



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0131511
(43) 공개일자 2017년11월29일

- (51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) **H04L 25/03** (2006.01)
H04L 29/10 (2006.01) **H04W 76/02** (2009.01)
- (52) CPC특허분류
H04L 69/24 (2013.01)
H04L 25/03 (2013.01)
- (21) 출원번호 10-2017-7029989
- (22) 출원일자(국제) 2016년03월18일
심사청구일자 없음
- (85) 번역문제출일자 2017년10월18일
- (86) 국제출원번호 PCT/US2016/023199
- (87) 국제공개번호 WO 2016/149641
국제공개일자 2016년09월22일
- (30) 우선권주장
62/135,007 2015년03월18일 미국(US)
15/073,398 2016년03월17일 미국(US)

- (71) 출원인
래티스세미컨덕터코퍼레이션
미국 오리건 (우편번호 97204) 포틀랜드 에스더블유 피프쓰 애비뉴 111 스위트 700
- (72) 발명자
바바지 투, 데이비드, 노엘
미국, 97204 오리건, 포틀랜드, 스위트 700, 에스더블유 피프쓰 애비뉴 111, 래티스 세미컨덕터 코퍼레이션 내
도안, 친, 후이
(뒷면에 계속)
- (74) 대리인
특허법인에이아이피

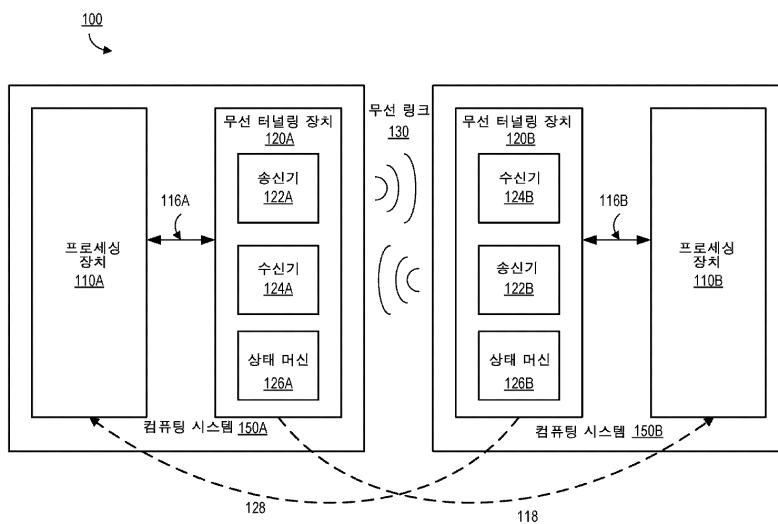
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 멀티-기가비트 무선 터널링 시스템

(57) 요약

개시된 무선 터널링 시스템은, 2개의 프로세싱 장치들 사이의 통신들의 유선 통신 프로토콜과의 컴플라이언스를 유지하면서, 유선 링크를 통한 2개의 장치들 사이의 통신들을 터널링한다. 일 실시예에 있어서, 무선 터널링 시스템은, 서로 무선 링크를 통해 통신하는 2개의 무선 터널링 장치들을 포함한다. 로컬 무선 터널링 장치는 유선 연결을 통해 로컬 프로세싱 장치에 결합되며, 원격 무선 터널링 장치는 다른 유선 연결을 통해 원격 프로세싱 장치에 결합된다. 일 측면에 있어서, 로컬 무선 터널링 장치는 원격 프로세싱 장치의 상태를 예측하고, 원격 프로세싱 장치의 예측된 상태를 미러링한다. 예측에 기초하여 미러링하는 것은, 고속 데이터 레이트 터널링에 영향을 주는 무선 터널링 장치들과 연관된 자연 없이 무선 링크를 통한 2개의 프로세싱 장치들 사이의 고속 데이터 레이트 터널링을 가능하게 한다.

대 표 도



(52) CPC특허분류

H04L 29/10 (2013.01)

H04W 76/022 (2013.01)

(72) 발명자

포브스, 마크, 그라함

미국, 97204 오리건, 포틀랜드, 스위트 700, 에스 더블유 피프쓰 애비뉴 111, 래티스 세미컨덕터 코퍼레이션 내

존, 브라이언, 헨리

미국, 97204 오리건, 포틀랜드, 스위트 700, 에스 더블유 피프쓰 애비뉴 111, 래티스 세미컨덕터 코퍼레이션 내

쿠마르, 니시트

미국, 97204 오리건, 포틀랜드, 스위트 700, 에스 더블유 피프쓰 애비뉴 111, 래티스 세미컨덕터 코퍼레이션 내

명세서

청구범위

청구항 1

원격 프로세싱 장치와 로컬 프로세싱 장치 사이의 통신들의 유선 통신 프로토콜과의 컴플라이언스(compliance)를 유지하면서, 상기 원격 프로세싱 장치와 상기 로컬 프로세싱 장치 사이의 통신들을 무선으로 터널링(tunneling)하기 위하여 원격 무선 터널링 장치와 함께 동작하는 로컬 무선 터널링 장치로서,

상기 원격 무선 터널링 장치로부터 무선 수신 신호를 수신하도록 구성되며, 상기 무선 수신 신호로부터 기저대역 신호를 생성하기 위하여 상기 무선 수신 신호를 다운컨버팅(downconvert)하는 무선 수신기;

상기 기저대역 신호에 기초하여 상기 원격 프로세싱 장치의 원격 프로세싱 상태를 예측하도록 구성된 프로세싱 컴포넌트 상태 머신; 및

상기 로컬 프로세싱 장치에 결합되며, (i) 상기 예측된 원격 프로세싱 상태 및 상기 기저대역 신호에 기초하여 상기 유선 통신 프로토콜에 따르는 출력 신호를 생성하고 및 (ii) 상기 유선 프로토콜을 통해 상기 로컬 프로세싱 장치로 상기 출력 신호를 제공하도록 구성된 인터페이스 회로를 포함하는, 로컬 무선 터널링 장치.

청구항 2

청구항 1에 있어서,

상기 로컬 무선 터널링 장치는,

다른 기저대역 신호를 생성하기 위하여 상기 로컬 프로세싱 장치로부터 입력 신호를 수신하고, (ii) 무선 송신 신호를 생성하기 위하여 상기 다른 기저대역 신호를 업컨버팅(upconvert)하며 및 (iii) 상기 무선 송신 신호를 송신하도록 구성된 무선 송신기를 더 포함하는, 로컬 무선 터널링 장치.

청구항 3

청구항 2에 있어서,

상기 프로세싱 컴포넌트 상태 머신은:

상기 프로세싱 컴포넌트 상태 머신의 현재 상태,

상기 로컬 프로세싱 장치로부터 수신된 입력들, 및

상기 예측된 원격 프로세싱 상태 중 하나 이상에 기초하여 상기 수신기 또는 상기 송신기의 전력 상태를 제어하는, 로컬 무선 터널링 장치.

청구항 4

청구항 2에 있어서,

상기 프로세싱 컴포넌트 상태 머신은, 상기 로컬 프로세싱 장치의 하나 이상의 로컬 프로세싱 상태들을 상기 프로세싱 컴포넌트 상태 머신의 단일 상태로 매핑(map)하고, 상기 단일 상태에 기초하여 상기 로컬 프로세싱 장치의 로컬 프로세싱 상태를 나타내는 상태 신호를 생성하도록 더 구성되며, 상기 송신기는 상기 로컬 프로세싱 장치의 상기 로컬 프로세싱 상태를 나타내는 상기 상태 신호를 가지고 상기 기저대역 신호를 인코딩하도록 구성되는, 로컬 무선 터널링 장치.

청구항 5

청구항 1에 있어서,

상기 로컬 무선 터널링 장치는,

상기 원격 프로세싱 장치의 상기 예측된 원격 프로세싱 상태 및 그 자체의 현재 상태에 기초하여 상기 무선 수신기의 무선 컴포넌트 상태를 결정하도록 구성된 무선 컴포넌트 상태 머신을 더 포함하는, 로컬 무선 터널링 장치.

청구항 6

청구항 5에 있어서,

상기 무선 컴포넌트 상태 머신은 상기 원격 프로세싱 장치의 상기 예측된 원격 프로세싱 상태 및 그 자체의 현재 상태에 기초하여 결정된 상기 무선 컴포넌트 상태에 따라 상기 무선 수신기의 동작 모드를 제어하는, 로컬 무선 터널링 장치.

청구항 7

청구항 6에 있어서,

상기 무선 컴포넌트 상태는 상기 예측된 원격 프로세싱 상태 및 그 자체의 현재 상태에 기초하여 결정되는, 로컬 무선 터널링 장치.

청구항 8

청구항 1에 있어서,

상기 기저대역 신호는 상기 원격 프로세싱 장치의 이전 상태를 나타내는 상태 신호를 가지고 인코딩되며, 상기 프로세싱 컴포넌트 상태 머신은 상기 기저대역 신호의 상기 상태 신호에 기초하여 상기 원격 프로세싱 장치의 상기 원격 프로세싱 상태를 예측하도록 구성되는, 로컬 무선 터널링 장치.

청구항 9

청구항 1에 있어서,

상기 프로세싱 컴포넌트 상태 머신은, 상기 무선 수신 신호에 대응하는 상기 기저대역 신호의 일 부분에 기초하여 상기 원격 프로세싱 장치의 상기 원격 프로세싱 상태를 예측하도록 구성되는, 로컬 무선 터널링 장치.

청구항 10

청구항 1에 있어서,

상기 프로세싱 컴포넌트 상태 머신은, 상기 로컬 프로세싱 장치의 하나 이상의 이전 로컬 프로세싱 상태들에 기초하여 상기 원격 프로세싱 장치의 상기 원격 프로세싱 상태를 예측하도록 구성되는, 로컬 무선 터널링 장치.

청구항 11

로컬 프로세싱 장치와 원격 프로세싱 장치 사이의 통신들의 유선 통신 프로토콜과의 캠플라이언스를 유지하면서 상기 로컬 프로세싱 장치와 상기 원격 프로세싱 장치 사이의 통신들을 무선으로 터널링하기 위한 방법으로서,

무선 수신기에 의해, 원격 무선 터널링 장치로부터 무선 수신 신호를 수신하는 단계;

상기 무선 수신기에 의해, 상기 무선 수신 신호로부터 기저대역 신호를 생성하기 위하여 상기 무선 수신 신호를 다운컨버팅하는 단계;

프로세싱 컴포넌트 상태 머신에 의해, 상기 기저대역 신호에 기초하여 상기 원격 프로세싱 장치의 원격 프로세싱 상태를 예측하는 단계;

상기 로컬 프로세싱 장치에 결합된 인터페이스 회로에 의해, 상기 예측된 원격 프로세싱 상태 및 상기 기저대역 신호에 기초하여 상기 유선 통신 프로토콜에 따르는 출력 신호를 생성하는 단계; 및

상기 인터페이스 회로에 의해, 상기 출력 신호를 상기 유선 통신 프로토콜을 통해 상기 로컬 프로세싱 장치에 제공하는 단계를 포함하는, 방법.

청구항 12

청구항 11에 있어서,

상기 방법은,

무선 송신기에 의해, 다른 기저대역 신호를 생성하기 위하여 상기 로컬 프로세싱 장치로부터 입력 신호를 수신하는 단계;

상기 무선 송신기에 의해, 무선 송신 신호를 생성하기 위하여 상기 다른 기저대역 신호를 업컨버팅하는 단계; 및

상기 무선 송신기에 의해, 상기 무선 송신 신호를 송신하는 단계를 더 포함하는, 방법.

청구항 13

청구항 12에 있어서,

상기 방법은,

상기 프로세싱 컴포넌트 상태 머신에 의해,

상기 프로세싱 컴포넌트 상태 머신의 현재 상태,

상기 로컬 프로세싱 장치로부터 수신된 입력들, 및

상기 예측된 원격 프로세싱 상태 중 하나 이상에 기초하여 상기 수신기 또는 상기 송신기의 전력 상태를 제어하는 단계를 더 포함하는, 방법.

청구항 14

청구항 12에 있어서,

상기 방법은,

프로세싱 컴포넌트 상태 머신에 의해, 상기 로컬 프로세싱 장치의 하나 이상의 로컬 프로세싱 상태들을 상기 프로세싱 컴포넌트 상태 머신의 단일 상태로 매핑하는 단계;

상기 프로세싱 컴포넌트 상태 머신에 의해, 상기 단일 상태에 기초하여 상기 로컬 프로세싱 장치의 로컬 프로세싱 상태를 나타내는 상태 신호를 생성하는 단계; 및

상기 송신기에 의해, 상기 로컬 프로세싱 장치의 상기 로컬 프로세싱 상태를 나타내는 상기 상태 신호를 가지고 상기 기저대역 신호를 인코딩하는 단계를 더 포함하는, 방법.

청구항 15

청구항 11에 있어서,

상기 방법은,

무선 컴포넌트 상태 머신에 의해, 상기 원격 프로세싱 장치의 상기 예측된 원격 프로세싱 상태에 기초하여 상기 무선 수신기의 무선 컴포넌트 상태를 결정하는 단계를 더 포함하는, 방법.

청구항 16

청구항 15에 있어서,

상기 방법은,

상기 무선 컴포넌트 상태 머신에 의해, 상기 원격 프로세싱 장치의 상기 예측된 원격 프로세싱 상태에 기초하여 결정된 상기 무선 컴포넌트 상태에 따라 상기 무선 수신기의 동작 모드를 제어하는 단계를 더 포함하는, 방법.

청구항 17

청구항 16에 있어서,

상기 무선 컴포넌트 상태는 상기 예측된 원격 프로세싱 상태에 기초하여 결정되는, 방법.

청구항 18

청구항 11에 있어서,

상기 기저대역 신호는 상기 원격 프로세싱 장치의 이전 상태를 나타내는 상태 신호를 가지고 인코딩되며, 상기 프로세싱 컴포넌트 상태 머신은 상기 기저대역 신호의 상기 상태 신호에 기초하여 상기 원격 프로세싱 장치의 상기 원격 프로세싱 상태를 예측하도록 구성되는, 방법.

청구항 19

청구항 11에 있어서,

상기 프로세싱 컴포넌트 상태 머신은, 상기 무선 수신 신호에 대응하는 상기 기저대역 신호의 일 부분에 기초하여 상기 원격 프로세싱 장치의 상기 원격 프로세싱 상태를 예측하도록 구성되는, 방법.

청구항 20

청구항 11에 있어서,

상기 프로세싱 컴포넌트 상태 머신은, 상기 로컬 프로세싱 장치의 하나 이상의 이전 로컬 프로세싱 상태들에 기초하여 상기 원격 프로세싱 장치의 상기 원격 프로세싱 상태를 예측하도록 구성되는, 방법.

발명의 설명

기술 분야

[0001] 본 개시의 실시예들은 전반적으로 무선 통신 분야에 관한 것으로서, 더 구체적으로는, 유선 통신 프로토콜의 무선 터널링에 관한 것이다.

배경 기술

[0002] 무선 터널링 시스템에 있어서, 전통적으로 유선 링크를 통해 통신되는 데이터가 그 대신에 무선 채널을 통해 터널링(tunnel)된다. 통상적으로, 무선 통신들은 실질적으로 유선 링크들을 통한 통신들보다 더 느리다. 따라서, 통상적인 무선 시스템들은 고속 프로토콜, 예를 들어, 범용 직렬 버스(Universal Serial Bus; USB), 고화질 매체 인터페이스(High-Definition Media Interface; HDMI), 및 멀티-기기비트 데이터 레이트들을 갖는 디스플레이포트(DisplayPort; DP)를 터널링하는 것이 불가능하다.

발명의 내용

과제의 해결 수단

[0003] 개시된 무선 터널링 시스템은, 2개의 프로세싱 장치들 사이의 통신들의 컴플라이언스(compliance)를 유지하면서, 유선 링크를 통한 2개의 장치들 사이의 통신들을 터널링한다.

