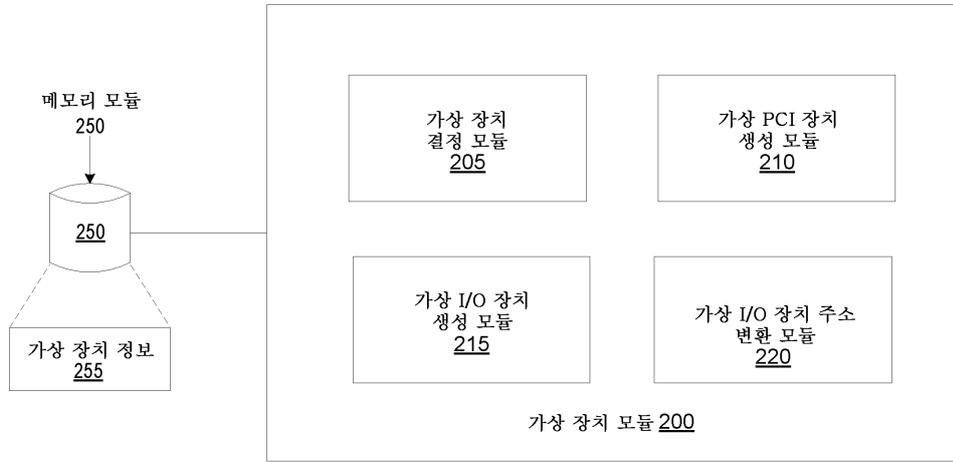
[0124] 본 개시 내용의 많은 변경 및 수정이 위의 설명을 읽은 후의 통상의 기술자에게 아마도 명백하겠지만, 예시를 통해 도시되고 설명된 임의의 특정 실시예는 결코 한정으로 간주되는 것을 의도하지 않는다는 것을 이해해야 한다. 따라서, 다양한 실시예들의 상세들에 대한 참조들은 본 발명으로서 간주되는 특징들만을 자체적으로 상술하는 청구항들의 범위를 한정하는 것을 의도하지 않는다.

