

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) . Int. Cl.⁷
 G06F 15/16

(11) 공개번호 10-2005-0028869
 (43) 공개일자 2005년03월23일

(21) 출원번호 10-2004-0074690
 (22) 출원일자 2004년09월17일

(30) 우선권주장	60/504,060	2003년09월18일	미국(US)
	60/552,352	2004년03월10일	미국(US)
	10/909,085	2004년07월29일	미국(US)

(71) 출원인 제네시스 마이크로칩 인코포레이티드
 미국 캘리포니아 95002, 알비소, 2150 골드 스트리트

(72) 발명자 고바야시, 오사무
 미국, 캘리포니아 94024-5809, 로스 알토스, 풀른 리프 레인 1464

(74) 대리인 장명구

심사청구 : 없음

(54) 패킷 기반 스트림 이송 스케줄러 및 그 이용 방법

요약

멀티미디어 소스 장치를 멀티미디어 싱크 장치에 연결하는 방법이 공개된다. 이 방법은,

- 송신기 유닛이 연결된 소스 장치를 제공하고,
- 수신기 유닛이 연결된 싱크 장치를 제공하며,
- 원본 스트림 속도에 따라 소스 데이터 스트림을 송신기 유닛에 의해 수신하고,
- 링크 유닛을 이용하여 송신기 유닛과 수신기 유닛을 연결하며,
- 다수의 멀티미디어 데이터 패킷들로 구성되는 멀티미디어 데이터 패킷 스트림을 형성하고, 그리고
- 송신기 유닛과 수신기 유닛 간에 링크 속도에 따라 멀티미디어 데이터 패킷 스트림을 이송하는 이송 스케줄을 발행시키며, 이때, 멀티미디어 데이터 패킷들이 링크 속도와 데이터 스트림 비트 속도에 따라 각기 고정 크기를 가지는

단계들을 포함하는 것을 특징으로 한다.

대표도

도 1

명세서

도면의 간단한 설명

도 1은 발명의 한 실시예에 따른 크로스 플랫폼 디스플레이 인터페이스(100)의 일반화된 도면.

도 2A-C는 발명의 다수의 실시예들에 따라 비디오 소스와 비디오 디스플레이 유닛을 연결시키는 데 사용되는 비디오 인터페이스 시스템의 도면.

도 3은 발명의 한 실시예에 따른 일례의 메인 링크 속도 도면.

도 4A는 발명의 한 실시예에 따른 메인 링크 데이터 패킷의 도면.

도 4B는 발명의 한 실시예에 따른 메인 링크 패킷 헤더의 도면.

도 5A는 발명의 한 실시예에 따라 서브-패킷 인클로저와 멀티-패킷 멀티플렉싱을 제공하도록 배열되는 시스템의 도면.

도 5B는 도 5A에 도시된 시스템의 또다른 구현 예 도면.

도 6은 도 5에 도시되는 스트림의 한 예로서 멀티플렉싱된 메인 링크 스트림의 하이-레벨 도면.

도 7은 발명의 한 실시예에 따른 데이터 스트림의 또다른 예 도면.

도 8은 발명의 한 실시예에 따른 멀티플렉싱된 데이터 스트림의 또다른 예 도면.

도 9A는 발명의 한 실시예에 따른 서브-패킷의 도면.

도 9B는 발명의 한 실시예에 따른 메인 링크 데이터 패킷의 도면.

도 10은 선택적으로 리프레시되는 그래픽스 이미지의 일례의 도면.

도 11은 발명의 한 실시예에 따른 일례의 링크 트레이닝 패턴의 도면.

도 12는 발명의 한 실시예에 따른 데이터 스트림 스케줄러를 가진 대표 시스템의 도면.

도 13은 도 12에 도시되는 데이터 스트림의 국부 상세도.

도 14는 발명의 한 실시예에 따른 제 4 데이터 스트림의 추가도면.

도 15는 발명의 한 실시예에 따른 단일 비압축 데이터 스트림의 디제너레이터 케이스 도면.

도 16은 도 15에 도시되는 디제너레이트 단일 데이터 스트림에 바탕한 비디오 디스플레이 도면.

도 17은 발명의 한 실시예에 따라 다수의 데이터 스트림을 스케줄링하는 과정의 순서도.

도 18은 발명의 한 실시예에 따라 링크 데이터 스트림을 형성하는 과정의 순서도.

도 19는 발명의 한 실시예에 따라 시스템의 로직 총배치를 도시하는 도면.

도 20은 발명의 한 실시예에 따라 8B/10B를 이용하는 일례의 특수 문자 매핑의 도면.

도 21은 발명의 한 실시예에 따른 일례의 맨체스터 II 인코딩 기법의 도면.

도 22는 발명의 한 실시예에 따른 보조 채널 전기 서브층 도면.