[0004] 일 실시예에 있어서, 무선 터널링 시스템은, 서로 무선 링크를 통해 통신하는 2개의 무선 터널링 장치들을 포함한다. 로컬 무선 터널링 장치는 유선 연결을 통해 로컬 프로세싱 장치에 결합되며, 원격 무선 터널링 장치는 다른 유선 연결을 통해 원격 프로세싱 장치에 결합된다. 2개의 프로세싱 장치들은, 마치 2개의 프로세싱 장치들이 유선 연결들을 통해 연결된 것처럼 2개의 무선 터널링 장치들을 사용하여 저-레이턴시(low-latency) 무선 링크를 통해 서로 통신할 수 있다.

[0005] 일 실시예에 있어서, 로컬 무선 터널링 장치는 무선 수신기, 프로세싱 컴포넌트 상태 머신, 및 인터페이스 회로를 포함한다. 무선 수신기는 원격 무선 터널링 장치로부터 무선 수신 신호를 수신하고, 무선 수신 신호로부터 기저대역 신호를 생성하기 위하여 무선 수신 신호를 다운컨버팅(downconvert)하도록 구성된다. 프로세싱 컴포넌트 상태 머신은 기저대역 신호에 기초하여 원격 프로세싱 장치의 원격 프로세싱 상태를 예측하도록 구성된다. 인터페이스 회로는 로컬 프로세싱 장치에 결합되며, (i) 예측된 원격 프로세싱 상태 및 기저대역 신호에 기초하여 유선 통신 프로토콜에 따르는 출력 신호를 생성하고 및 (ii) 유선 통신 프로토콜을 통해 로컬 프로세싱 장치로 출력 신호를 제공하도록 구성된다.

[0006] 하나 이상의 실시예들에 있어서, 로컬 무선 터널링 장치는, (i) 다른 기저대역 신호를 생성하기 위하여 로컬 프로세싱 장치로부터 입력 신호를 수신하고, (ii) 무선 송신 신호를 생성하기 위하여 상기 다른 기저대역 신호를 업컨버팅(upconvert)하며 및 (iii) 무선 송신 신호를 송신하도록 구성된 무선 송신기를 더 포함한다. 프로세싱 컴포넌트 상태 머신은: (a) 프로세싱 컴포넌트 상태 머신의 현재 상태, (b) 로컬 프로세싱 장치로부터 수신된 입력들, 및 (c) 예측된 원격 프로세싱 상태 중 하나 이상에 기초하여 수신기 또는 송신기의 전력 상태를 제어할 수 있다. 프로세싱 컴포넌트 상태 머신은, 로컬 프로세싱 장치의 하나 이상의 로컬 프로세싱 상태들을 프로세싱 컴포넌트 상태 머신의 단일 상태로 매핑(map)하고, 단일 상태에 기초하여 로컬 프로세싱 장치의 로컬 프로세싱 상태를 나타내는 상태 신호를 생성하도록 더 구성될 수 있다. 송신기는, 로컬 프로세싱 장치의 로컬 프로세싱 상태를 나타내는 상태 신호를 가지고 기저대역 신호를 인코딩하도록 더 구성될 수 있다.

[0007] 하나 이상의 실시예들에 있어서, 로컬 무선 터널링 장치는, 원격 프로세싱 장치의 예측된 원격 프로세싱 상태 및 그 자체의 현재 상태에 기초하여 무선 수신기의 무선 컴포넌트 상태를 결정하도록 구성된 무선 컴포넌트 상태 머신을 더 포함한다. 무선 컴포넌트 상태 머신은 원격 프로세싱 장치의 예측된 원격 프로세싱 상태 및 그 자체의 현재 상태에 기초하여 결정된 무선 컴포넌트 상태에 따라 무선 수신기의 동작 모드를 제어할 수 있다.

[0008] 하나 이상의 실시예들에 있어서, 기저대역 신호는 원격 프로세싱 장치의 이전 상태를 나타내는 상태 신호를 가지고 인코딩된다. 프로세싱 컴포넌트 상태 머신은 기저대역 신호의 상태 신호에 기초하여 원격 프로세싱 장치의 원격 프로세싱 상태를 예측하도록 구성될 수 있다.

[0009] 하나 이상의 실시예들에 있어서, 프로세싱 컴포넌트 상태 머신은, 무선 수신 신호에 대응하는 기저대역 신호의 일 부분에 기초하여 원격 프로세싱 장치의 원격 프로세싱 상태를 예측하도록 구성된다.

[0010] 하나 이상의 실시예들에 있어서, 프로세싱 컴포넌트 상태 머신은, 로컬 프로세싱 장치의 하나 이상의 로컬 프로세싱 상태들에 기초하여 원격 프로세싱 장치의 원격 프로세싱 상태를 예측하도록 구성된다.

[0011] 하나 이상의 실시예들에 있어서, 로컬 프로세싱 장치와 원격 프로세싱 장치 사이의 통신들의 유선 통신 프로토콜과의 컴플라이언스를 유지하면서 로컬 프로세싱 장치와 원격 프로세싱 장치 사이의 통신들을 무선으로 터널링 하는 방법이 개시된다. 방법은: 무선 수신기에 의해, 원격 무선 터널링 장치로부터 무선 수신 신호를 수신하는 단계; 무선 수신기에 의해, 무선 수신 신호로부터 기저대역 신호를 생성하기 위하여 무선 수신 신호를 다운컨버

팅하는 단계; 프로세싱 컴포넌트 상태 머신에 의해, 기저대역 신호에 기초하여 원격 프로세싱 장치의 원격 프로세싱 상태를 예측하는 단계; 로컬 프로세싱 장치에 결합된 인터페이스 회로에 의해, 기저대역 신호 및 예측된 원격 프로세싱 상태에 기초하여 유선 통신 프로토콜에 따르는 출력 신호를 생성하는 단계; 및 인터페이스 회로에 의해, 유선 통신 프로토콜을 통해 로컬 프로세싱 장치로 출력 신호를 제공하는 단계를 포함한다.

도면의 간단한 설명

[0012]

본원에 개시된 실시예들의 교시들은 첨부된 도면들과 함께 다음의 상세한 설명을 고려함으로써 용이하게 이해될 수 있다.

도 1은 무선 터널링 시스템의 일 실시예를 예시한다.

도 2는 일 실시예에 따른, 원격 프로세싱 장치로부터 로컬 프로세싱 장치로 수신된 통신을 터널링하는 로컬 무선 터널링 장치의 예시적인 프로세스를 예시한다.

도 3은 일 실시예에 따른, 로컬 프로세싱 장치로부터 원격 프로세싱 장치로 수신된 통신을 터널링하는 로컬 무선 터널링 장치의 예시적인 프로세스를 예시한다.

도 4는 일 실시예에 따른 무선 터널링 장치의 상세 아키텍처를 예시한다.

도 5는 일 실시예에 따른 무선 터널링 장치의 예시적인 상태 천이 다이어그램을 예시한다.

도 6은 일 실시예에 따른 USB3.0 프로토콜에 따르는 무선 터널링 장치의 상세 아키텍처를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0013]

본원에서 설명되는 특징들 및 이점들이 완전히 포괄적인 것은 아니며, 구체적으로, 다수의 추가적인 특징들 및 이점들이 도면들, 명세서, 및 청구항들을 고려하여 당업자에게 자명해질 것이다. 또한, 명세서에서 사용되는 언어는 원칙적으로 가독성 및 교육적 목적들을 위해 선택되었으며, 본 발명의 내용을 제한하거나 또는 묘사하도록 선택되지 않았을 수 있다는 것을 주의해야만 한다.

[0014]

도면들 및 다음의 설명은 오로지 예시적으로 선호되는 실시예들과 관련된다. 다음의 논의로부터, 본원에 개시되는 구조들 및 방법들의 대안적인 실시예들이 본 발명의 원리들로부터 벗어나지 않고 이용될 수 있는 실행 가능한 대안들로서 용이하게 인식될 것이라는 것을 주의해야만 한다.

[0015]

이제 본 발명(들)의 몇몇 실시예들에 대해 상세한 참조가 이루어질 것이며, 이들의 예들은 첨부된 도면들에 예시된다. 어디에서든지 동일하거나 또는 유사한 실용적인 참조 번호들이 도면들에서 사용될 수 있으며, 동일하거나 또는 유사한 기능성을 표시할 수 있다는 것을 주의해야 한다. 도면들은 오로지 예시의 목적들을 위하여 실시예들을 도시한다. 당업자는 다음의 설명으로부터, 본원에서 예시되는 구조들 및 방법들의 대안적인 실시예들이 본원에서 설명되는 원리들로부터 벗어나지 않고 이용될 수 있다는 것을 용이하게 인식할 것이다.

[0016]

시스템 개괄

[0017]

본원의 실시예들은 주로, 호스트들, 디바이스들, 및 허브들을 포함하는, 연결형 토폴로지(connected topology)에서 임의적인 노드로 플러그(plug)될 수 있는 터널링 시스템의 맥락에서 설명된다. 일부 실시예들에 있어서, 터널링 시스템은 USB 3.0 시스템의 상황에서 동작할 수 있다. 그러나, 본원의 실시예들은 또한 다른 통신 프로토콜들 예컨대 USB 표준의 상이한 버전들 또는 완전히 상이한 프로토콜들 예컨대 HDMI, DisplayPort, 또는 다른 직렬 통신 프로토콜들을 사용하여 통신하기 위하여 사용될 수 있다.

[0018]

도 1은 무선 터널링 시스템(100)의 일 실시예를 예시한다. 무선 터널링 시스템(100)은 무선 링크(130)를 통해 제 2 컴퓨팅 시스템(150B)과 통신하는 제 1 컴퓨팅 시스템(150A)을 포함한다.

[0019]

일 실시예에 있어서, 무선 링크(130)는 60GHz 무선 링크를 포함한다. 무선 링크(130)는, 무선 터널링 장치들(120)이 서로 매우 인접하여(예를 들어, 몇 밀리미터 이내에) 존재하는 근거리 통신으로 한정될 수 있다. 무선 링크(130)를 통한 데이터 송신들은, 예를 들어, 초당 6 기가비트 또는 그 이상의 데이터 레이트를 가질 수 있다. 다른 실시예들에 있어서, 무선 링크는 장거리 통신들에 적절할 수 있거나 및/또는 다른 주파수 대역들에 대하여 구현될 수 있다.

[0020]

제 1 컴퓨팅 시스템(150A)은 유선 연결(116A)을 통해 무선 터널링 장치(120A)에 결합된 프로세싱 장치(110A)를 포함하며, 제 2 컴퓨팅 시스템(150B)은 유선 연결(116B)을 통해 무선 터널링 장치(120B)에 결합된 프로세싱 장치

(110B)를 포함한다. 무선 터널링 장치들(120A 및 120B)(본원에서 "무선 터널링 장치들(120)" 또는 "트랜시버들(120)"로도 지칭됨)은 무선 링크(130)를 통해 서로 통신하며, 프로세싱 장치들(110A 및 110B)(본원에서 "프로세싱 장치들(110)" 또는 "소스 장치들(110)"로도 지칭됨) 사이의 통신들을 터널링한다. 프로세싱 장치는 다른 전자 장치와 유선 통신 프로토콜에 따라 (양방향으로 또는 단방향으로) 데이터를 교환할 수 있는 전자 장치를 포함할 수 있다. 프로세싱 장치의 예들은, 소스 디바이스, 싱크 디바이스, 소스 디바이스와 싱크 디바이스 사이의 중간 디바이스, USB 호스트/디바이스, 저장 디바이스, 등을 포함한다. 일 실시예에 있어서, 무선 터널링 장치(120)는 프로세싱 장치(110)의 포트 또는 케이블(예를 들어, USB 포트 또는 케이블, HDMI 포트 또는 케이블, DisplayPort 포트 또는 케이블)에 결합될 수 있는 착탈가능 동글(dongle)로서 구현된다. 다른 실시예들에 있어서, 무선 터널링 장치(120)는 (예를 들어, 인쇄 회로 보드 상의 트레이스(trace)들을 통해) 프로세싱 장치(110)에 내부적으로 결합되거나 또는 (예를 들어, 접적 회로 내에서) 프로세싱 장치(110)와 완전히 통합될 수도 있다.

[0021] 컴퓨팅 시스템(150)(및 이의 컴포넌트들)은 아날로그 회로 컴포넌트, 디지털 로직, 소프트웨어, 또는 이들의 조합을 사용하여 구현될 수 있다. 일 실시예에 있어서, 컴퓨팅 시스템(150)의 하나 이상의 컴포넌트들은, 프로세서 및 프로세서에 의해 실행될 때 프로세서가 컴포넌트들에 기인하는 기능들을 수행하게끔 하는 명령어들을 저장하는 비-일시적인 컴퓨터-판독가능 저장 매체로서 구현될 수 있다. 대안적으로, 또는 부가적으로, 디지털 컴포넌트들은 애플리케이션 특정 접적 회로(application specific integrated circuit; ASIC), 필드-프로그램가능 게이트 어레이(field-programmable gate array; FPGA)로서, 또는 구현예들의 조합을 사용하여 구현될 수 있다.

[0022] 일 실시예에 있어서, 무선 터널링 시스템(100)은 USB, HDMI, DisplayPort, 또는 다른 직렬 통신 프로토콜들과 같은 통상적인 유선 통신들에 대한 대체를 제공한다. 예를 들어, 프로세싱 장치들(110A, 110B)이 전통적인 케이블을 통해 서로 직접적으로 통신하는 것이 아니라, 그 대신에 프로세싱 장치들(110A, 110B)은 그들의 개별적인 무선 터널링 장치들(120A, 120B)과 통신하고, 이들은 그런 다음 전통적인 유선 통신들을 사용하여 달성될 수 있는 속도들을 초과하는 속도들로 고속 점-대-점 직렬 무선 링크(130)를 통해 데이터를 터널링한다.

[0023] 프로세싱 장치들(110A, 110B)의 관점에서, 통신들은 마치 프로세싱 장치들(110A, 110B)이 통상적인 구성으로 직접적으로 연결된 것과 동일한 방식으로 구현될 수 있다. 따라서, 통상적인 프로세싱 장치(110A, 110B)에 대한 어떠한 수정도 반드시 요구되는 것은 아니다(예를 들어, 어떠한 소프트웨어 수정도 필수적이지 않다). 다시 말해서, 무선 터널링 장치들(120A, 120B) 및 이들 사이의 무선 링크(130)는 통상적인 케이블에 대한 직접 대체물로서 동작할 수 있다. 예를 들어, 각각의 무선 터널링 장치(120A, 120B)는 이를 그것의 개별적인 프로세싱 장치(110A, 110B)의 전통적인 케이블 인터페이스 내에 플러그하는 것을 가능하게 하는 인터페이스를 포함하며, 무선 터널링 장치(120A, 120B)는, 프로세싱 장치들(110A, 110B)이 직접적으로 연결된 것으로 보이도록 하는 통신을 가능하게 한다. 대안적인 실시예들에 있어서, 무선 터널링 장치들(120A, 120B)은 그들의 개별적인 프로세싱 장치들(110A, 110B)과 통합될 수 있다.

[0024] USB를 예로서 취하면, USB 인터페이스들을 갖는 전통적인 무선 장치들은 USB 프로토콜을 무선 장치 내에서 종료하고, 송신을 위하여 데이터를 상이한 무선 프로토콜로 재-인코딩한다. 전통적인 무선 장치들은 USB 트리 토폴로지에서 노드들(USB 허브들, USB 디바이스들 또는 USB 중계기들)로서 보일 수 있다. 이와 대조적으로, 무선 터널링 장치는 수정들 없이 그리고 USB 프로토콜 계층을 종료하지 않고 USB 링크-계층 데이터 트래픽이 매우 낮은 레이턴시로 송신되는 것을 가능하게 한다. 이러한 무선 터널링 장치들은 USB 토폴로지에서 보이지 않는다.