도면

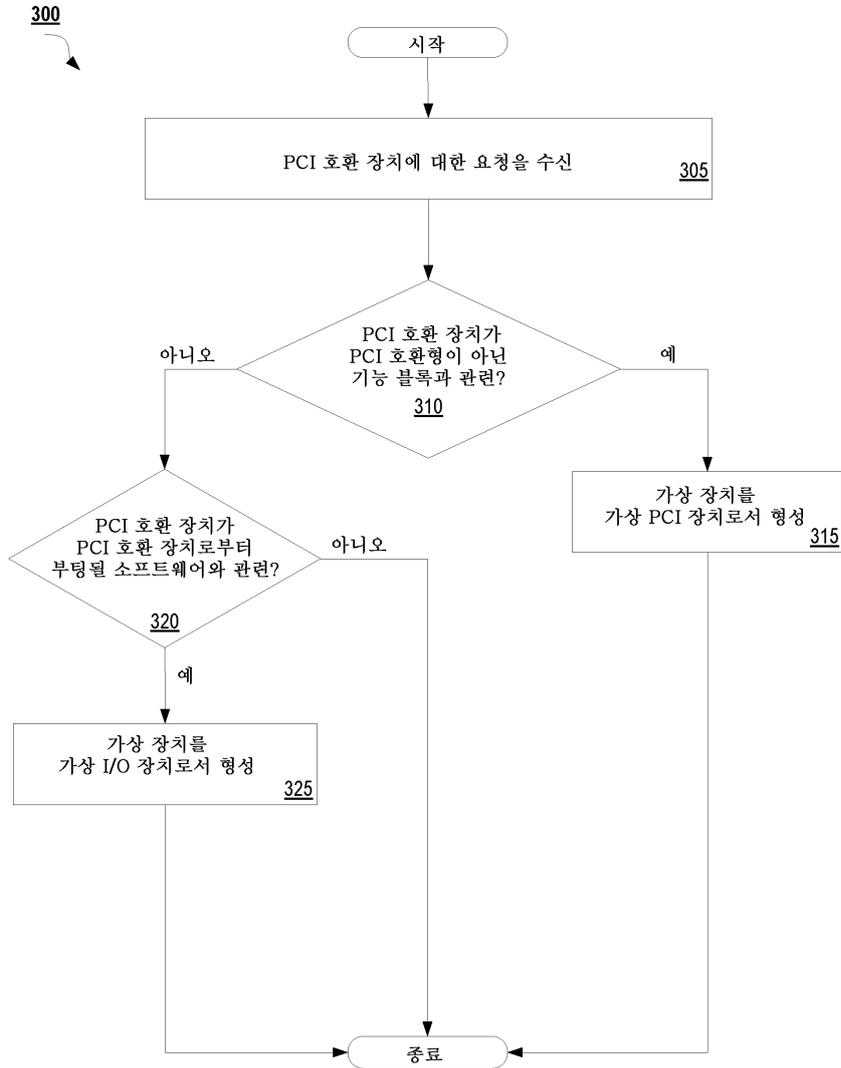
도면1



도면2

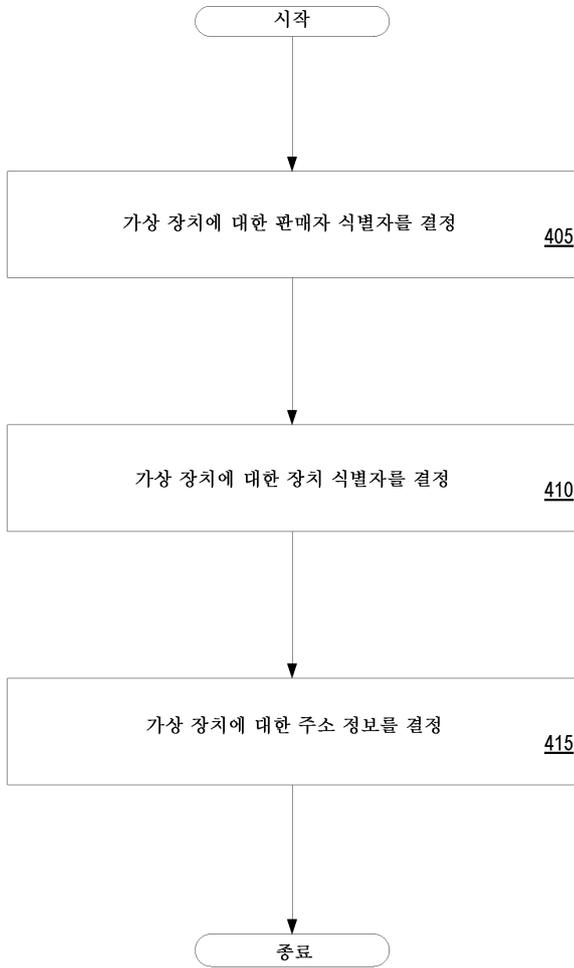


도면3