도 23은 발명의 한 실시예에 따른 메인 링크 전기적 서브층의 도면.

도 24는 발명의 한 실시예에 따른 커넥터의 도면.

도 25는 발명의 한 실시예에 따른 소스 상태 도면.

도 26은 발명의 한 실시예에 따른 디스플레이 상태 도면.

도 27-31은 발명의 다양한 컴퓨터 기반 구현들의 도면.

도 32는 발명의 한 실시예에 따른 인터페이스의 동작 모드를 결정하는 과정의 순서도.

도 33은 발명의 여러 태양들에 따라 실시간 비디오 이미지 품질 확인을 제공하는 과정의 순서도.

도 34A와 34B는 발명의 한 실시예에 따른 링크 셋업 과정의 순서도.

도 35는 발명의 한 실시예에 따른 트레이닝 세션을 실행하는 과정의 순서도.

도 36은 발명의 구현에 사용되는 컴퓨터 시스템의 도면.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 멀티미디어 장치에 관한 것이다. 특히 본 발명은 데이터 패킷 스트림 스케줄러 및 그 이용 방법에 관한 것이다.

라스터 스캔 비디오 트랜스포트 프로토콜은 CRT 기반 디스플레이 시스템에서 사용하기 위해 개발되었다. 이 시스템은 디스플레이되는 이미지를 한번에 한 라인씩 물리적으로 "그리기 위해" 전자총이 사용된다는 사실을 고려하여야 한다. 예를 들어, 표준 화질(VGA) 비디오 이미지는 480개의 액티브 디스플레이 라인들을 포함하는 액티브 영역으로 형성된다. 각각의 라인은 640개의 회소들로 형성된다. 즉, 640 x 480의 해상도를 지닌다. 그러나 액티브 영역에 추가하여, 디스플레이되지 않으나 비디오 신호에 포함되는 블랭킹 영역이 존재한다. 왜냐하면, 이는 수평 및 수직 리트레이스용으로 요구되는 시간을 나타내기 때문이다. 예를 들어, VGA 이미지의 각각의 프레임(즉, 한 개의 풀프레임은 640개 화소들로 각각 구성되는 480개의 라인들에 해당함)은 수평 리트레이스용으로 라인당 160 화소를 럭과, 수직 리트레이스용으로 대략 45 라인 주기와 같은 시간 주기를 필요로 한다. 이 방식으로(화소 클럭 당 한 개의 화소를 가정), VGA 이미지의 디스플레이에 필요한 비디오 데이터를 이동시키는 데 요구되는 비디오 신호는 800 화소 클럭 수준(640 액티브 화소 클럭 + 160 블랭킹 화소 클럭)이어야 한다. 따라서, 트랜스포트 효율은 80% 수준(즉, 640/800)이다. 트랜스포트 효율은 총 데이터 스트림 대역폭에 대한 디스플레이가능한 데이터의 대역폭으로 규정된다.

최근에 HDTV 및 그 외 다른 고화질 그래픽을 수용하기 위해 CRT의 해상도가 증가함에 따라, 라스터 스캔 비디오 트랜스포트 프로토콜들이 대략 90%로 증가하였다. 이는 수평 리트레이스를 160 화소 클럭으로 제한하는 것(따라서 관련 블랭킹 주기를 감소시키는 것)을 요한다. 예를 들어, UVGA 이미지(즉, 1600 x 1200)의 경우에, 수평 리트레이스가 160 화소 클럭으로 유지될 경우 트랜스포트 효율은 대략 90%이다(1600/1760). 라스터 스캔 비디오 트랜스포트 프로토콜이 효율적이고(90% 수준) 대형 버퍼를 요하지 않는다. 하지만, 이 프로토콜들은 유통성이 부족하여, 단지 데이터를 디스플레이할 수 있음에 지나지 않는다.

라스터 스캔 비디오 트랜스포트 프로토콜들에 추가하여, 디지털 비디오 기반 시스템의 출현하여 디지털 비디오 트랜스포트 프로토콜들에 대한 필요성을 만들어냈다. IEEE 1394 또는 FireWall이라 불리는 이러한 디지털 비디오 트랜스포트 프로토콜 중 하나는 균일한 비트 속도를 보장하고 다중 데이터 스트림들(가령, 비디오 스트림과, 오디오 스트림 형태의 관련 사운드트랙)간 동기성을 유지하기 위해 대형 버퍼에 의존하는 동기 패킷 트랜스포트(isochronous packet transport)를 바탕으로 한다. 동기 패킷 트랜스포트 프로토콜들이 기본적으로 탄력성을 띠지만(패킷 기반 속성으로 인함), 대형 버퍼 요건들은 매우 맷가가 크다.