[0025] 일 실시예에 있어서, 각각의 무선 터널링 장치(120)는 무선 터널링 장치(120)가 결합된 프로세싱 장치(110)의 상대방(counterpart)의 상태들 및 동작들을 미러링(mirror)하기 위하여 그것의 연결된 프로세싱 장치(110)와 통신한다. 따라서, 예를 들어, 무선 터널링 장치(120A)가 화살표(118)에 의해 표시되는 바와 같이 프로세싱 장치(110B)의 상태들을 미러링하며, 무선 터널링 장치(120B)가 화살표(128)에 의해 표시되는 바와 같이 프로세싱 장치(110A)의 상태들을 미러링한다. 따라서, 무선 터널링 장치(120A)로부터 프로세싱 장치(110A)로 통신되는 데이터는 프로세싱 장치(110B)로부터 무선 터널링 장치(120B)로의 통신들을 미러링하며, 무선 터널링 장치(120B)로부터 프로세싱 장치(110B)로 통신되는 데이터는 프로세싱 장치(110A)로부터 무선 터널링 장치(120A)로의 통신들을 미러링한다.

[0026] 구체적으로, 무선 터널링 장치들(120)의 각각은 그것의 원격(즉, 상대방) 프로세싱 장치(110)의 동작 상태(예를 들어, 전력 상태 또는 다른 동작 상태)를 예측하며, 유선 연결(116)을 통해 예측된 상태에 따라 그것의 로컬 프로세싱 장치(110A)와 인터페이싱한다. 예를 들어, 프로세싱 장치(110B)는 데이터의 속도 또는 전력 관리 상태에

의존하여 유선 통신 프로토콜(예를 들어, USB)에 따라서 복수의 프로세싱 상태들 중 하나로 동작한다. 무선 터널링 장치(120A)는 프로세싱 장치(110B)의 동작 상태를 예측하고, 유선 연결(116A)을 통해 프로세싱 장치(110A)와 인터페이싱하기 위하여 프로세싱 장치(110B)의 예측된 상태를 미러링한다. 미러링된 상태는 프로세싱 장치(110B)의 동작 상태와 동일하거나 또는 실질적으로 유사할 수 있다. 일 측면에 있어서, 프로세싱 장치(110)의 프로세싱 상태들의 세트는, 표 4와 관련하여 추가로 상세하게 설명되는 바와 같이, 무선 터널링 장치(120)의 단일 상태 또는 더 적은 수의 상태들로 매핑되거나 또는 축약(collapse)될 수 있다.

[0027] 무선 터널링 장치(120)는 송신기(122), 수신기(124), 및 상태 머신(126)을 포함한다. 송신기(122)는 프로세싱 장치(110)로부터 데이터를 수신하고, 상이한 컴퓨팅 시스템(150)의 수신기(124)로 무선 링크(130)를 통해 데이터를 송신한다. 수신기(124)는 다른 컴퓨팅 시스템(150)의 송신기(122)로부터 무선 링크(130)를 통해 데이터를 수신하고, 수신된 데이터를 프로세싱 장치(110)에 제공한다. 상태 머신(126)은, 고 주파수 데이터를 송신하기 위한 고 전력 상태와 하나 이상의 저 전력 상태들 사이에서 무선 터널링 장치(120)를 스위칭함으로써 무선 터널링 장치(120)의 전력 상태를 제어하며, 이는 이하에서 추가로 상세하게 설명될 것이다. 무선 터널링 장치들(120)은 또한 터널링된 프로토콜 내에서 시그널링(signal)되는 저-전력 상태들을 모방한다. 일 실시예에 있어서, 무선 터널링 장치(120)는, 무선 데이터 링크(130)를 통해 데이터를 동시에 송신하고 수신할 수 있도록 전-이중 통신이 가능하다.

[0028] 예를 들어, 예시된 실시예에 있어서, 프로세싱 장치(110A)는 업스트림(upstream) 장치로서 구성되고 "호스트"로서 상태 머신(126A)에 따라 동작하며, 여기에서 프로세싱 장치(110B)는 다운스트림(downstream) 장치로서 구성되고 "디바이스"로서 상태 머신(126B)에 따라 동작한다. "호스트"로서 기능하는 프로세싱 장치(110A)는 "디바이스"로서 기능하는 프로세싱 장치(110B)의 동작들을 제어하거나 또는 이와 통신한다. 업스트림 무선 터널링 장치(120A)는 유선 연결(116A)을 통해 업스트림 프로세싱 장치(110A)(또는 "호스트")를 인터페이싱하며, 유사하게 다운스트림 무선 터널링 장치(120B)는 유선 연결(116B)을 통해 다운스트림 프로세싱 장치(110B)(또는 "디바이스")를 인터페이싱한다. 무선 터널링 장치들(120A 및 120B)은 무선 링크(130)를 통해 개별적인 프로세싱 장치들(110)의 스테이터스(status), 상태(state)들, 또는 제어 정보를 포함하는 데이터를 교환한다.

[0029] 일 실시예에 있어서, 무선 터널링 장치들(120A, 120B)은 실질적으로 동일한 장치들이다. 대안적으로, 무선 터널링 장치들(120A, 120B)은 상이한 상보적인 장치 유형들이며, 이들은 동일한 고 레벨 아키텍처들을 갖지만, 본원에서 설명되는 바와 같이 특정 아키텍처 또는 동작 특성에서 상이하다. 예를 들어, 일 실시예에 있어서, 제 1 무선 터널링 장치(120A)는 도킹 스테이션으로서 구현된 프로세싱 장치(110A)와 동작하도록 구성된 제 1 장치 유형을 포함하며, 반면 제 2 무선 터널링 장치(120B)는 모바일 장치로서 구현된 프로세싱 장치(110B)와 동작하도록 구성된 제 2 장치 유형을 포함한다. 일 실시예에 있어서, 전-이중 통신을 구현하기 위하여, 상이한 유형들의 상보적인 무선 터널링 장치들(120)은, 2개의 상이한 송신/수신 안테나 쌍들이 양 방향으로 동시에 동작할 수 있도록 상이한 안테나 편파(antenna polarization)를 갖는다. 예를 들어, 무선 터널링 장치(120A)는 유형 X 송신 안테나 및 유형 Y 수신 안테나를 가질 수 있으며, 반면 무선 터널링 장치(120B)는 상보적인 유형 Y 송신 안테나 및 유형 X 수신 안테나를 가질 수 있다. 추가로, 상이한 유형들의 무선 터널링 장치들(120)은 쌍 내의 무선 터널링 장치들(120) 중 하나의 전력 효율을 최적화하기 위하여 상이한 제어 기법들에 따라 동작할 수 있다. 예를 들어, 제 1 무선 터널링 장치(120A)가 도킹 스테이션과 동작하기 위하여 구성되고 제 2 무선 터널링 장치(120B)는 모바일 장치와 동작하기 위하여 구성될 때, 무선 터널링 장치들(120A, 120B)은, 도킹 스테이션에 의해 호스팅되는 무선 터널링 장치(120B)를 회생하여 모바일 장치에 의해 호스팅되는 무선 터널링 장치(120A)의 전력 소비를 낮추기 위해 비대칭적으로 동작할 수 있다. 도킹 스테이션이 전형적으로 연속적인 전원에 연결되며 반면 모바일 장치는 제한된 전력의 배터리에 의존하기 때문에, 이러한 트레이드오프(tradeoff)가 바람직 할 수 있다.

[0030] 일 실시예에 있어서, 무선 터널링 장치(120)와 연관된 장치 유형(및 이와 연관된 동작)은 무선 터널링 장치(120) 내에 영구적으로 설계될 수 있다. 대안적으로, 무선 터널링 장치(120)는 스위치, 제어 펌(즉, 칩의 제어 입력) 또는 레지스터 세팅에 기초하여 2개 이상의 장치 유형들 사이에서 구성이 가능할 수 있다. 상보적인 쌍 내의 무선 터널링 장치들(120A, 120B)의 상이한 구성들 사이의 아키텍처 및/또는 동작적 차이들이 이하에서 추가로 상세하게 설명된다.

[0031] 도 2는 일 실시예에 따른, 원격 프로세싱 장치(110B)로부터 로컬 프로세싱 장치(110A)로 수신된 통신을 터널링 하는 로컬 무선 터널링 장치(120A)의 예시적인 프로세스를 예시한다.

[0032] 로컬 무선 터널링 장치(120A)는 무선 링크(130)를 통해 원격 무선 터널링 장치(120B)로부터 무선 수신 신호를 수신한다(210). 무선 수신 신호는 제 1 주파수(예를 들어, ~60 GHz)로 수신된다. 무선 수신 신호는 원격 프로세

성 장치(110B)로부터의 원격 데이터 신호에 따라 무선 터널링 장치(120B)에 의해 생성된다. 원격 데이터 신호는 로컬 프로세싱 장치(110A)로 송신될 컨텐트 정보를 포함하며, 유선 통신 프로토콜(예를 들어, USB 프로토콜)을 따른다.

[0033] 무선 수신 신호를 수신하는 것에 응답하여, 로컬 무선 터널링 장치는 무선 수신 신호에 기초하여 기저대역 신호를 획득한다. 구체적으로, 로컬 무선 터널링 장치(120A)는 무선 수신 신호를 제 1 주파수보다 더 낮은 제 2 주파수(예를 들어, 수 Gbps)로 다운컨버팅한다(220). 다운컨버팅된 무선 수신 신호가 기저대역 신호이다.

[0034] 일 측면에 있어서, 로컬 무선 터널링 장치(120A)는 기저대역 신호에 기초하여 원격 프로세싱 장치(110B)의 상태를 예측한다(230). 기저대역 신호는 원격 프로세싱 장치(110B)의 이전 상태를 나타내는 상태 신호를 가지고 인코딩될 수 있다. 로컬 무선 터널링 장치(120A)는 원격 프로세싱 장치(110B)의 상태 신호를 획득하기 위하여 기저대역 신호를 디코딩하고, 원격 프로세싱 장치(110B)의 이전 상태에 기초하여 원격 프로세싱 장치(110B)의 상태를 예측할 수 있다. 다른 측면에 있어서, 로컬 무선 터널링 장치(120A)는, 무선 수신 신호에 대응하는 기저대역 신호의 일 부분에 기초하여 원격 프로세싱 장치의 원격 프로세싱 상태를 예측한다. 예를 들어, USB의 맥락에서, 로컬 무선 터널링 장치(120A)는 수신된 USB 데이터 패킷들을 분석함으로써 원격 프로세싱 상태를 예측할 수 있다. 또 다른 측면에 있어서, 로컬 무선 터널링 장치(120A)는 로컬 이벤트(예를 들어, 타임아웃 이벤트들과 같은 로컬 프로세싱 장치의 현재 또는 하나 이상의 이전의 로컬 프로세싱 상태들)에 기초하여 원격 프로세싱 장치의 원격 프로세싱 상태를 예측한다.

[0035] 로컬 무선 터널링 장치(120A)는 원격 프로세싱 장치(110B)의 상태를 미러링한다(240). 구체적으로, 로컬 프로세싱 장치(110A)를 인터페이싱하는 로컬 무선 터널링 장치(120A)는 원격 프로세싱 장치(110B)의 예측된 상태를 미러링하고, 기저대역 신호 및 미러링된 상태에 기초하여 미러링된 원격 데이터 신호를 생성한다(250). 미러링된 원격 데이터 신호는 원격 프로세싱 장치(110B)에서 생성된 원격 데이터 신호와 동일하거나 또는 실질적으로 유사하다. 예를 들어, 로컬 무선 터널링 장치(120A)는 예측된 상태에 따라 로컬 프로세싱 장치(110A)로 미러링된 원격 데이터 신호를 제공한다. 따라서, 원격 프로세싱 장치(110B)로부터 로컬 프로세싱 장치(110A)로의 통신의 터널링이 달성될 수 있다.

[0036] 도 3은 일 실시예에 따른, 로컬 프로세싱 장치(110A)로부터 원격 프로세싱 장치(110B)로의 통신을 터널링하는 로컬 무선 터널링 장치(120A)의 예시적인 프로세스를 예시한다.

[0037] 로컬 무선 터널링 장치(120A)는 로컬 프로세싱 장치(110A)로부터 로컬 데이터 신호를 수신한다(310). 로컬 데이터 신호는 원격 프로세싱 장치(110B)로 송신될 정보를 포함하며, 유선 통신 프로토콜(예를 들어, USB 프로토콜)을 따른다.

[0038] 일 측면에 있어서, 로컬 무선 터널링 장치(120A)는 로컬 프로세싱 장치(110A)의 하나 이상의 상태들을 획득하고, 로컬 프로세싱 장치(110A)의 하나 이상의 상태들을 하나 이상의 대응하는 상태들로 매핑한다(320). 로컬 무선 터널링 장치(120A)는 로컬 프로세싱 장치(110A)로부터 로컬 프로세싱 장치(110A)의 현재 동작 상태를 나타내는 신호를 수신한다. 대안적으로, 로컬 무선 터널링 장치(120A)는 수신된 로컬 데이터 신호(예를 들의 이의 값들)에 따라 로컬 프로세싱 장치(110A)의 동작상태 및/또는 로컬 프로세싱 장치(110A)의 이전의 동작 상태들을 결정한다.

[0039] 로컬 무선 터널링 장치(120A)는 로컬 데이터 신호 및 매핑된 상태에 기초하여 제 2 주파수(예를 들어, 수 Gbps)로 기저대역 신호를 생성한다(330). 일 측면에 있어서, 로컬 무선 터널링 장치(120A)는 매핑된 상태에 따라 기저대역 신호를 인코딩한다. 예를 들어, 로컬 무선 터널링 장치(120A)는 로컬 무선 터널링 장치(120A)의 매핑된 상태를 나타내는 상태 신호를 가지고 기저대역 신호를 인코딩한다.

[0040] 로컬 무선 터널링 장치(120A)는 제 1 주파수(예를 들어, 60 GHz)로 무선 송신 신호를 생성하기 위하여 제 2 주파수(예를 들어, 수 Gbps)의 기저대역 신호를 업컨버팅하고(340), 그런 다음 무선 링크(130)를 통해 원격 무선 터널링 장치(120B)로 무선 송신 신호를 송신한다(350).

[0041] 유익하게는, 무선 터널링 장치(120)는 원격 프로세싱 장치(110)의 동작 상태를 예측하고, 원격 프로세싱 장치(110)의 동작을 미러링한다. 따라서, 유선 통신 프로토콜의 데이터를 다른 프로토콜(예를 들어, 무선 통신 프로토콜)로 변환하거나 또는 무선 터널링 장치의 상이한 상태들 사이에서 천이하기 위한 임의의 지연들이 감소될 수 있다. 따라서, 무선 링크(130)를 통한 고속 데이터 레이트(예를 들어, 멀티-Gbps)를 갖는 2개의 프로세싱 장치들(110A, 110B) 사이의 통신의 매끄러운 터널링이 달성될 수 있다.

[0042] 상세 무선 터널링 장치 아키텍처

도 4는 일 실시예에 따른 무선 터널링 장치(120)의 상세 아키텍처를 도시한다. 도 4의 다이어그램이 도 1에 예시된 무선 터널링 장치들(120)에 대응할 수 있지만, 이는 일 실시예에 따른 장치들(120)의 동작을 양호하게 설명하기 위하여 추가로 세부적으로 일 예를 예시한다. 일 실시예에 있어서, 무선 터널링 장치(120)는 USB 3.0 속도로 터널링이 가능한 전-이중 고속 데이터-경로를 포함한다. 일 구현예에 있어서, 무선 터널링 장치(120)는 USB PHY(402), USB 디지털(404), 인코더(416), 송신기(420), 무선 컴포넌트 상태 머신(480), 디코더(454), 및 수신기(440)를 포함한다. USB 프로토콜에 따른 적절한 통신을 위하여 추가적인 컴포넌트들이 구현될 수 있다. 이러한 컴포넌트들은 함께 2개의 프로세싱 장치들(110) 사이의 통신을 터널링하도록 동작한다.