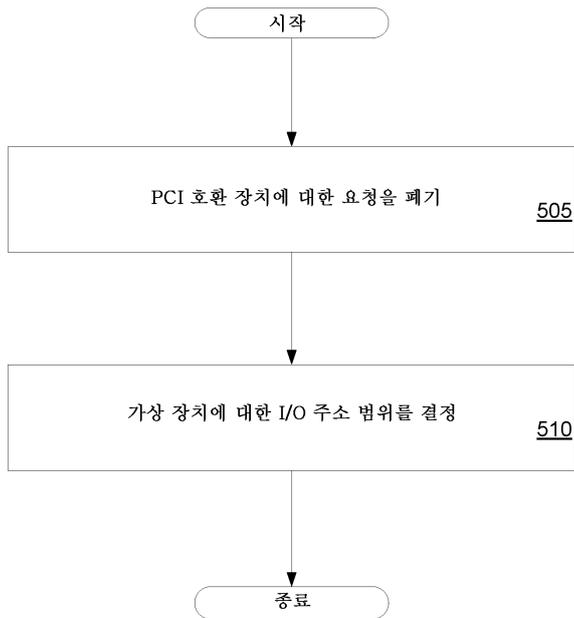
도면4

400



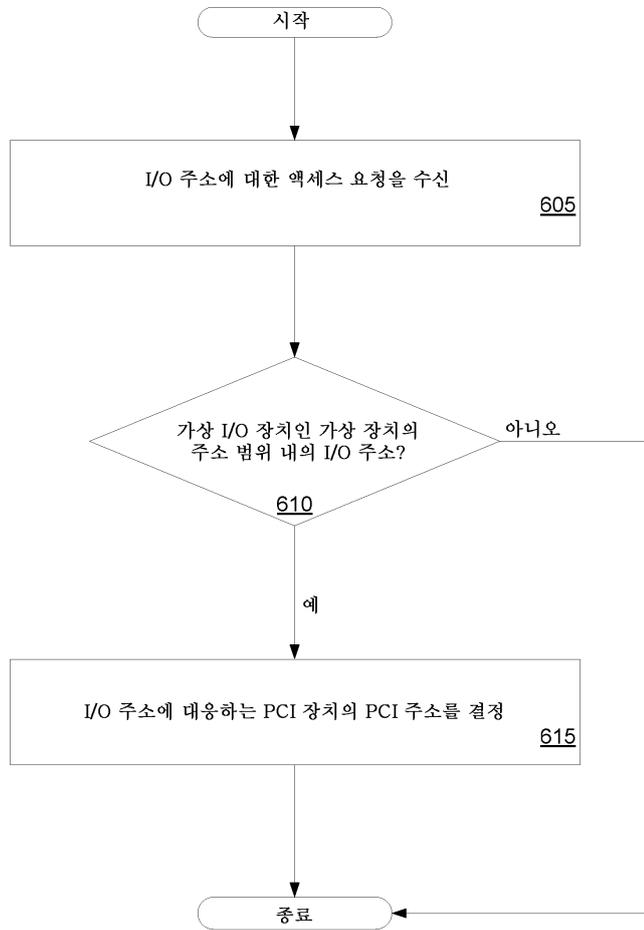
도면5

500