따라서, 라스터 스캔 트랜스퍼 프로토콜의 효율성(트랜스포트 효율 및 메모리 리소스 이용 효율)을 가지면서도 동기 패킷 트랜스퍼 프로토콜의 탄력성을 지닌 데이터 스트림 트랜스포트 프로토콜이 요망된다.

발명이 이루고자 하는 기술적 과제

멀티미디어 소스 장치를 멀티미디어 싱크 장치에 연결하는 방법이 제공된다. 이 방법은, 1) 송신기 유닛을 연결한 소스 장치를 제공하고, 2) 수신기 유닛을 연결한 싱크 장치를 제공하며, 3) 송신기 유닛에 의한 원본 스트림 속도에 따라 소스 데이터 스트림을 수신하고, 4) 링크 유닛을 통해 송신기 유닛과 수신기 유닛을 연결하며, 5) 다수의 멀티미디어 데이터 패킷들로 구성되는 멀티미디어 데이터 패킷을 형성하고, 그리고 6) 송신기 유닛과 수신기 유닛 간에 링크 속도에 따라 멀티미디어 데이터 패킷 스트림을 이송하는 트랜스포트 스케줄을 발생시키는 단계들로 구성되며, 이때, 멀티미디어 데이터 패킷들은 링크 속도와 데이터 스트림 비트 속도를 바탕으로 각각 고정 크기를 가진다.

데이터 링크를 통해 데이터 소스와 데이터 싱크 간의 다수의 데이터 패킷들의 이송을 스케줄링하는 방법이 제공된다. 이 방법은, 1) 데이터 패킷 소스로부터 데이터 패킷 싱크까지 데이터 패킷 속성들을 전송하고, 2) 소스로부터 싱크까지 전달될 다수의 데이터 스트림 각각에 대한 데이터 링크 비트 속도에 스트림 비트 속도를 비교하며, 3) 상기 비교를 바탕으로 데이터 스트림 각각에 대한 패킷 크기를 설정하고, 이때, 상기 패킷 크기는 고정 패킷 크기이며, 4) 각각의 데이터 패킷 중 한 개 이상을 조합하며, 그리고 5) 상기 소스로부터 싱크까지 조합된 데이터 패킷들을 이송하는 단계들을 포함한다.

또하나의 실시예에서, 데이터 링크를 통해 데이터 소스와 데이터 싱크 간에 다수의 데이터 패킷들의 이송을 스케줄링하는 컴퓨터 프로그램 프로토콜이 제공된다. 이 컴퓨터 프로그램 프로토콜은, 1) 데이터 패킷 소스로부터 데이터 패킷 싱크까지 데이터 패킷 속성을 전송하는 컴퓨터 코드, 2) 소스로부터 싱크까지 전달될 다수의 데이터 스트림들 각각에 대한 데이터 링크 비트 속도에 스트림 비트 속도를 비교하는 컴퓨터 코드, 3) 상기 비교 과정을 바탕으로 데이터 스트림 각각에 대해 패킷 크기를 설정하는 컴퓨터 코드(이때, 패킷 크기는 고정 패킷 크기), 4) 각각의 데이터 패킷 중 한 개 이상을 조합하는 컴퓨터 코드, 5) 조합된 데이터 패킷들을 소스로부터 싱크까지 이송하는 컴퓨터 코드, 그리고 6) 이 코드들을 저장하는 컴퓨터 판독형 매체를 포함한다.

발명의 구성 및 작용

발명은 패킷 기반 디지털 인터페이스를 통해 비디오, 성크나 수신기에 연결되는 비디오 소스를 구비한 비디오 디스플레이 시스템을 이용하여 설명될 것이다. 소스 장치에 연결된 송신기 유닛은 관련 스트림 속성들을 각각 가지는 패킷화된 비디오 데이터 스트림들을 수신한다. 설명되는 비디오 시스템의 측면에서, 이러한 속성들은 비디오 포맷, 칼라 깊이, 등등을 포함한다. 수신기 유닛은 데이터를 통해 소스에 연결되고, 또는 메인 링크에 연결되고, 메인 링크를 통한 데이터 패킷들의 전송 이전에 소스로부터 수신기에 스트림 속성 데이터를 이송하는 데 사용되는 관련 보조 링크에 연결된다. 이 방식으로, 패킷 헤더들은 데이터 패킷의 어떤 데이터 스트림이 관련되었는지를 식별하고 어떤 데이터 스트림이 스트림 ID만을 포함할 수 있는지, 또는 그 외 다른 이러한 식별자를 포함할 수 있는지를 식별하는데 사용된다. 이 방식으로 패킷 오버헤드가 실질적으로 감소되어, 효율적인 패킷 이송 메커니즘을 제공하도록 비디오 및 오디오 데이터 같은 멀티미디어 콘텐트에 대해 메인 링크 대역폭을 보존한다. 메인 링크에서의 데이터 전송을 조율하기 위해, 트랜스포트 스트림 스케줄러는 데이터 링크에 대해 다수의 패킷화된 데이터 스트림들 중 전부 또는 일부를 스케줄링하는 효율적이고 탄력적인 시스템, 방법, 장치를 제공한다. 이때, 소스로부터 성크까지 스트림 속성 데이터를 전송하는 보조 채널이 또한 제공될 수 있다.