[0044] USB PHY(402)는, 일 실시예에 있어서 USB 3.0 전기 명세와 완전히 호환되며 4개의 상이한 USB 속도들: 초고속(super-speed)(5Gbps), 고속(high-speed)(480Mbps), 전속(full-speed)(12Mbps), 및 저속(low-speed)(1.5Mbps)을 지원하는 혼합-신호 인터페이스 회로(mixed-signal interface circuit)이다. 이는 USB 3.0 호스트 및 주변 애플리케이션들의 완전한 범위를 지원한다. USB PHY(402)는 PIPE3.0(SS를 위한) 및 UTMI+(HS/FS/LS를 위한)에 따라 디지털 인터페이스들을 제공한다. UTMI+는 FS/LS에 대한 2개의 인터페이스들: 표준 8-bit/16-bit 인터페이스 또는 비트-직렬 인터페이스를 제공한다. 본원에서 설명되는 터널링 아키텍처의 일 실시예는 2개의 무선 터널링 장치들(120)에 걸친 엔드-투-엔드(end-to-end) 레이턴시를 최소화하기 위하여 비트-직렬 인터페이스를 사용한다. 비트-직렬 인터페이스는 비트 데이터를 직렬화하고 역-직렬화(deserialize)하도록 역할하며, 그럼으로써 레이턴시를 감소시킨다.

[0045] USB PHY(402)는 USB 3.0 명세에 정의된 모든 저 전력 상태들: 초고속에 대한 U0/U1/U2/U3 및 HS/FS/LS에 대한 중지-재개(suspend-resume)에 대한 감소된 전력 소비를 구현한다. 이는 또한, 저 전력 상태들로부터 빠져 나오기 위하여, 명세에 정의된 바와 같은 저 주파수 주기적 시그널링(Low Frequency Periodic Signaling; LFPS)의 송신 및 수신을 지원한다.

[0046] USB 디지털(404)은 USB PHY(402)와 무선 컴포넌트들(예를 들어, 송신기(420) 및 수신기(440)) 사이를 인터페이싱하는 회로 컴포넌트이다. USB 디지털(404)은 USB PHY(402) 및 무선 컴포넌트들의 동작 상태들을 결정한다. USB 디지털(404)은 로컬 프로세싱 장치(110A)로부터 원격 프로세싱 장치(110B)로의 통신을 터널링할 때, 및 원격 프로세싱 장치(110B)로부터 로컬 프로세싱 장치(110A)로의 통신을 터널링할 때 둘 모두에서 기능한다.

[0047] 로컬 프로세싱 장치(110A)로부터 원격 프로세싱 장치(110B)로의 통신을 터널링하기 위하여, USB 디지털(404)은 USB PHY(402)를 통해 로컬 프로세싱 장치(110A)로부터 로컬 데이터 신호를 수신하고, 로컬 데이터 신호를 인코더(416)로 제공한다. 일 측면에 있어서, USB 디지털(404)은 로컬 프로세싱 장치(110A)의 로컬 프로세싱 상태를 나타내는 로컬 프로세싱 장치(110A)의 상태 신호를 생성한다. 로컬 프로세싱 장치(110A)의 상태 신호는, 상대방 무선 터널링 장치(120B)의 USB 디지털(404)이 로컬 프로세싱 장치(110A)의 동작 상태를 예측하는 것을 가능하게 한다. USB 디지털(404)은 로컬 데이터 신호에 기초하여 로컬 프로세싱 장치(110A)의 상태를 식별할 수 있다. 추가로, USB 디지털(404)은 매핑된 상태에 따라 상태 신호를 생성하고, 로컬 프로세싱 장치(110A)의 상태 신호 및 로컬 데이터 신호를 인코더(416)에 제공한다.

[0048] 인코더(416)는 로컬 프로세싱 장치(110A)의 상태 신호를 가지로 로컬 데이터 신호를 인코딩하며, 인코딩된 신호를 송신기(420)에 제공한다. 일 측면에 있어서, 로컬 데이터 신호는, 인코더 또는 스크램블러(scrambler)(간략함을 위하여 미도시)에 의해 인코딩 이전 또는 이후에 비-제로 DC 바이어스(bias)(0들 및 1들의 수가 동일하지 않음)과 같은 희망되지 않는 속성을 제거하기 위하여 스크램블링될 수 있다. USB 케이블을 통한 USB 초고속 데이터는 5Gbps의 원시(raw) 속도를 가지며, 이 밖의 20%는 8b/10b 코딩에 의해 기여된다. 이는 케이블 연관 비트 오류들로부터 보호하기 위하여 USB 표면에서 지정된다. 이러한 오버헤드(overhead)는 무선 송신을 위하여 제거되며, 초고속 데이터 대역폭이 4Gbps로 감소된다. 그 대신에 오류 정정 코드들이 무선 송신을 위하여 부가되며, 그 결과 무선 연관 비트 오류들이 검출되고 정정될 수 있다. 인코딩에 따라서, 수신기 측이 아마도 무선 송신에 의해 도입된 비트 오류들을 정정할 수 있다. 일 구현예에서 사용되는 FEC는 (232, 216) BCH(Bose-Chaudhuri-Hocquenghem) 코드(BCH 코드)이며, 이는 그 내부에서 각각의 출력 코드-워드가 주어진 216-비트 시퀀스에 대하여 232 비트를 갖는 순환 오류-정정 코드이다. BCH 코드는 유익하게 오류 정정 성능을 가지며, 낮은 인코딩 및 디코딩 레이턴시를 갖는다. 코드는 시스터메틱(systematic)할 수 있으며, 이는 첫 번째 216 비트 출력이 단지 입력 시퀀스로부터 복사된다는 것을 암시한다. 마지막 16 비트는 BCH 코드 생성기 매트릭스를 사용하여 인코딩될 수 있다.

[0049] 송신기(420)는 인코더(416)로부터 인코딩된 신호를 수신하고, 인코딩된 신호를 업컨버팅하여, 업컨버팅된 신호

를 무선으로 송신한다. 일 측면에 있어서, 송신기(420)는 고주파수 송신 회로(422) 및 저주파수 송신 회로(424)를 포함한다. 고주파수 송신 회로(422)는 높은 데이터 레이트(예를 들어, 6 Gbps)의 디지털 기저대역 신호를 RF 주파수(예를 들어, 60 GHz)로 업컨버팅하고 업컨버팅된 신호를 송신하기 위하여 사용된다. 예를 들어, 고주파수 송신 회로(422)는 USB 프로토콜, HDMI 프로토콜, DisplayPort 프로토콜, 또는 다른 통신 프로토콜에 따르는 높은 데이터 레이트의 신호를 업컨버팅하고, 업컨버팅된 신호를 무선 링크(130)를 통해 송신하는데 적절하다. 저주파수 송신 회로(424)는 낮은 데이터 레이트(예를 들어, ~100 kbps)의 디지털 기저대역 신호를 RF 주파수(예를 들어, 60 GHz)로 업컨버팅하고 업컨버팅된 신호를 송신하기 위하여 사용된다. 예를 들어, 저주파수 송신 회로(424)는, 동작 또는 무선 컴포넌트 상태 머신(480)의 상이한 상태들 사이에서의 천이, 또는 무선 컴포넌트의 전력 상태에 대한 제어 정보를 포함하는 낮은 데이터 레이트 신호를 업컨버팅하는데 적절하다. 고주파수 송신 회로(422)가 저주파수 송신 회로(424)보다 더 높은 데이터 레이트 신호를 송신하는 것이 가능하지만, 반면 고주파수 송신 회로(422)는 저주파수 송신 회로(424)보다 더 많은 회로 컴포넌트들을 포함할 수 있으며, 저주파수 송신 회로보다 더 많은 전력을 소비할 수 있다. 일 측면에 있어서, 고주파수 송신 회로(422) 및 저주파수 송신 회로(424) 중 하나는, 적절한 데이터 레이트 신호의 송신을 위하여 무선 컴포넌트 상태 머신(480)에 따라 선택된다.

[0050] 원격 프로세싱 장치(110B)로부터 로컬 프로세싱 장치(110A)로의 통신을 터널링하기 위하여, 수신기(440)는 다른 무선 터널링 장치(120)의 송신기로부터 무선 수신 신호를 수신하고, 기저대역 신호를 획득하기 위하여 무선 수신 신호를 다운컨버팅한다. 일 측면에 있어서, 수신기(440)는 고주파수 수신 회로(442) 및 저주파수 수신 회로(446)를 포함한다. 고주파수 수신 회로(442)는 RF 주파수(예를 들어, 60 GHz)를 높은 데이터 레이트(예를 들어, 6 Gbps) 디지털 기저대역 신호를 다운컨버팅하기 위하여 사용된다. 저주파수 수신 회로(446)는 RF 주파수(예를 들어, 60 GHz)를 낮은 데이터 레이트(예를 들어, ~100 kbps) 디지털 기저대역 신호를 다운컨버팅하기 위하여 사용된다. 고주파수 수신 회로(442)가 저주파수 수신 회로(446)보다 더 높은 데이터 레이트 신호를 다운컨버팅하는 것이 가능하지만, 반면 고주파수 수신 회로(442)는 저주파수 수신 회로(446)보다 더 많은 회로 컴포넌트들을 포함할 수 있으며, 저주파수 수신 회로(446)보다 더 많은 전력을 소비할 수 있다. 일 측면에 있어서, 고주파수 수신 회로(442) 및 저주파수 수신 회로(446) 중 하나는, 적절한 데이터 레이트 신호의 수신을 위하여 무선 컴포넌트 상태 머신(480)에 따라 선택된다.

[0051] 디코더(454)는 수신기(440)로부터 다운컨버팅된 신호를 수신하고, 복원된 신호를 디코딩한다. 하나의 접근방식에 있어서, 다운컨버팅된 신호는 디코더(454)에 의한 디코딩 이전 또는 이후에 디스크램블링될 수 있다(디스크램블러(descrambler)는 간략함을 위하여 도시되지 않음). 디코더(454)는 원격 프로세싱 장치(110B)의 이전 상태를 나타내는 원격 프로세싱 장치(110B)의 상태 신호 및 원격 데이터 신호를 획득하기 위하여 다운컨버팅된 신호를 디코딩할 수 있다. 일 실시예에 있어서, 경-판정(hard-decision) 기반 BCH 디코더가 구현된다. 디코더(454)는 다운컨버팅된 신호 내의 임의의 비트 오류를 검출하고 정정할 수 있다. (232, 216) BCH 코드는 232-비트 코드-워드 내에서 최대 2 비트 오류들을 정정할 수 있다. 이러한 코딩 기법은 독립적이고 랜덤한 비트 오류들을 개선한다. 디코더(454)는 USB 디지털(404)로 원격 프로세싱 장치(110B)의 상태 신호 및 원격 데이터 신호를 제공한다.

[0052] 다시 USB 디지털(404)을 참조하면, USB 디지털(404)은 디코더(454)로부터 원격 프로세싱 장치(110B)의 상태 신호 및 원격 데이터 신호를 수신한다. USB 디지털(404)은 원격 프로세싱 장치(110B)의 상태 신호에 기초하여 원격 프로세싱 장치(110B)의 동작 상태를 예측할 수 있다. 원격 프로세싱 장치(110B)의 상태 신호에 기초하여, USB 디지털은, 예를 들어, 원격 프로세싱 장치(110B)와 로컬 프로세싱 장치(110A) 사이의 통신 지연, 원격 프로세싱 장치(110B) 또는 로컬 프로세싱 장치(110A)의 이전 동작 상태들, 원격 데이터 신호 또는 이 둘의 조합을 고려함으로써, 원격 프로세싱 장치(110B)의 동작 상태를 예측한다. USB 디지털(404)은 원격 프로세싱 장치(110B)의 예측된 상태에 따라 USB PHY(402)를 구성하고, 무선 터널링 장치들(120A 및 120B)의 임의의 지연이 회피될 수 있는 방식으로 USB PHY(402)를 통해 로컬 프로세싱 장치(110A)로 원격 데이터 신호를 제공한다.

[0053] 일 실시예에 있어서, USB 디지털(404)은 원격 프로세싱 장치(110B)의 동작 상태를 미러링하기 위한 프로세싱 컴포넌트 상태 머신(408)을 포함한다. 프로세싱 컴포넌트 상태 머신(408)은 USB 3.0 명세로부터 링크 트레이닝 및 시스템 상태 머신(Link Training and System Status Machine; LTSSM)의 적용된 구현예를 포함할 수 있다. USB 3.0 명세의 링크 트레이닝 및 시스템 상태 머신(Link Training and System Status Machine; LTSSM)은 링크 전력 관리 및 링크 연결에 대하여 정의된 상태 머신이다. 추가적으로, 프로세싱 컴포넌트 상태 머신(408)은 USB 2.0 명세로부터의 리셋 프로토콜 상태 머신(Reset Protocol State Machine; RPSM)을 포함할 수 있다. 프로세싱 컴포넌트 상태-머신(408)은, 도 1에 도시된 바와 같은, 무선 링크의 반대 측 상의 USB 호스트/디바이스/허브의

LTSSM 상태 또는 RPSM 상태를 추적하도록 설계된다. 프로세싱 컴포넌트 상태 머신(408)은 원격 프로세싱 장치(110B)의 동작 상태를 예측하고, 원격 프로세싱 장치(110A)의 상태를 미러링한다.

[0054] 일 측면에 있어서, 프로세싱 컴포넌트 상태 머신(408)은 그 자체의 하나 이상의 상태들을 무선 컴포넌트 상태 머신(480)의 하나 이상의 대응하는 상태들에 매핑하며, 무선 컴포넌트 상태 머신(480)의 상태를 구성한다. 무선 컴포넌트 상태 머신(480)은, 무선 컴포넌트의 전력 효율을 개선하기 위하여 송신기(420) 및 수신기(440)의 무선 컴포넌트들의 전력 상태를 제어한다. 예를 들어, 고주파수 송신 회로(422)의 전력 소비가 저주파수 송신 회로(424)에 비하여 상대적으로 높기 때문에, 무선 컴포넌트 상태 머신(480)은 고주파수 송신 회로(422)가 사용되지 않을 때 저주파수 송신 동안 고주파수 송신 회로(422)를 턴 오프하거나 또는 저 전력 상태에서 동작하도록 고주파수 송신 회로(422)를 제어할 수 있다. 고주파수 송신들 동안, 저주파수 송신 회로(424)는 전원이 차단될 수 있다. 유사하게, 고주파수 수신 회로(442)의 전력 소비가 저주파수 수신 회로(446)에 비하여 상대적으로 높기 때문에, 무선 컴포넌트 상태 머신(480)은 고주파수 수신 회로(442)가 사용되지 않을 때 저주파수 수신 동안 고주파수 수신 회로(442)를 턴 오프하거나 또는 저 전력 상태에서 동작하도록 고주파수 수신 회로(442)를 제어할 수 있다. 고주파수 수신 동안, 저주파수 수신 회로(446)는 전원이 차단될 수 있다.

[0055] 추가적으로, 무선 컴포넌트 상태-머신(480)은 예를 들어 다음을 포함하는 다수의 시스템 기능들을 제어한다:

(1) USB 호스트/디바이스의 부착 및 분리를 검출하는 것

(2) USB 전력 상태에 기초하여 무선 블록들 & USB PHY의 전력 상태를 제어하는 것

(3) 무선 링크에 걸친 링크 응답을 재생함으로써 USB 호스트 및 디바이스의 동기성을 보장하는 것.

[0056] [0057] [0058] [0059] 도 5는 일 실시예에 따른 USB 데이터를 터널링할 수 있는 무선 터널링 시스템에 대한 예시적인 상태 천이 다이어그램을 예시한다. 무선 컴포넌트 상태 머신(480)의 각각의 상태는, 이하의 표 2에 도시되는 바와 같이 프로세싱 컴포넌트 상태 머신(408)의 대응하는 상태에 매핑될 수 있다. 이러한 실시예에 있어서, 5개의 가능한 전력 상태들이 이용이 가능하다: W0 상태(502), W2 상태(506), W3 상태(508), 근접 검출 상태(510), 및 슬립(sleep) 상태(504). 일 측면에 있어서, 무선 컴포넌트 상태 머신(480)은 도 5에 도시된 전력 상태 중 하나의 상태로 동작한다.