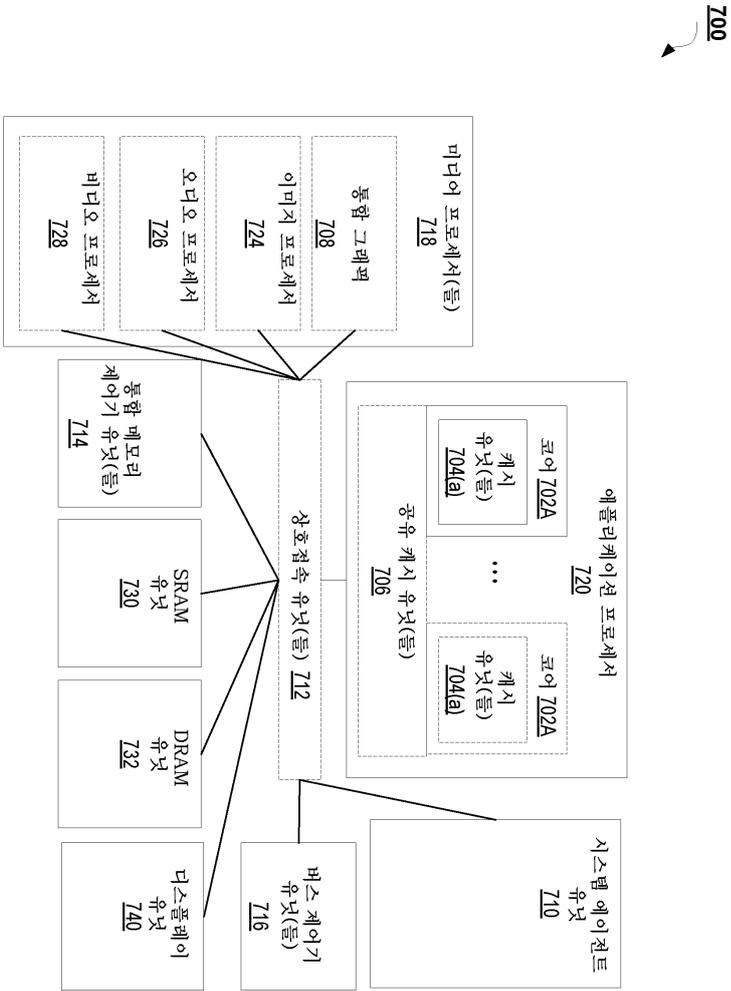


도면6

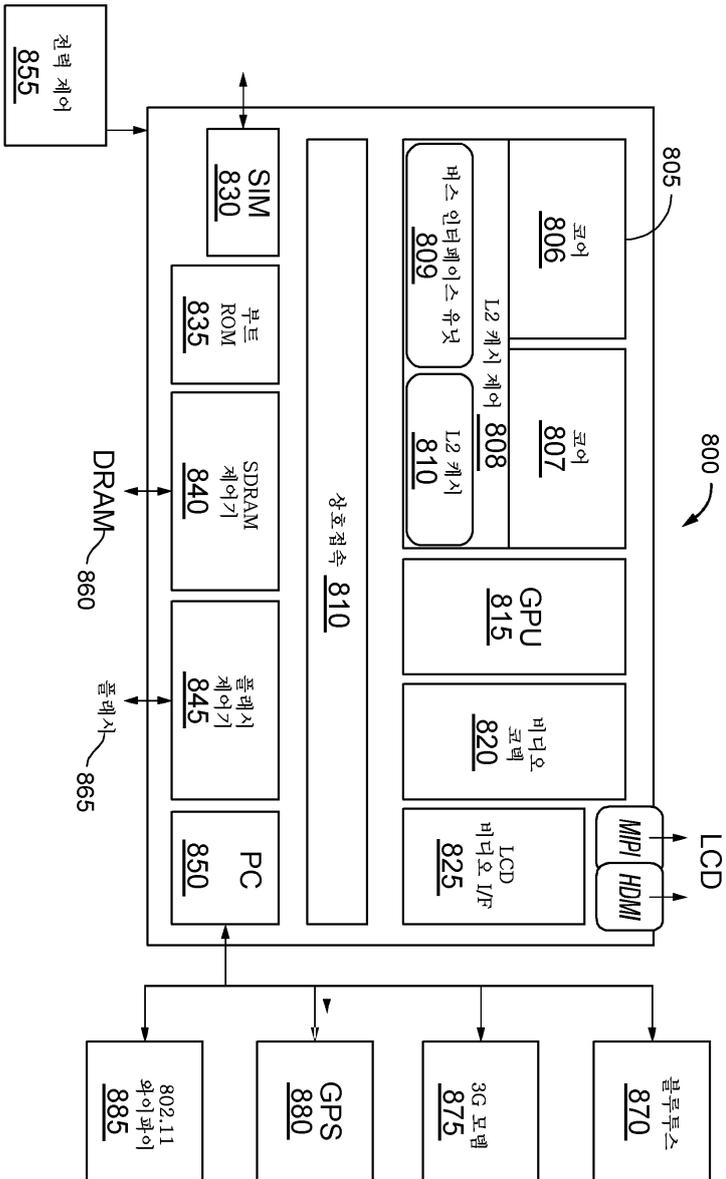
600



도면7



도면8



도면9