각각의 데이터 스트림은 다수의 관련 데이터 패킷들로 형성되며, 그 크기는 특정 데이터 스트림에 대해 요구되는 링크 대역폭의 상대적 부분에 의존한다. 특정 데이터 스트림 i의 경우에, 관련 패킷 크기 PSi는 최대 패킷 크기 MPS, 링크 비트 속도 LBR, 그리고 스트림 비트 속도 SBRi의 함수이다.

$$PSi = MPS * SBRi / (LBRi + 1)$$

이 방식으로, 데이터 링크 대역폭에 대한 데이터 스트림의 상대적 대역폭을 바탕으로 각각의 데이터 스트림에 대해 패킷 크기가 결정된다. 예를 들어, 최대 패킷 크기가 64개의 링크 심별이고 링크 비트 속도 LBR이 2.5 Gbps/레인 일 경우에, 표 1은 선택된 스트림 비트 속도에 대응하는 패킷 크기들을 제시한다.

표 1

링크 비트 속도(LBR)	스트림 비트 속도(SBR)	패킷 크기(PS)
2.5 Gbps/lane	1.0 Gbps/lane	32 링크 심별
2.5 Gbps/lane	0.3125 Gbps/lane	10 링크 심별
2.5 Gbps/lane	0.25 Gbps/lane	8 링크 심별
InterLaneAlignment(ILA)		2 링크 심별

스케줄러 시간 분할은 다중 스트림들의 패킷을 대응하는 링크 데이터 스트림으로 멀티플렉싱(송신기측)하고 디멀티플렉싱(수신기측)한다. 본 실시예에서, 송신기는 최대 데이터 링크 속도, 사용 범위(速率) TBR(Transmission Bit Rate) 유닛의 수 등을 이용하여 수신기의 능력을 판독한다. 이 지식으로, 송신기는 가장 효율적인 트랜스포트 구성(구성)을 결정할 수 있고, 수신기에 추가적인 질의를 전송할 필요없이 수신기에 의해 차후 데이터 스트림이 수용될 수 있는지 여부를 결정할 수 있다. 데이터 스트림 트랜스포트를 시작하기 전에, 송신기는 비디오 데이터의 경우에 각각의 데이터 스트림에 관련된 칼라 포맷 및 깊이, 형태, 그리고 패킷 크기 같은 스트림 속성을 수신기에 알린다. 속성 데이터를 통신함으로서, 패킷 헤더의 크기는 단일 스트림 ID만이 요구되는 포인트까지 감소될 수 있다. 이 방식으로, 패킷 헤더들이 크기 때문에 보다 큰 오버헤드를 요구하는 기존 패킷 기반 트랜스포트 프로토콜에서 제공되는 트랜스포트 효율에 비해 이 경우의 트랜스포트 효율이 훨씬 증가한다.

발명에 따른 스트림 스케줄러의 논의를 위한 배경을 제공하기 위해, 발명의 구현에 적합한 표본 디지털 비디오 시스템이 기술된다.

따라서, 도 1은 발명의 다수의 실시예들을 구현하기에 적합한 패킷 기반 디지털 비디오 디스플레이 인터페이스(100)의 일반화된 도면을 도시한다. 인터페이스(100)는 물리적 링크(106)(파이프라고도 불림)를 통해 송신기(102)를 수신기(104)에 연결한다. 본 실시예에서, 다수의 데이터 스트림(108-112)들이 송신기(120)에서 수신되어, 필요시, 각각을 대응하는 수의 데이터 패킷(114)들로 패킷화한다. 이 데이터 패킷들은 대응하는 데이터 스트림으로 형성되며, 각각의 데이터 스트림은 관련 가상 파이프(116-120)를 통해 수신기(104)에 전달된다. 각각의 가상 링크에 대한 링크 속도(즉, 데이터 패킷 트랜스퍼 속도)는 관련 링크 속도를 각기 가진 데이터 스트림들을 운반하는 물리적 링크(106)를 야기하는 특정 데이터 스트림에 대해 최적화될 수 있다. 데이터 스트림(110-114)은 비디오, 그래픽, 오디오 등과 같은 여러 형태를 취할 수 있다.