[0060] W0 상태(502)는, 고주파수 송신 회로(442), 고주파수 수신 회로(442), 및 연관된 컴포넌트들이 인에이블(enable)되며, 무선 터널링 장치(120)가 고주파수 직렬 데이터(예를 들어, USB 데이터)를 능동적으로 송신하거나 또는 송신을 위해 이용가능하고 준비가 된, 고 전력 상태를 나타낸다. W0 상태에서, 고주파수 송신 회로(422) 및 고주파수 수신 회로(442)가 턴 온되며, 무선 터널링 장치(120)가 USB 데이터를 능동적으로 터널링할 수 있다. 다른 장치에 대한 근접성이 상실되는 경우, 무선 터널링 장치(120)는 근접 검출 상태로 천이한다. 근접 검출 상태(510)에서, 고주파수 송신 회로(422) 및 고주파수 수신 회로(442)는 턴 오프된다. 저주파수 송신 회로(424) 및 저주파수 수신 회로(446)는 다른 장치에 대한 근접성을 주기적으로 체크하기 위하여 턴 온되며, 사용되지 않을 때 턴 오프된다. 무선 터널링 장치들(120A 및 120B)은, 근접 검출이 성공했지만 프로세싱 장치(110)가 부착되지 않은 것으로 결정될 때, W0 상태로부터 슬립 상태(504)로 진입할 수 있다. 이러한 결정은 W0 상태(502)에서 이루어진다. 슬립 상태(504)에서, 전력 효율을 위하여 오로지 "항상-온(always-on)" 블록들만이 구동되며 다른 컴포넌트들을 턴 오프된다. 무선 터널링 장치는 미리-정의된 시간 동안 슬립 상태(504)로 남아 있으며, 그런 다음 무선 근접성이 유지된다는 것을 확인하기 위하여 다시 근접 검출 상태(510)로 진행한다. 인접 장치가 검출되는 경우, 무선 터널링 장치(120)는 다시 W0 상태(502)로 천이하며, 여기에서 프로세싱 장치의 부착이 체크된다. W2 및 W3 상태들(506, 508)은, 무선 터널링 장치 쌍(120A/120B)이 무선 근접 상태이며, 프로세싱 장치들(110A/110B)이 부착된 상태이지만 프로세싱 장치들(110)이 저전력 상태이거나 또는 능동적으로 데이터를 통신하고 있지 않을 때, 진입된다. 예를 들어, W2 상태(506)는 프로세싱 장치(110)가 USB 3.0 초고속의 "U2" 저-전력 상태에 있을 때 진입되며, W3 상태(508)는 프로세싱 장치(110)가 USB 3.0 초고속의 "U3" 상태이거나 또는 USB 2.0 초고속의 "중지" 상태일 때 진입된다.

[0061] 도 5의 각각의 아크(arc)는 상태들 사이의 가능한 천이를 나타낸다. 상태들 사이에서 천이하기 위한 조건들이 표 1에서 요약되며, 이하에서 더 상세하게 설명된다.

아크	로컬 장치 요건들	원격 장치 요건들
A	USB 2.0 분리됨 및 USB 3.0 디세이블(disable)됨	제안된_링크_상태 == 슬립
b1	USB 2.0 분리되거나 또는 중지됨 및 USB 3.0 U2 상태.	제안된_링크_상태 == W2
b2	USB 2.0 USB 3.0 로컬적인 또는 원격적인 웨이크업 이벤트	n a
c1	USB 2.0 분리되거나 또는 중지됨 및 USB 3.0 U3 상태	제안된_링크_상태 == W3
c2	USB 2.0 USB 3.0 로컬적인 또는 원격적인 웨이크업 이벤트	n a
d1	슬립 타이머 타임 아웃	n a
E	N 초 동안 무선 캡-얼라이브(keep-alive) 신호가 보이지 않음	n a
F	N 초 동안 무선 캡-얼라이브 신호가 보이지 않음	n a
g2	W0 으로: 근접이 검출됨	n a
g1	HF 동기화 상태 머신에 의한 HF 무선 링크가 상실됨.	n a

[0062]

표 1: 무선 터널링 장치에 대한 상태 천이

[0063]

(예를 들어, 아크들 a, b1, c1, 및 g1을 통해) W0 상태(502)를 벗어나는 천이들은 로컬 무선 터널링 장치의 상태뿐만 아니라 로컬 무선 터널링 장치에 근접한 원격 무선 터널링 장치의 상태 둘 모두에 의존한다. 로컬 장치의 상태를 원격 장치로 통신하기 위하여, 그리고 이의 역을 수행하기 위하여, 신호 제안된_링크_상태는, W0 상태(502)에서 로컬 장치 조건들에 의해 지시되는 새로운 상태로의 천이를 나타낼 때 장치들 사이에서 주기적으로 송신된다. 예를 들어, 일 실시예에 있어서, 신호 제안된_링크_상태는 그것의 조건들에 기초하여 로컬 장치에 의해 통지되는 상태를 인코딩하는 2-비트 신호이다(예를 들어, '0'은 W0을 나타내고, '1'은 W2를 나타내며, '2'는 W3을 나타내고, 및 '3'은 슬립을 나타낸다). 신호 제안된_링크_상태는 W0 상태(502)일 때 무선 링크를 통해 주기적으로 갱신되고 교환된다.

[0065]

장치가 프로세싱 장치(110)가 디세이블되거나 또는 분리되었다는 것을 검출할 때, 장치는 슬립 상태(504)로의 천이(아크 a)를 통지한다. 장치는, 이것이 USB 2.0이 분리되었거나 또는 중지되었다는 것 및 USB 3.0이 저-전력 상태에 진입하였다는 것을 검출할 때, W2 상태(506)로의 천이(아크 b1)를 통지한다. 장치는, 이것이 USB 2.0이 분리되었거나 또는 중지되었다는 것 및 USB 3.0이 U3 상태에 진입하였다는 것을 검출할 때, W3 상태(508)로의 천이(아크 c1)를 통지한다. 장치는, HF 무선 링크(즉, 고주파수 송신 및 수신 데이터 경로들을 사용하는 무선 링크(130))가 상실될 때 근접 검출 상태(510)로의 천이를 통지한다.

[0066]

W0을 벗어나는 상태 변화는 오로지, 무선 링크(130)의 양 측들 둘 모두가 동일한 저 전력 상태(예를 들어, 슬립, W2, 또는 W3)를 통지한 이후에만 발생한다. 그렇지 않다면, 장치들 둘 모두는 W0 상태(502)로 남아 있다. W0 상태(502)에서, 제안된_링크-상태의 값이 주기적으로 송신된다. 로컬 및 원격 장치들 둘 모두가 동일한 저 전력 상태를 통지한 이후에, 장치들일 그 상태로 천이한다.

[0067]

W2 상태(506)로부터 W0 상태(502)로 빠져 나오는 것(아크 b2)은 이탈(exit) 이벤트를 전송하는 업스트림 또는 다운스트림 프로세싱 장치에 의해 트리거(trigger)된다. 예를 들어, USB 3.0에서, 이탈 이벤트는 무선 터널링

장치(120)에 대한 U2 이탈 LFPS(Low Frequency Periodic Signaling)를 포함할 수 있다. W2 이탈이 일어나게 하기 위하여, 무선 터널링 장치(120)는 기초 직렬 프로토콜의 요건들(예를 들어, USB 3.0 링크들에 대하여 2ms)을 충족시키기에 충분히 낮은 레이턴시를 갖는 핸드쉐이크(handshake) LFPS를 다시 송신한다. 일 구현예에 있어서, 빠른 W2 이탈은, W2 상태(506)에서 동작할 때 모든 위상-잠금 루프(phase-locked loop; PLL)들을 전력이 공급되는 상태로 유지함으로써 가능해 진다.

[0068] 일 예에 있어서, 프로세싱 컴포넌트 상태 머신(408)의 상태들은, 아래의 표 2에 도시되는 바와 같이, 무선 컴포넌트 상태 머신(480)의 상태들에 매핑된다.

LTSSM 상태들 (USB3.0 명세로부터)	무선 전력 상태들
U1	W0
U2	W2
U3	W3
SS.Disabled, SS.Inactive, Rx.Detect	W0 슬립 사이의 루프
U0 및 모든 다른 활성 상태들	W0

[0069]

표 2: USB3.0 LTSSM 상태들과 무선 전력 상태들 사이의 매핑

[0071] USB3.0에 대한 무선 터널링 장치 아키텍처

[0072] 도 6은 일 실시예에 따른 USB3.0 프로토콜에 따르는 무선 터널링 장치의 아키텍처를 도시한다. USB3.0 표준은 4 개의 상이한 전송 속도들에 대한 지원을 규정한다: 초고속(본원에서 "SS"로도 지칭됨), 고속(본원에서 "HS"로도 지칭됨), 전속(본원에서 "FS"로도 지칭됨) 및 저속(본원에서 "LS"로도 지칭됨). 도 6은 USB PHY(402) 및 USB 디지털(404)를 통한 4개의 모든 USB 속도들에 대한 데이터흐름을 도시한다.

[0073] USB PHY(402)의 일 측 상에 프로세싱 장치(110)와의 케이블 인터페이스가 존재한다. USB PHY(402)의 다른 측 상에 USB 디지털(404)에 대한 디지털 데이터 인터페이스가 존재하며, 이는 케이블 인터페이스보다 더 낮은 주파수로 동작한다. USB PHY(402)는 초고속 USB(601) 기능의 모든 측면들에 대한 지원을 제공하며, 산업 표준 PIPE 인터페이스(611)를 통해 USB 디지털(404)과 인터페이싱한다. 이러한 양-방향 인터페이스는 2개의 버스들 - 각기 USB PHY(402)로 들어오는 그리고 이를 벗어나는 데이터에 대한 버스들을 포함한다. PIPE 인터페이스(611)에서 버스 폭은 16 또는 32 비트이다. 일 실시예에 있어서, 16-비트 버스 폭은 레이턴시를 감소시키기 위하여 이용된다. 유사하게, 케이블 인터페이스로부터의 고속 데이터(602)는 산업-표준 UTMI 인터페이스를 사용하여 USB 디지털(404)과 인터페이싱된다. 가장 통상적인 USB PHY 구현예들에 있어서, 전속 및 저속 데이터(603)는 UTMI 인터페이스(612) 및 직렬 인터페이스(613) 상에 둘 모두가 제공된다. 일 측면에 있어서, UTMI 인터페이스(612)는 디지털 설계 파이프라인들을 설계하기 위하여 이용된다. 그러나, UTMI 인터페이스는 8 또는 16-비트 너비 버스들을 포함하며, FS/LS 데이터 비트들의 직렬화 또는 역직렬화를 위한 사이클들에 기인하는 큰 레이턴시들을 겪을 수 있고, 그럼으로써 이를 저-레이턴시 터널링 설계에 부적절하게 만든다. 이러한 큰 레이턴시를 극복하기 위하여, 일 실시예에 있어서, 직렬 인터페이스(613)가 USB 디지털(404)과 FS/LS 데이터(603)를 인터페이싱하기 위하여 이용된다.

[0074] USB 디지털(404)은 SS, HS, FS, 및 LS에 대한 컴포넌트들을 포함한다. 초고속 서브시스템(620)은 SS 수신 데이터-경로 블록(622) 및 SS 송신 데이터-경로 블록(623), 및 데이터-경로 블록들의 동작을 제어하는 USB 3.0 초고속 상태 머신(621)을 포함하며, 이는 이하에서 설명될 것이다. 유사하게, HS/FS/LS 서브시스템(630)은, HS 수신 데이터-경로 블록(632), HS 송신 데이터-경로 블록(633), 직렬 수신 데이터-경로 블록(634), 직렬 송신 데이터-경로 블록(635), 및 USB2.0 상태 머신(631)을 포함한다. USB2.0 상태 머신(631)은 HS/FS/LS 서브시스템(630) 내의 이러한 데이터 경로 블록들의 동작을 제어하며, 이는 이하에서 설명될 것이다. 임의의 주어진 세션에서, 터널링 시스템은, 프로세싱 장치(110)와 무선 터널링 장치(120) 사이의 검출된 상태에 의존하여 HS, FS, 또는 LS 모드들 중 하나로 동작한다. 특정 USB 연결 토폴로지들에 있어서, SS 및 HS/FS/LS 중 하나는, 예를 들어, 무선 터널링 장치들의 쌍이 USB3.0 호스트인 프로세싱 장치와 USB3.0 허브인 프로세싱 장치 사이에 존재할 때, 동

시에 활성화될 수 있다.

[0075] USB 디지털(404)은, 병렬로 동작할 수 있는 2개의 인터페이스들: 초고속(651)에 대하여 하나, 및 HS/FS/LS(652)에 대하여 다른 하나를 사용하여 무선 송신을 위하여 인코더(416)에 데이터를 제공한다. 인코더(416)는 고정된 프레임 구조에 따라서 이러한 인터페이스들을 모두로부터의 데이터를 패킹(pack)하며, 무선 송신을 위하여 단일 데이터 스트림을 송신기(420)로 제공한다. 유사하게, 디코더(454)와의 USB 디지털(404)의 인터페이스는 초고속(653) 데이터에 대한 하나의 인터페이스, 및 HS/FS/LS(654) 데이터에 대한 다른 인터페이스를 포함한다. 디코더(454)는 무선 수신기(440)로부터 데이터 스트림을 수신하고, 고정된 프레임 구조체 따라 데이터 스트림을 언패킹(unpack)하며, 초고속(653) 데이터 및 HS/FS/LS(654) 데이터에 대한 인터페이스 둘 모두 상에 동시에 데이터를 제공한다.

[0076] USB 프로토콜에 대한 예시적인 상태 머신 구현예

[0077] USB 3.0 초고속 상태 머신

[0078] USB 3.0 명세의 링크 트레이닝 및 시스템 상태 머신(Link Training and System Status Machine; LTSSM)은 링크 전력 관리 및 링크 연결에 대하여 정의된 상태 머신이다. 명세는, 표 3에서 요약되는 바와 같은, 특정 기능들에 대하여 24개의 서브-상태들을 갖는 12개의 상태들을 정의한다.

LTSSM 서브-상태들(24)	LTSSM 상태들(12)	기능
U0	U0	SS 패킷 전송이 진행 중인 상태
U1	U1	짧은 이탈 레이턴시를 갖는 저 전력 상태
U2	U2	U1 보다 더 많은 이탈 레이턴시를 갖는 저 전력 상태
U3	U3	최고의 이탈 레이턴시를 갖는 저 전력 상태
SS.Inactive.Disconnect.Detect	SS.Inactive	링크가 비-동작가능 상태이며 시스템 소프트웨어 개입이 필요한 링크 오류 상태
SS.Inactive.Quiet		
Rx.Detect.Reset	Rx.Detect	USB 포트가 SS 링크 파트너가 존재하는지 여부를 결정하려고 시도하는 단계로서, 존재의 검출 시에 링크 트레이닝이 시작된다
Rx.Detect.Active		
Rx.Detect.Quiet		
SS.Disabled.Default	SS.Disabled	초고속 연결이 디세이블되고, 링크는 USB 2.0 모드 하에서 동작할 수 있다
SS.Disabled.Error		
Compliance Mode	Compliance Mode	송신기 컴플라이언스 테스트를 가능하게 하기 위한 상태
Loopback.Active	Loopback	비트 오류 테스트를 가능하게 하기 위한 상태
Loopback.Exit		
Recovery.Active	Recovery	저 전력 상태를 이탈한 이후 또는 링크가 U0 속성으로 동작하지 않는다는 것을 검출한 이후, 또는 링크 파트너가 링크 동작의 모드를 변화시키는 것을 결정한 이후 링크를 재트레이닝하기 위한 상태
Recovery.Configuration		
Recovery.Idle		

[0079]

Hot Reset.Active	Hot Reset	다운스트림 포트가 그것의 업스트림 포트를 리셋하는 것을 허용하도록 정의된 상태
Hot Reset.Exit		
Polling.LFPS	Polling	2 개의 링크 파트너들이 그들의 SS 송신기들 및 수신기들을 트레이닝시키고, 동기화하며, 패킷 전송을 위해 준비시키기 위해 정의된 상태
Polling.RxEQ		
Polling.Active		
Polling.Configuration		
Polling.Idle		

[0080]

[0081] 표 3: USB 3.0 명세로부터의 LTSSM 상태들 및 서브-상태들

[0082] USB 디지털(404) 내부의 25-상태 USB3.0 초고속 상태 머신(631)은 일 실시예에 따른 24-상태 LTSSM를 최적화하고 무선 터널링 위해 적응시킴으로써 구현된다. 이는 표 4에 도시된다. USB3.0 초고속 상태 머신(631)은 다음 중 하나에 의해 LTSSM로부터 유도된다: 다수의 서브-상태들을 하나로 축약하는 것, 단일 서브-상태를 복수의 상태들로 분할하는 것, 또는 새로운 상태를 추가하는 것. 이러한 유도가 표 4에 도시된다.