통상적으로, 소스가 비디오 소스일 경우, 데이터 스트림(110-114)은 컴포지트 비디오, 시리얼 디지털, 패럴렐 디지털, RGB, 또는 소비자 디지털 비디오 같은 잘 알려진 포맷을 가질 수 있다. 소스(102)가 아날로그 텔레비전 정지화상 카메라, 아날로그 VCR, DVD 플레이어, 캡코더, 레이저 디스크 플레이어, TV 투너, 셋톱박스(위성 DSS나 케이블 신호), 등등과 같은 아날로그 비디오 소스 형태를 포함할 경우 비디오 신호가 아날로그 비디오 신호일 수 있다. 소스(102)가 디지털 텔레비전(DTV), 디지털 정지화상 카메라, 등등과 같은 디지털 이미지 소스를 포함할 수도 있다. 디지털 비디오 신호는 SMPTE 274M-1995(1920x1080 해상도, 프로그레시브 또는 인터레이스 스캔), SMPTE 296M-1997(1280x720 해상도, 프로그레시브 스캔), 그리고 표준 480 프로그레시브 스캔 비디오 등의 잘 알려진 비디오 포맷들의 종류와 수를 취할 수 있다.

소스(102)가 아날로그 이미지 신호를 제공하는 경우에, 아날로그-디지털 컨버터(ADC)는 아날로그 전압이나 전류 신호를, 디지털 처리에 적합한 적절한 디지털 이미지 데이터 워드를 형성하는 디지털 방식으로 인코딩된 일련의 개별적인 수치(신호)로 변환한다. 다양한 ADC가 사용될 수 있다. 예를 들어, 그 외 다른 ADC는 Philips, Texas Instruments, Analog Devices, Brooktree, 등등에 의해 제작된 컨버터들을 포함할 수 있다.

예를 들어, 데이터 스트림(110)이 아날로그 신호일 경우, 송신기(102)에 포함되거나 송신기(102)에 연결된 아날로그-디지털 컨버터는 아날로그 데이터를 디지털화할 것이다. 이 디지털화된 데이터는 패킷화이저에 의해 패킷화되어, 디지털화된 데이터 스트림(110)으로부터 다수의 데이터 패킷(114)으로 변환될 것이다. 각각의 데이터 패킷은 가상 링크(116)를 통해 수신기(104)에 송신될 것이다. 수신기(104)는 데이터 패킷(114)들을 원본 포맷으로 적절하게 재조합함으로서 데이터 스트림(110)을 재구성할 것이다. 링크 속도가 원본 스트림 속도에 독립적임을 주목하여야 한다. 유일한 요건은 물리적 링크(106)의 링크 대역폭이 전송될 데이터 스트림의 총괄 대역폭보다 높아야 한다는 것이다. 본 실시예에서, 입력 데이터(비디오 데이터의 경우 화소 데이터)는 데이터 패킹 규정을 바탕으로 가상 링크에서 패킹(packing)된다. 이 방식으로, 물리적 링크(106) 또는 그 외 다른 구성 가상 링크(106)는 DVI같은 종래의 상호 연결에서처럼, 링크 문자 클럭당 한 개의 화소 데이터를 운반하지 않는다.

이 방식으로, 인터페이스(100)는 비디오 및 그래픽 데이터뿐 아니라 오디오 및 그 외 다른 애플리케이션 데이터의 트랜스포트를 위한 스케일러블 매체(scaleable medium)를 제공한다. 추가적으로, 발명은 핫-플러그 이벤트 감지 를 지원하며, 물리적 링크(또는 파이프)를 최적 전송 속도로 자동적으로 설정한다. 발명은 다중 플랫폼에 적합한 모든 디스플레이에 대한 로우 핀 카운트(low pin count), 순수 디지털 디스플레이 상호연결을 제공한다. 이러한 플랫폼들은 호스트 투 디스플레이, 램프/올인원, 그리고 HDTV 및 그 외 다른 소비자 전자 장치를 포함한다.

비디오 및 그래픽 데이터를 제공함에 추가하여, 디스플레이 타이밍 정보가 디지털 스트림에 내포되어, 완전하고 순간적인 디스플레이 정렬을 제공하고, "자동조정" 등과 같은 특징들에 대한 필요성을 경감시킨다. 발명에 따른 인터페이스의 패킷 기반 속성은 멀티미디어 애플리케이션에 대한 다중 비디오/그래픽 스트림 및 오디오 스트림들과 같은 다중의 디지털 데이터 스트림을 지원하는 스케일러빌리티(scalability)를 제공한다. 추가적으로, 주변 부착 장비 및 디스플레이 제어를 위한 USB 트랜스포트가 추가 케이블 설치에 대한 필요성 없이 제공될 수 있다.

발명에 따른 디스플레이 인터페이스에 대한 다른 실시예들이 아래에 설명될 것이다.

도 2는 도 1에 도시되는 시스템(100)을 바탕으로 한 시스템(200)으로서, 비디오 소스(202)와 비디오 디스플레이 유닛(204)을 연결하는 데 사용되는 시스템(200)의 도면이다. 도시되는 실시예에서, 비디오 소스(202)는 디지털 이미지(또는 디지털 비디오 소스)(206)와 아날로그 이미지(또는 아날로그 비디오 소스)(208) 중 한가지 또는 두가지 모두를 포함할 수 있다. 디지털 이미지 소스(206)의 경우에, 디지털 데이터 스트림(210)이 송신기(102)에 제공되고, 아날로그 비디오 소스(208)의 경우엔 ADC 컨버터 유닛(212)이 아날로그 데이터 스트림(213)을 대응하는 디지털 데이터 스트림(214)로 변환한다. 디지털 데이터 스트림(214)은 송신기(102)에 의해 디지털 데이터 스트림(210)과 같은 방식으로 처리된다. 디스플레이 유닛(204)은 아날로그 형 디스플레이일 수도 있고 디지털 형 디스플레이일 수도 있으며, 어떤 경우에는 아날로그나 디지털 신호 중 어느 것도 처리할 수 있다. 어느 경우에도, 디스플레이 유닛(204)은 아날로그 형 디스플레이의 경우에 디스플레이(218)와 디지털-아날로그 컨버터 유닛(DAC)(220)으로 수신기(104)와 면하는 디스플레이 인터페이스(216)를 포함한다. 본 실시예에서, 비디오 소스(202)는 어떤 형태로 취할 수 있고(가령, 레스크탑 컴퓨터, 디지털/아날로그 TV, 셋톱박스, 등등), 반면 비디오 디스플레이 유닛(204)은 비디오 디스플레이의 형태를 취할 수 있다(가령, LCD형 디스플레이, CRT형 디스플레이, 등등).

그러나 비디오 소스나 비디오 성크의 종류에 상관없이, 다양한 데이터 스트림들이 물리적 링크(106)를 통해 전송되기 전에 디지털화되고(필요시) 패킷화된다. 이 물리적 링크(106)는 동기 데이터 스트림을 위한 일방향 메인 링크(222)를 포함하고, 비디오 소스(202)와 비디오 디스플레이(204) 간의 라인 설정 및 그 외 다른 데이터 트래픽(가령, 다양한 링크 관리 정보, USB 데이터, 등등)을 위한 양방향 보조 채널(224)을 포함한다.

따라서 메인 링크(222)는 다중 동기 데이터 스트림(가령, 다중 비디오/그래픽 스트림과 다-채널 오디오 스트림, 등등)을 동시 전송할 수 있다. 본 실시예에서, 메인 링크(222)는 다수의 여러 다른 가상 채널들을 포함하며, 각각의 채널은 동기 데이터 스트림을 여러 기가비트/초(Gbps)로 이송할 수 있다. 논리적 관점에서, 메인 링크(222)는 물리적 단일 파이프로 나타나며, 이 물리적 단일 파이프 내에는 여러 가상 파이프들이 구축될 수 있다. 이 방식으로, 로직 데이터 스트림들이 물리적 채널에 할당되지 않고, 각각의 로직 데이터 스트림은 그 고유 로직 파이프(즉, 상술한 가상 채널)로 운반된다.

본 실시예에서, 메인 링크의 속도, 즉, 트랜스퍼 속도(222)는 링크 조건을 보상하도록 조정 가능하다. 예를 들어 한 구현에서, 메인 링크(222)의 속도는 1.0 Gbps에서 2.5Gbps까지의 범위 내에서 0.4 Gbps 증분양으로 조정될 수 있다(도 3 참조). 채널 당 2.5Gbps의 경우, 메인 링크(222)는 단일 채널에 대해 화소당 18비트의 칼라 깊이로 SXGA 60Hz를 지원할 수 있다. 채널 수 감소는 상호연결 비용만을 절감시키는 것이 아니라, 휴대용 장치처럼 소비전력에 민감한 장비에 대해 중요한 고려사항이 되는 전력 소모도 감소시킨다. 그러나, 채널 수를 4까지 증가시킴으로서, 메인 링크(222)는 60Hz에서 화소당 24비트의 칼라 깊이를 가지는 WQSXGA(3200 x 2048 이미지 해상도)나, 60Hz에서 화소당 18비트의 칼라 깊이를 가지는 QSXGA(2560x 2048 이미지 해상도)를 지원할 수 있다(데이터 압축없이). 채널당 최저 1.0 Gbps의 경우에도, 압축되지 않은 HDTV(즉, 1080i 또는 720p) 데이터 스트림을 지원하는 데 채널이 두 개만 요구된다.