	USB 3.0 초고속 상태 머신 (무선 터널링을 위한)	LTSSM 서브-상태들 USB3.0 명세로부터	코멘트
1	POWER_OFF	SS.Disabled.Default (VBUS)	분할된 LTSSM 서브-상태
2	SS.Disabled.Default	SS.Disabled.Default (Clock)	분할된 LTSSM 서브-상태
		SS.Inactive.Quiet	축약된 LTSSM 서브-상태
		SS.Inactive.Disconnect.Detect	축약된 LTSSM 서브-상태
3	SS.Disabled.Error	SS.Disabled.Error	
4	Rx.Detect.Reset	Rx.Detect.Reset	
5	Rx.Detect.Active	Rx.Detect.Active	
6	Rx.Detect.Quiet	Rx.Detect.Quiet	
7	Rx.Detect.Remote		부가된 새로운 상태
8	Polling.LFPS	Polling.LFPS	
9	Polling.RxEQ	Polling.RxEQ	
10	PollingTraining	Polling.Active	축약된 3 개의 LTSSM 서브-
		Polling.Config	
		Polling.Idle	상태들
11	NonPollingTraining	Recovery.Active	
		Recovery.Configuration	축약된 5 개의 LTSSM 서브-
		Recovery.Idle	
		Hot Reset.Active	상태들
		Hot Reset.Exit	
12	U0		
13	U0_Exit	U0	2 개로 분할된 LTSSM 서브-상태
14	U1		
14	U1_Exit	U1	2 개로 분할된 LTSSM 서브-상태
16	U2		
17	U2_Exit	U2	2 개로 분할된 LTSSM 서브-상태
18	U3		
19	U3_Exit	U3	2 개로 분할된 LTSSM 서브-상태

[0083]

20	Compliance Mode	Compliance Mode	
21	Loopback.Active.Local		2 개로 분할된 LTSSM 서브-
22	Loopback.Active.Through	Loopback.Active	상태
23	Loopback.Exit.Local		2 개로 분할된 LTSSM 서브-
24	Loopback.Exit.Through	Loopback.Exit	상태
25	IdleTraining		부가된 새로운 상태

[0084]

표 4: 무선 터널링을 위한 USB3.0 초고속 상태 머신의 상태들

[0085]

USB3.0 초고속 상태-머신은, 도 1에 도시된 바와 같은, 무선 링크의 반대 측 상의 USB 호스트/디바이스/허브의 LTSSM 상태를 추적하도록 설계된다. 따라서, 상태 천이들은 다음의 3개의 유형의 입력들 중 하나에 기초하여 이루어진다:

[0086]

원격 무선 터널링 장치의 USB3.0 초고속 상태 머신으로부터 무선을 통해 수신된 시그널링 정보(예를 들어, 표 7로부터의 remote.RX_SIG_POWEROFF),

[0087]

원격 무선 터널링 장치로부터 무선으로 통해 수신된 USB 패킷 데이터(remote.data로서 표시됨), 및

[0088]

타임아웃들과 같은 로컬적으로 생성된 신호들/이벤트들(예를 들어, 표 7의 timeout_12ms).

[0089]

표 5는 사용되는 모든 초고속 시그널링 정보의 리스트를 요약한다. 표 6은 무선을 통해 시그널링 정보를 송신하기 위한 패킷 구조를 도시한다. 초고속 시그널링 정보가 인코딩되며, 대역-내 페이로드(payload)로서 무선을 통해 통신된다. 시그널링 정보는, 송신될 초고속 패킷 데이터가 존재하지 않을 때라면 언제든지 무선으로 통해 전송될 수 있다.

	시그널링	인코딩된 값 (8 비트 - 바이너리)	코멘트
1	RX_SIG_POWEROFF	0000_00xx	0=Rx 유효, 1=Rx 전기적 아이들(Idle), 2=전력 존재
2	RX_SIG_LFPS	0000_0100	LFPS 시작
3	RX_SIG_DATA	0000_0101	데이터 시작
4	RX_SIG_IDLE_RX_SIG_POWERON	0000_0110	아이들 시작
5	RX_DETECT_SUCCESS	0001_0000	검출 패스(pass) 수신
6	RX_DETECT_FAIL	0001_0001	검출 실패 수신
7	RX_LFPS_POLLING	0010_0000	유형 Polling LFPS 를 가지고 LFPS 종료
8	RX_LFPS_PING	0010_0001	유형 Ping LFPS 를 가지고 LFPS 종료
9	RX_LFPS_WARMRESET	0010_0010	유형 Warmreset LFPS 를 가지고 LFPS 종료
10	RX_LFPS_U1EXIT	0010_0011	유형 U1 Exit LFPS 를 가지고 LFPS 종료
11	RX_LFPS_U2EXIT	0010_0100	유형 U2 Exit LFPS 를 가지고 LFPS 종료
12	RX_LFPS_U3WAKEUP	0010_0101	유형 U3 Wakeup LFPS 를 가지고 LFPS 종료
13	RX_LFPS_ERROR	0010_0110	유효하지 않은 LFPS 유형을 가지고 LFPS 종료
14	TSEQ_START	0011_0000	TSEQ 시작

[0091]

표 5: 초고속 시그널링 정보

바이트 3	바이트 2	바이트 1	바이트 0
K23.7 (EPF)	K26.8 (ESC)	K26.8 (ESC)	K26.8 (ESC)
Dxx.x	Dxx.x	Dxx.x	Dxx.x

[0093]

표 6: 시그널링 정보 패킷 구조

[0094]

표 6에서, Dxx.x는 인코딩된 시그널링 심볼이며 - 표 5로부터의 인코딩된 값들 중 하나이다. 일 실시예에 있어서, 무선 오류들에 대한 복원력을 개선하기 위하여 Dxx.x를 4회 복제하는 것이 이용된다.

[0095]

표 7은 25-상태 USB 3.0 초고속 상태 머신(631)에 대한 천이 조건들 및 다음 상태들을 요약하며, 이는 일 실시 예에 따른 USB3.0 초고속 링크 연결 및 링크 전력 관리를 지원하도록 설계된다. 표의 하단을 향하여, 복수의 상태들에 적용되고 2개의 고정된 상태들: POWER_OFF 및 SS.Disabled.Default로의 천이를 가능하게 하는 2개의 전

역 조건들이 존재한다.

	상태	조건	다음 상태
1	POWER_OFF	$(US \&\& power_present) (DS \&\& !remote.RX_SIG_POWEROFF) \&\& wireless_link_up$	SS.Disabled.Default
2	SS.Disabled.Default	$!phy_status$	Rx.Detect.Reset
3	SS.Disabled.Error	$power_on_reset$ - (After "Sleep" event in the wireless apparatus, this is described later)	SS.Disabled.Default
4	Rx.Detect.Reset	$(US \&\& !fps)$	Rx.Detect.Remote
		$(DS \&\& !fps)$	Rx.Detect.Active
5	Rx.Detect.Active	rx_detect_fail	Rx.Detect.Quiet
		$(US \&\& rx_detect_pass)$	Polling.LFPS
		$(DS \&\& rx_detect_pass)$	Rx.Detect.Remote
		rx_detect_fatal	SS.Disabled.Error
6	Rx.Detect.Quiet	$timeout_12ms$	Rx.Detect.Active
7	Rx.Detect.Remote	$(US \&\& remote.RX_DETECT_SUCCESS)$	Rx.Detect.Active
		$(DS \&\& remote.RX_DETECT_SUCCESS)$	Polling.LFPS
8	Polling.LFPS	$US \&\& compliance_done \&\& timeout_360ms$	SS.Disabled.Default
		$DS \&\& compliance_done \&\& timeout_360ms$	Rx.Detect.Active
		$!compliance_done \&\& timeout_360ms$	Compliance Mode
		$remote.TSEQ_START$	Polling.RxEQ
		$warm_reset$	Rx.Detect.Reset
9	Polling.RxEQ	$remote.data == TS1_ordered_set$	PollingTraining
10	PollingTraining	$US \&\& remote.RX_SIG_IDLE$	SS.Disabled.Default
		$warm_reset$	Rx.Detect.Reset
		$idle_handshake \&\& TS2_loopback$	Loopback.Active
		$idle_handshake \&\& TS2_hotreset$	NonPollingTraining
		$idle_handshake \&\& !TS2_loopback \&\& ! TS2_hotreset$	U0
11	NonPollingTraining	$US \&\& remote.RX_SIG_IDLE$	SS.Disabled.Default
		$warm_reset$	Rx.Detect.Reset
		$idle_handshake \&\& TS2_loopback$	Loopback.Active
		$idle_handshake \&\& !TS2_loopback \&\& ! TS2_hotreset$	U0
12	U0	$lgou1_success$	U1
		$lgou2_success$	U2
		$lgou3_success$	U3
		$remote.data == TS1$	NonPollingTraining

		<i>US && remote.RX_SIG_IDLE</i>	SS.Disabled.Default
		<i>warm_reset</i>	Rx.Detect.Active
13	U1	<i>remote.data == TSEQ</i>	PollingTraining
		<i>warm_reset</i>	Rx.Detect.Active
		<i>timeout_n2</i>	U2
		<i>remote.data == TSI</i>	NonPollingTraining
14	U2	<i>Lfps</i>	U2_Exit
15	U2_Exit	<i>warm_reset</i>	Rx.Detect.Reset
		<i>remote.data == TSI</i>	NonPollingTraining
16	U3	<i>Lfps</i>	U3_Resume
		<i>timeout_100ms</i>	U3_RxDetect
17	U3_RxDetect	<i>RxDetectPass</i>	U3
		<i>RxDetectFail</i>	Rx.Detect.Reset
		<i>Lfps</i>	U3_Resume
18	U3_Exit	<i>warm_reset</i>	Rx.Detect.Reset
		<i>remote.data == TSI</i>	NonPollingTraining
19	Compliance Mode	<i>warm_reset</i>	Rx.Detect.Reset
20	Loopback.Active	<i>Lfps</i>	Loopback.Exit
		<i>warm_reset</i>	Rx.Detect.Reset
21	Loopback.Exit	<i>!Lfps</i>	Rx.Detect.Reset
		<i>timeout_2ms</i>	SS.Disabled.Default
	Any state	<i>exittoPowerOFF</i>	PowerOFF
	Any state but U1/U2/U3	<i>DS && remote.RX_SIG_IDLE</i>	SS.Disabled.Default
<p><i>DS = (id_dig == 0) .. id_dig is UTMI pin</i> <i>US = (id_dig == 1) .. id_dig is UTMI pin</i> <i>Lfps = !rx_electrical_idle && !rx_val .. Both are USB PHY pins</i> <i>rx_detect_pass = phy_status && (rx_status == 3'b011) // Both are USB PHY pins</i> <i>rx_detect_fail = phy_status && (rx_status != 3'b011)</i> <i>warm_reset = US && (DATA == warm_reset) DS && (remote.DATA == RX_LFPS_WARMRESET)</i> <i>compliance_done = (state == ComplianceMode) && warm_reset; //Level signal; once set remains set</i> <i>idle_handshake = ((DATA == IDLE)*8) && ((remote.DATA == IDLE)*16)</i> <i>TS2_loopback = (DATA == TS2_loopback) (remote.DATA == TS2_loopback)</i> <i>TS2_hotreset = DS && (remote.DATA == TS2_hotreset) US && (DATA == TS2_hotreset)</i> <i>lgo1_success = LGO_U1 -> LAU -> LPMA Timeout</i> <i>lgo2_success = LGO_U2 -> LAU -> LPMA Timeout</i> <i>lgo3_success = LGO_U3 -> LAU -> LPMA Timeout</i></p>			

[0098]

```

timeout_n2 = U2 Inactivity Timeout
tDisabledCount = (count_for_RxDetectFail == 10)
exittoPowerOFF = !WirelessLinkUp || tUS && !PowerPresent | DS &&
remote.RX_SIG_POWEROFF

```

[0099]

표 7: USB 3.0 초고속 상태 머신에 대한 상태 천이들

[0100]

표 7의 하단에서, 조건을 설명하기 위하여 사용되는 용어들은 USB 3.0 명세 및 PIPE 인터페이스 명세로부터 차용된 명칭들 및 심볼들을 사용한다. 또한, 표 5로부터의 시그널링 정보가 이러한 식들에서 사용된다. 예를 들어, 표 7의 로우(row) 7에서 사용된 *remote.RX_DETECT_SUCCESS*는 원격 무선 장치로부터 무선 링크를 통해 수신된 "Receive Detection Pass" 시그널링이다.

[0102]

무선 터널링 장치에 대하여 저 전력 상태를 구현하는 것은 다수의 사용 시나리오들, 예를 들어, USB 디바이스

플러그인을 기다리면서 저 전력 상태에 진입하는 것(슬립/W0 루프) 또는 W2 이탈 또는 W3 이탈을 기다리면서 저 전력 상태에 진입하는 것에서 전력 소비를 절감한다. 이러한 시나리오들 둘 모두에 있어서, 이러한 상태들 안으로 그리고 밖으로의 천이들이 일반적으로 사람의 타임스케일(timescale)로 트리거되기 때문에, 저 전력 상태는 장시간 지속될 수 있다는 것을 주의해야 한다. 따라서, 무선 터널링 장치가 이러한 시나리오들에 대하여 저 전력 상태들을 지원하도록 설계하는 것이 유익하다.

- [0103] 저-전력 상태의 장치들을 웨이크-업(wake-up)하기 위하여, 별개의 저-주파수(low-frequency; LF) 및 저-전력 무선 데이터-경로가 구현된다. LF TX 회로(424)는 무선 링크에 걸쳐 비동기적 신호들을 송신하기 위하여 사용된다. USB 디지털 로직은, 장치들이 U2 또는 U3 저 전력 상태에 있는 동안 USB PHY가 케이블 인터페이스 상에서 LFPS(저 주파수 주기적 시그널링)을 검출할 때면 언제든지 신호 SSUWakeup(또는 대안적으로 USB 2.0에 대하여 HSUWakeup)을 하이(high)로 비동기적으로 드라이브(drive)한다. 이러한 비동기적 신호는 LF TX 회로(424)를 사용하여 그리고 무선 컴포넌트 상태-머신(480)의 제어 하에서 송신된다.
- [0104] 유사하게, 무선 링크를 통해 웨이크업 신호가 검출될 때면 언제든지 신호 SSUWakeup이 LF RX 회로(446)에 의해 드라이브된다. 이는 장치들을 동작 상태 U0로 천이시키기 위하여 무선 컴포넌트 상태 머신(480) 및 USB 디지털(404)에 의해 사용된다.
- [0105] 표 8 및 표 9는 각기 일 실시예에 따른 예시적인 W2 입장/이탈 시퀀스 및 슬립 입장/이탈 시퀀스를 설명한다. 표들의 개시자 컬럼은 다음의 표기법들을 사용한다:
 - [0106] "호스트": USB 호스트 또는 업스트림 USB 허브(예를 들어, 프로세싱 장치(110A))
 - [0107] "디바이스": USB 주변기기 또는 다운스트림 USB 허브(예를 들어, 프로세싱 장치(110B))
 - [0108] "US-U": 업스트림 측 상의 USB 디지털(404)(예를 들어, 무선 터널링 장치(120A))
 - [0109] "US-W": 업스트림 측 상의 무선 블록들(예를 들어, 무선 터널링 장치(120A))
 - [0110] "DS-U": 다운스트림 측 상의 USB 디지털(404)(예를 들어, 무선 터널링 장치(120B))
 - [0111] "DS-W": 다운스트림 측 상의 무선 블록들(예를 들어, 무선 터널링 장치(120B))
 - [0112] W2/W3 상태들에서, 디지털 클럭들이 중단되고, 혼합된-신호/라디오 블록들이 저-전력 상태에 놓인다. 그러나, 슬립 모드에서, 전체 USB 디지털(404)(상태 머신들을 포함함) 및 무선 블록들의 대부분이 전력을 상실한다. 일 실시예에 있어서, 슬립 동안에도 전력을 유지하는 소형 항상-온 디지털 상태-머신이 존재한다.