본 실시예에서, 구성 가상 링크들의 총괄 대역폭보다 큰 대역폭을 가지도록 메인 링크 데이터 속도가 선택된다. 인터페이스에 전달되는 데이터는 원본 속도로 송신기에 도달한다. 수신기(104) 내의 시간-베이스 복원(TBR) 유닛(226)은 메인 링크 데이터 패킷들에 내포된 시간 스텝프들을 이용하여 스트림의 원본 속도를 재발생시킨다(필요시). 그러나, 적절하게 설정된 디지털 디스플레이 장치(232)의 경우에(도 2B 참조), 시간 베이스 복원은 불필요하다. 왜냐하면, 디스플레이 데이터가 링크 문자 클럭 속도로 디스플레이 드라이버 전자장치에 전달되어, 요구되는 채

널 수를 크게 감소시키기 때문이다. 이 때문에 디스플레이의 복잡도와 비용이 절감된다. 예를 들어, 도 2C는 어레이(240) 내에 선택된 디스플레이 소자들(238)을 구동하기 위해 행 드라이버(236)와 조합하여 사용되는 여러 열 드라이버(234)에 디스플레이 데이터가 파이프라인 처리되기 때문에, 어떤 시간 베이스 복원도 없은 방식으로 설정되는 일례의 LCD 패널(232)을 도시한다.

다른 실시예들은 링크 속도와 화소/오디오 클럭 속도에 대한 간단한 수치화 방법을 설명한다. 오늘날 존재하는 모든 표준 화소/오디오 클럭 주파수는 다음 마스터 주파수의 서브세트이다.

$$23.76 \text{ GHz} = 210 \times 33 \times 57 \times 111 \text{ Hz}$$

이는 화소(또는 오디오) 클럭 속도가 네 개의 매개변수 A, B, C, D로 표현될 수 있음을 의미한다.

$$\text{화소 클럭 속도} = 2A * 3B * 5C * 11D$$

$$A = 4\text{비트}, B = 2 \text{비트}, C = 3\text{비트}, \text{그리고 } D = 1\text{비트}$$

링크 속도(8B/10B 문자들같이 10비트 문자를 이용하는 링크의 경우 시리얼 링크 비트 속도/10)가 화소 클럭 속도와는 다를 수 있는 링크의 경우에도, 이 네 매개변수 A', B', C', D'으로 링크 속도를 규정하는 것에 장점이 있다. 이 장점은 링크 클럭으로부터 화소/오디오 클럭을 재생성하는 데 있어서의 단순함에 있다. 예를 들어, 링크 속도가 A'=6, B'=3, C'=7, D'=0으로 설정되고 대응하는 링크 속도가 135MHz라고 하자. 그러나, 화소 클럭 속도는 A=8, B=3, C=6, D=0 (= 135MHz)라고 가정할 경우, 화소 클럭은 링크 클럭으로부터 발생될 수 있다. 왜냐하면, 화소 클럭 속도가 링크 속도 * 22/51과 같기 때문이다.

시간 베이스 복원을 요하는 시스템으로 되돌아가서, 시간 베이스 복원 유닛(226)은 디지털 클럭 신세사이저로 구현될 수 있다. 비압축 비디오 스트림의 경우에, 시간 스템프는 아래에서 보다 상세하게 설명되는 바와 같이 패킷 헤더에 저장된다(20비트 값). 주어진 스트림의 경우에, 20비트 중 네 개가 각각의 헤더에 순차적으로 저장된다(TS3-0, TS7-4, TS11-8, TS15-12, TS19-16). 원본 스트림 주파수(Freq_native)는 링크 문자 클럭 주파수(Freq_link_char)로부터 다음과 같이 얻을 수 있다.

$$\text{Freq_native} = \text{Freq_link_char} * (\text{TS19-0})/220$$

송신기(102)는 220 사이클에 달하는 링크 문자 클럭 주파수 주기로 원본 스트림 클럭들의 수를 카운팅함으로서 시간 스템프를 발생시킨다. 카운터는 링크 문자 클럭의 매 220 사이클마다 그 값을 업데이트한다. 이 두 클럭들이 서로 비동기 관계에 있기 때문에, 시간 스템프 값은 시간에 따라 1만큼 변할 것이다. 업데이트를 사이에서, 송신기(102)는 주어진 패킷 스트림의 헤더에 도일한 시간 스템프를 반복적으로 전송할 것이다. 시간 스템프 값의 급격한 변화는 스트림 소스의 불안정한 조건의 표시로 수신기가 인식할 수 있다.