	이벤트	개시자	업스트림 링크 상태	다운스트림 링크 상태	무선 전력 상태
--	-----	-----	------------------	-------------------	----------------

W2 입장 시퀀스

1	/gon2_success	호스트 또는 디바이스	U0->U2	U0->U2	W0
2	SSUPowerDown = W2	DS-U 및 US-U	U2	U2	W0
3	무선을 통한 W2 요청 교환	DS-W 및 US-W	U2	U2	W0
4	WLinkW0Mode 가 1로부터 0 으로 진행	DS-W 및 US-W	U2	U2	W2

W2 이탈 시퀀스 - 호스트 개시형

1	lfps (U2 이탈)	호스트	U2 -> U2 Exit	U2	W2
2	SSUWakeup = 1	US-U	U2_Exit	U2	
3	무선 웨이크업 요청	US-W	U2_Exit	U2	
4	SSWWakeup = 1 drvLfps = 1	DS-W DS-U	U2_Exit	U2->U2_Exit	W0
5	lfps (핸드세이크)	디바이스	U2_Exit	U2_Exit	
6	RY_SIG_LFPS	DS-U	U2_Exit	U2_Exit	
7	RY_SIG_LFPS Rx @US	US-W	U2_Exit	U2_Exit	
8	drvLfps = 1	US-U	U2_Exit	U2_Exit	
9	LFPS end sequence				
10	TS1 Training	호스트 및 디바이스	U2_Exit -> NonPolling Training	U2_Exit -> NonPolling Training	W0

W2 이탈 시퀀스 - 디바이스 개시형

1	lfps (U2 이탈)	디바이스	U2	U2->U2_Exit	W2
2	SSUWakeup = 1	DS-U	U2	U2_Exit	

3	무선 웨이크업 요청	DS-W	U2	U2_Exit	
4	SSW4syncWakeup = 1 drvLfps = 1	US-W US-U	U2->U2_Exit	U2_Exit	W0
5	lfps (핸드세이크)	호스트	U2_Exit	U2_Exit	
6	RY_SIG_LFPS	US-U	U2_Exit	U2_Exit	
7	RY_SIG_LFPS Rx @US	DS-W	U2_Exit	U2_Exit	
8	drvLfps = 1	DS-U	U2_Exit	U2_Exit	
9	LFPS 가 시퀀스를 종료				
10	TS1 Training	호스트 및 디바이스	U2_Exit -> NonPolling Training	U2_Exit -> NonPolling Training	W0

[0113]

[0114]

[0115] 표 8: USB3.0 초고속에 대한 W2 입장/이탈 시퀀스

	이벤트	개시자	업스트림 링크 상태	다운스트림 링크 상태	무선 전력 상태
<u>슬립 입장 시퀀스</u>					
1	USB Rx 가분리 시도를 검출한다(SSDisconnect=1)		Rx.Detect 상태 중 임의의 상태	Rx.Detect 상태 중 임의의 상태	W0
2	SS.Disconnect=1 인 상태에서 150ms 타임아웃	DS-W 및 US- W	링크 이니트(Init) 상태 중 임의의 상태	Rx.Detect 상태 중 임의의 상태	W0
3	무선을 통한 슬립 요청 교환	DS-W 및 US- W	Rx.Detect 상태 중 임의의 상태	Rx.Detect 상태 중 임의의 상태	W0
4	슬립 상태에 입장 WirelessLinkUp = 0		Rx.Detect 상태 중 임의의 상태 -> PowerOFF	Rx.Detect 상태 중 임의의 상태 ->PowerOFF	W0 ->슬립
<u>슬립 이탈 시퀀스 - 호스트 개시형</u>					
1	1 초 슬립 타임아웃	DS-W 및 US- W	전력 없음	전력 없음	슬립
2	파워 업	DS-W 및 US- W			슬립 ->W0
3	정상 파워 업 시퀀스	모든 블록들			W0

[0116] 표 9: USB3.0 초고속에 대한 슬립 입장/이탈 시퀀스

[0117] USB 2.0 상태 머신

[0118] HS/FS/LS 테이터에 대한 USB2.0 상태 머신(631)은 USB 2.0 명세에 설명된 리셋 프로토콜 상태 머신(Reset Protocol State Machine; RPSM)으로부터 시작하여 적응되고 최적화된다. 이는 D+/D- 라인들을 사용하여 고속, 전속, 및 저속 동작들을 처리한다.

[0119] 표 10은 USB 2.0 상태-머신에 의해 생성되는 시그널링 정보를 도시한다. 무선 통신을 위한 바이트들의 인코딩은 (표 6에서 설명된) 초고속과 동일한 기법을 사용한다.

	시그널링	인코딩된 값 (8 비트 - 바이너리)	코멘트
1	LINE_STATE_SE0	0000_0000	원격 슬링샷(Slingshot) UTMI에서 SE0으로의 라인 상태의 변화
2	LINE_STATE_0	0000_0010	원격 슬링샷 UTMI에서 차이 "0"으로의 라인 상태의 변화
3	LINE_STATE_1	0000_0001	원격 슬링샷 UTMI에서 차이 "1"로의 라인 상태의 변화
4	LINE_STATE_SE1	0000_0011	원격 슬링샷 UTMI에서 SE1로의 라인 상태의 변화
5	HostDisconnect	0001_0001	원격 슬링샷 UTMI에서 HostDisconnect 0->1 변화. 오로지 DS에 대해서만 유효
6	HostConnect	0001_0000	원격 슬링샷 UTMI에서 HostDisconnect 1->0 변화. 오로지 DS에 대해서만 유효
7	VBUS_ON	0010_0001	원격 슬링샷에서 PowerPresent 0->1 천이
8	VBUS_OFF	0010_0000	원격 슬링샷에서 PowerPresent 1->0 천이

[0121]

표 10: USB 2.0 시그널링 정보

[0122]

표 11은 17-상태 USB 2.0 상태 머신(631)에 대한 천이 조건 및 다음 상태를 요약한다. 초고속 동작이 더 정교화된 전력 관리를 구현하기 때문에, 이러한 상태-머신은 초고속에 대한 상태-머신보다 더 단순하다. 다른 측 상의 USB 2.0 상태 머신(631)은 HS, FS, 및 LS를 지원하기 위하여 상이한 상태들을 필요로 한다 - 이는 이하의 표에서 3번 복제된 몇몇 상태들로부터 명백하다. 표의 하단을 향하여 복수의 현재 상태들에 적용되는 3개의 전역 조건들이 존재한다. 이들은 3개의 고정된 상태들: Poweroff, Disconnected, 및 Reset_SE0으로의 천이를 가능하게 한다.

	현재 상태	조건	다음 상태
1	Poweroff	<i>sigPowerOn</i>	Disconnected
2	Disconnected	<i>sigHostConnect</i>	Connected
3	Connected	<i>sigline_host_SE0</i>	Reset_SE0
4	Reset_SE0	<i>sigline_dev_I</i> <i>sigline_dev_0</i>	Reset_FS LS_Default
5	Reset_FS	<i>sigline_dev_K</i> <i>timeout_10ms</i>	Reset_chirpK FS_Default
6	Reset_chirpK	<i>!sigline_dev_K</i> <i>timeout_2.5us</i>	Reset_JK_handshake FS_Default
7	Reset_JK_handshake	<i>sigline_host_SE0</i>	Reset_SE02
8	Reset_SE02	<i>!sigline_host_SE0</i>	HS_Default
9	HS_Default	<i>idle_more_than_3ms</i>	HS_Suspend
10	FS_Default	<i>idle_more_than_3ms</i>	FS_Suspend
11	LS_Default	<i>idle_more_than_3ms</i>	LS_Suspend
12	HS_Suspend	<i>(Remote.LineState==K) (LineState==K)</i>	HS_Suspend_Exit
13	HS_Suspend_Exit	<i>(Remote.LineState==J) (LineState==J)</i>	HS_Default
14	FS_Suspend	<i>(Remote.LineState==K) (LineState==K)</i>	FS_Suspend_Exit
15	FS_Suspend_Exit	<i>(Remote.LineState==J) (LineState==J)</i>	FS_Default
16	LS_Suspend	<i>(Remote.LineState==K) (LineState==K)</i>	LS_Suspend_Exit
17	LS_Suspend_Exit	<i>(Remote.LineState==J) (LineState==J)</i>	LS_Default
	임의의 상태	<i>sigPowerOff</i>	Poweroff
	임의의 상태	<i>sigHostDisconnect</i>	Disconnected
	Reset 상태들 0 외의 임의의 상태	<i>sigline_host_SE0</i>	Reset_SE0
<p>DS = <i>(IdDig == 0) .. IdDig is UTMI pin</i> US = <i>(IdDig == 1) .. IdDig is UTMI pin</i> <i>sigPowerOn = WirelessLinkUp && (US && PowerPresent DS && remote.VBUS_ON)</i> <i>sigPowerOff = !WirelessLinkUp (US && !PowerPresent DS && remote.VBUS_OFF)</i> <i>sigHostConnect = US && remote.HostConnect DS && !HostDisconnect</i> <i>sigHostDisconnect = US && remote.HostDisconnect DS && HostDisconnect</i> <i>sigline_host_SE0 = DS && remote.LINE_STATE_SE0 US && LINESTATE_SE0</i> <i>sigline_host_I = DS && remote.LINE_STATE_I US && LINESTATE_I</i> <i>sigline_host_0 = DS && remote.LINE_STATE_0 US && LINE_STATE_0</i> <i>sigline_host_J = (FS HS) && sigline_host_I LS && sigline_host_0</i> <i>sigline_host_K = (FS HS) && sigline_host_0 LS && sigline_host_I</i></p>			

[0124]

<i>sigline_dev_I = DS && LINE_STATE_I US && remote.LINE_STATE_I</i>
<i>sigline_dev_0 = DS && LINE_STATE_0 US && remote.LINE_STATE_0</i>
<i>sigline_dev_J = (FS HS) && sigline_dev_I LS && sigline_dev_0</i>
<i>sigline_dev_K = (FS HS) && sigline_dev_0 LS && sigline_dev_I</i>

[0125]

표 11: USB 2.0 상태-머신에 대한 상태들 & 천이들

[0126]

표 11의 하단에서, 조건에서 사용되는 용어들은 USB 2.0 명세 및 UTMI 인터페이스 명세로부터 차용된 명칭들 및 심볼들을 사용한다. 또한, 표 10으로부터의 USB 2.0 시그널링 정보가 이러한 식들에서 사용된다.

[0127]

표 12 및 13은 고속, 전속, 및 저속 동작들에 대한 W3 입장/이탈 및 슬립 입장/이탈 시퀀스들을 설명한다. 이러한 표들에 있어서, "xx_" 접두사는 HS, FS, 및 LS 중 하나를 나타낸다.

[0128]

이상에서 설명된 바와 같이, 무선 설계는 전력 상태들: W0, W2, W3, 및 슬립을 지원한다. 상태들 W0, W2, 및 W3은 USB3.0 초고속 상태들 U0, U2, 및 U3에 매핑된다. 일 실시예에 있어서, U1에 대한 이탈 레이턴시 요구이 매우 짧기 때문에 장치는 U1 상태에서 무선 블록들의 전원을 차단하지 않는다. USB 2.0의 "중지" 전력 상태는 W3

에 매핑된다. 이는, "중지-재개" 이탈 레이턴시 요건이 초고속 U3의 그것에 비할만하기 때문이다.

[0130]

일 실시예에 있어서, 무선 터널링 장치는 USB 3.0 허브의 업스트림 측 상에서 사용될 수 있으며, 그럼으로써 USB 3.0 초고속 및 USB 2.0 데이터 전송들 둘 모두를 동시에 사용할 수 있다. 따라서, 이러한 실시예에 있어서, 전력 상태 W3은, USB 2.0 상태 머신(631) 및 USB 3.0 초고속 상태 머신(621) 둘 모두를 체크한 이후에만 진입된다. 예를 들어, W3은, USB 3.0 초고속 링크가 U3 또는 디세이블된 상태이고 USB 2.0 링크가 중지 상태인 경우에 한하여 진입된다. 유사하게, 슬립 상태는, USB 3.0 초고속 링크가 디세이블된 상태이고 USB 2.0 링크가 분리된 상태인 경우에 한하여 진입된다.

	이벤트	개시자	업스트림 링크 상태	다운스트림 링크 상태	무선 전력 상태
<u>W3 입장 시퀀스</u>					
1	IDLE > 3ms	호스트	xx_Default -> xx_Suspend	xx_Default -> xx_Suspend	W0
2	USB 상태-머신이 중지 신호를 무선 설계로 어서트(assert)한다	DS-U 및 US-U	xx_Suspend	xx_Suspend	W0
3	무선을 통한 W3 요청 교환	DS-W 및 US-W	xx_Suspend	xx_Suspend	W0
4	WLinkW0Mode = 1 - > 0	DS-W 및 US-W	xx_Suspend	xx_Suspend	W0->W3
<u>W3 이탈 시퀀스 - 호스트 개시형</u>					
1	20ms 동안 Linestate == K	호스트	xx_Suspend -> xx_Suspend_Exit	xx_Suspend	W3
2	HSUWakeup = 1	US-U	xx_Suspend_Exit	xx_Suspend	
3	무선 웨이크업 요청	US-W	xx_Suspend_Exit	xx_Suspend	
4	HSUWakeup = 1 drvResume = 1	DS-W DS-U	xx_Suspend_Exit	xx_Suspend -> xx_Suspend_Exit	W3->W0
5	Idle (후 재개)	호스트	xx_Default	xx_Suspend_Exit	W0
6	sigline_host_J	US-U	xx_Default	xx_Default	W0
<u>W3 이탈 시퀀스 - 디바이스 개시형</u>					
1	1-15ms 동안 Linestate == K	디바이스	xx_Suspend	xx_Suspend -> xx_Suspend_Exit	W3
2	HSUWakeup = 1	DS-U	xx_Suspend	xx_Suspend_Exit	

[0131]

3	무선 웨이크업 요청	DS-W	xx_Suspend	xx_Suspend_Exit	
4	WAsyncWakeups = 1 DrvResume = 1	US-W 및 US-U	xx_Suspend -> xx_Suspend_Exit	xx_Suspend_Exit	W3->W0
5	Idle (후 재개)	디바이스	xx_Suspend_Exit	xx_Default	W0
6	Sigline_host_J	DS-U	xx_Default	xx_Default	W0

[0132]

표 12: USB 2.0 중지-재개에 대한 W3 입장/이탈 시퀀스

	이벤트	개시자	업스트림 링크 상태	다운스트림 링크 상태	무선 전력 상태
슬립 입장 시퀀스					
1	호스트 또는 디바이스가 분리되거나 또는 전력 차단된다	호스트 또는 디바이스			W0
2	USB 상태-머신이 분리 신호를 무선 설계로 어서트한다	DS-U 또는 US-U	PowerOff/ Disconnect	PowerOff/ Disconnect	W0
3	disconnect = 1 인 상태에서 150ms 타임아웃	DS-W 및 US-W	PowerOff/ Disconnect	PowerOff/ Disconnect	W0
4	무선을 통한 슬립 요청 교환	DS-W 및 US-W	PowerOff/ Disconnect	PowerOff/ Disconnect	W0
5	슬립에 입장: WirelessLinkUp = 0	전력 상실			슬립
슬립 이탈 시퀀스 - 호스트 개시형(연결을 위한 USB 2.0 풀)					
1	1 초 슬립 타임아웃	DS-W 및 US-W	전력 없음	전력 없음	슬립
2	파워 업	DS-W 및 US-W			슬립 -> W0
3	정상 파워 업 시퀀스	모든 블록들			W0

[0134]

표 13: USB 2.0 분리-연결에 대한 슬립 입장/이탈 시퀀스

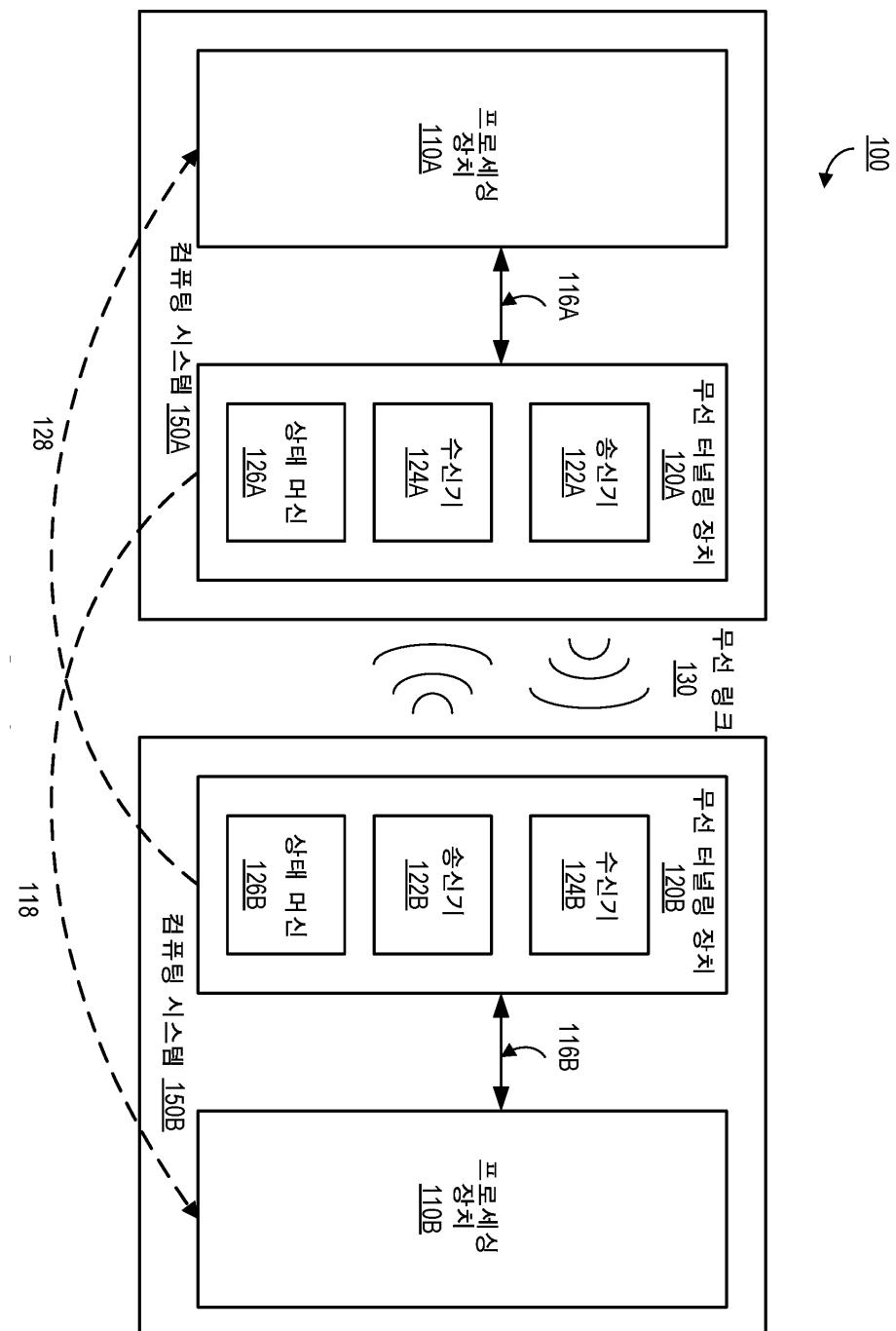
[0135]

본 개시를 숙독할 때, 당업자들은 본원에 개시된 원리들을 통해 추가적이고 대안적인 실시예들을 인식할 것이다. 따라서, 특정 실시예들 및 애플리케이션들이 예시되고 설명되었지만, 개시된 실시예들이 본원에 개시된

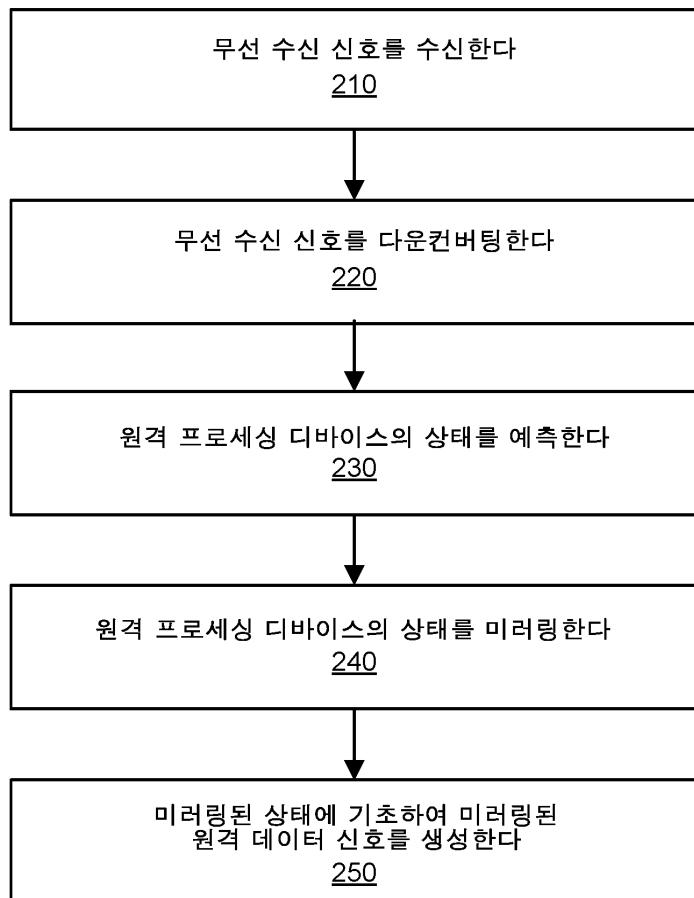
정확한 구성 및 컴포넌트들에 한정되지 않는다는 것이 이해될 것이다. 당업자들에게 자명할 다양한 수정들, 변화들 및 변형들이 본원에서 설명된 범위로부터 벗어나지 않고 본원에 개시된 방법 및 장치의 배열, 동작 및 세부사항들에 대하여 이루어질 수 있다.

도면

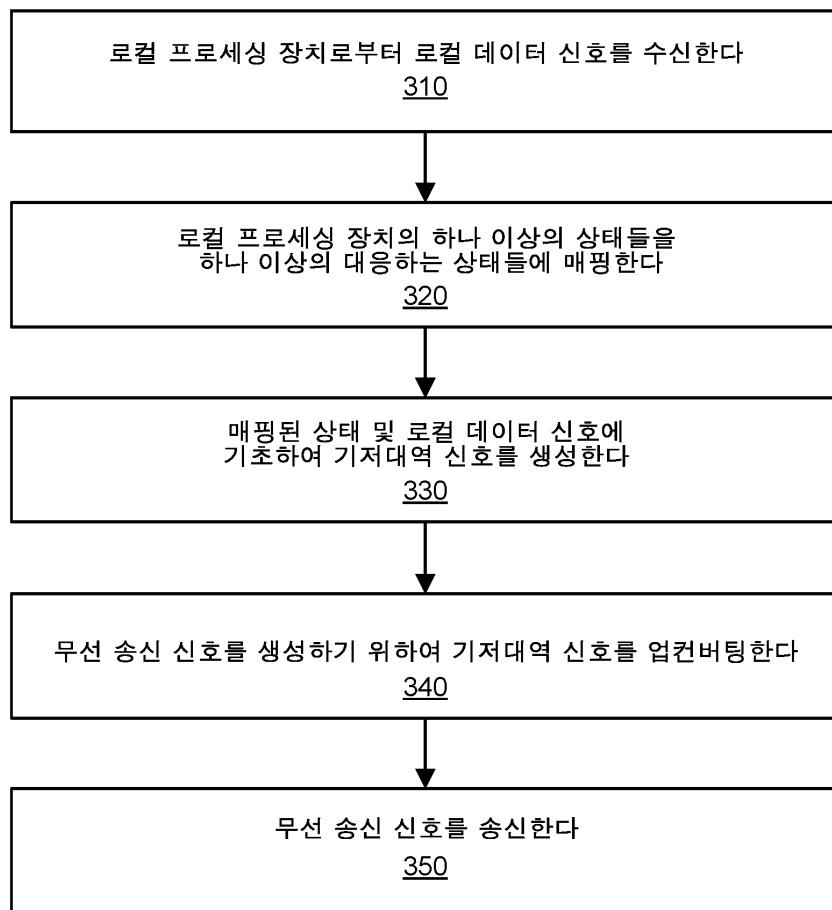
도면1



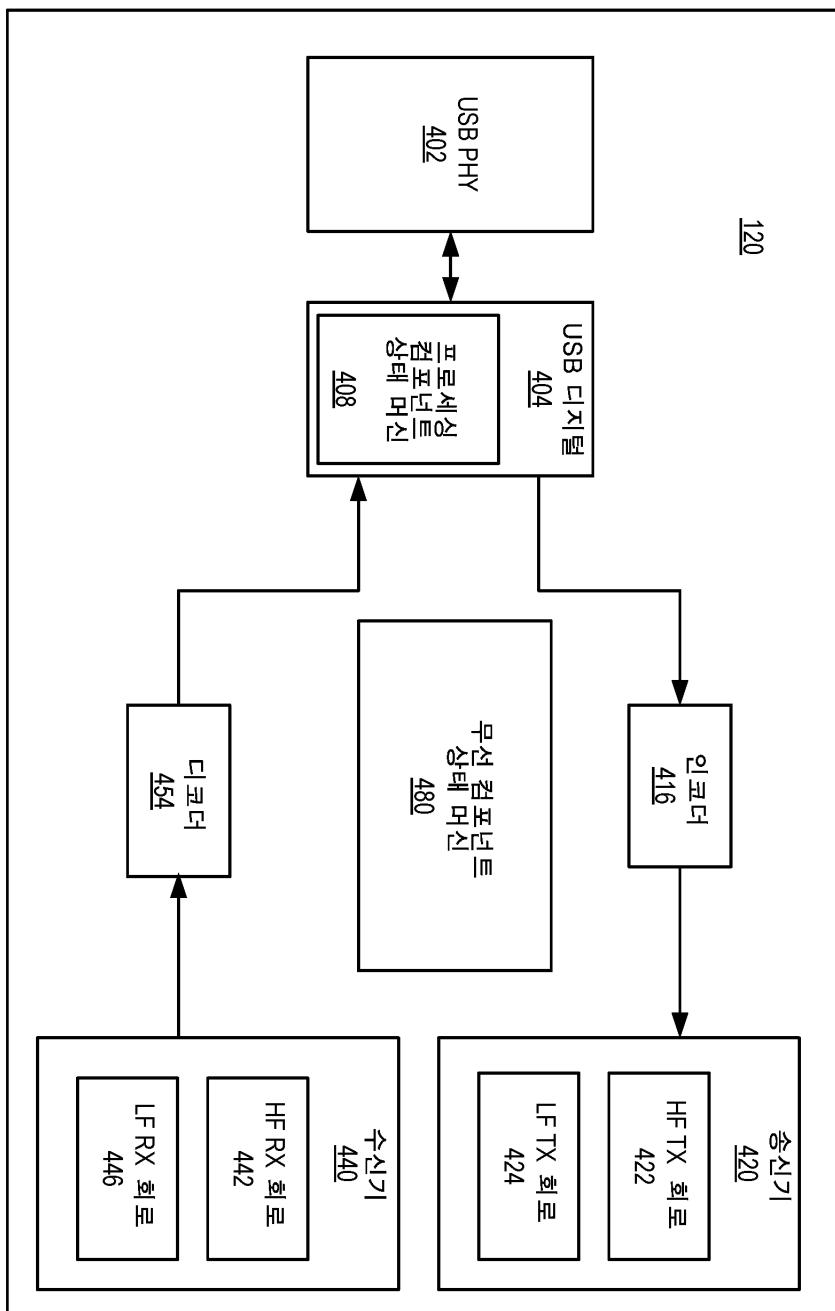
도면2



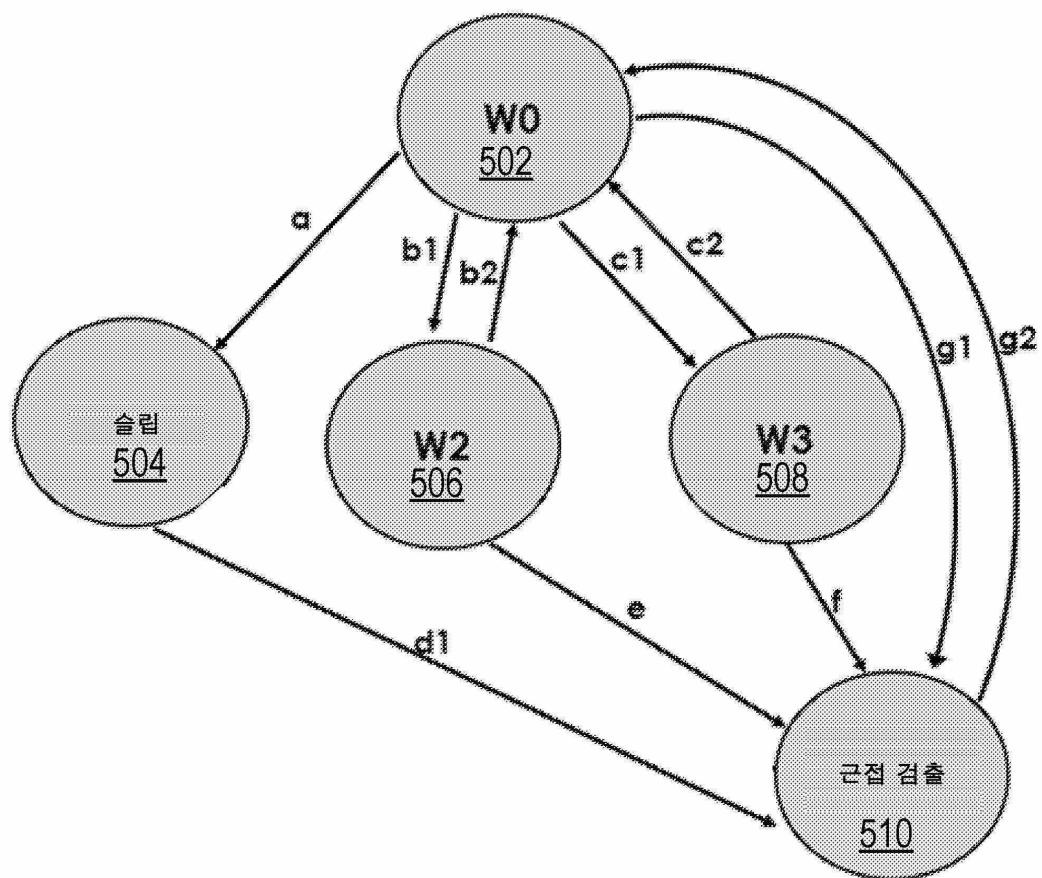
도면3



도면4



도면5



도면6