오디오 스트림에 대하여는 어떤 시간 스템프도 통신소통되지 않는다. 이 경우에, 소스 장치는 디스플레이 장치에 샘플 당 비트 수와 오디오 샘플 속도를 알린다. 오디오 속도를 가능한 낮게, 그리고 링크 문자 속도를 결정함으로서, 디스플레이 장치는 원본 오디오 스트림 속도를 재발생시킨다.

$$\text{오디오 속도} = (\text{오디오 샘플 속도}) \times (\text{샘플 당 비트 수}) \times (\text{채널 수})$$

도 4A에 도시되는 메인 링크 데이터 패킷(400)은 도 4B에 도시되는 메인 링크 패킷 헤더(402)를 포함한다. 메인 링크 헤더(402)는 16비트로 형성되며, 이때 비트 3-0은 스트림 ID(SID)(최대 스트림 카운트가 16임을 표시)이고 비트 4는 시간 스템프(TS) LSB이다. 비트 4가 1과 같을 경우, 이 패킷 헤더는 시간 스템프 값의 최소 유효 4비트를 가진다(비압축 비디오 스트림에 대해서만 사용됨). 비트5는 프레임 카운터의 최소 유효 비트로 기능하는 비디오 프레임 시퀀스 비트로서, 비디오 프레임 경계에서 0에서 1로, 또는 1에서 0으로 토글된다(비압축 비디오 스트림의 경우에만 사용됨). 비트 7과 6은 예비용으로 예약되고 비트 8-10은 4비트 CRC로서 앞선 8개의 비트에 대한 오류를 점검한다. 비트 15-12는 시간 스템프/스트림 ID 인버전(TSP/SIDn)이다. 이는 비압축 비디오를 위해 20비트 시간 스템프 값의 네 비트로 사용된다.

발명에 따른 인터페이스의 장점 중 하나는 서로 다른 포맷을 가질 수 있는 여러 다른 데이터 스트림들을 멀티플렉싱 할 수 있고, 또한, 다수의 서브 패킷들을 포함할 수 있는 메인 링크 데이터 패킷들을 가진다는 점이다. 예를 들어, 도 5A는 발명의 한 실시예에 따라 서브패킷 인클로저와 다중 패킷 멀티플렉싱을 제공하도록 배열되는 시스템(500)을 도시한다. 시스템(500)은 도 2에 도시되는 시스템(200)의 특정 실시예이며, 따라서, 발명의 범위를 제한하는 것으로 간주되어서는 안될 것이다. 시스템(500)은 멀티플렉싱된 데이터 스트림(506)을 형성하기 위해 스트림 1 보완 데이터 스트림(504)을 데이터 스트림(210)과 조합하는 데 사용되는 송신기(102)에 포함되는 스트림 소스 멀티플렉서(502)를 포함한다. 멀티플렉싱된 데이터 스트림(506)은 다수의 데이터 패킷(512)들로 형성되는 멀티플렉싱된 메인 링크 스트림(510)을 형성하기 위해 다수의 데이터 스트림 중 일부 또는 전부를 조합하는 링크 층 멀티플렉서(508)에 전달된다. 데이터 패킷(512) 중 일부는 다수의 서브패킷(514)을 포함할 수 있다. 링크 층 디멀티플렉서(516)는 멀티플렉싱된 데이터 스트림(510)을, 스트림 ID 및 관련 서브패킷 헤더에 따라 그 구성요소 데이터 스트림으로 분할한다. 이때, 스트림 쟁크 디멀티플렉서(518)는 서브패킷에 내포된 스트림 1 보완 데이터 스트림을 추가적으로 분리한다.

도 6은 세 개의 스트림이 메인 링크(222) 상에서 멀티플렉싱될 때 도 5에 도시되는 스트림(510)의 일례로 멀티플렉싱된 메인 링크 스트림(600)의 하이레벨 다이어그램이다. 세 스트림은 UXGA 그래픽(스트림 ID = 1), 1280x720p

비디오(스트림ID = 2), 그리고 오디오(스트림ID=3)이다. 메인 링크 패킷(400)의 패킷 헤더 크기가 작아서, 패킷 오버헤드를 최소화시키고, 따라서 매우 높은 링크 효율을 얻을 수 있다. 패킷 헤더가 이렇게 작을 수 있는 이유는 메인 링크(222) 상에서의 패킷 전송 이전에 보조 채널(224)을 통해 패킷 속성들이 전송될 수 있기 때문이다.

일반적으로, 서브패킷 인클로저는 메인 패킷 스트림이 비압축 비디오일 때 효과적인 기법이다. 왜냐하면 비압축 비디오 데이터 스트림은 비디오-블랭킹 주기에 대응하는 데이터 아이들을 주기들을 가지기 때문이다. 따라서, 비압축 비디오 스트림으로 형성되는 메인 링크 트래픽은 이 주기 중에 일련의 널(Null) 특수 문자 패킷들을 포함할 것이다. 여러 데이터 스트림들을 멀티플렉싱하는 능력에 집중함으로서, 본 발명의 일부 구현은 소스 스트림이 비디오 데이터 스트림일 때 메인 링크 속도와 화소 데이터 속도간의 차이를 보상하도록 여러 방법을 이용한다. 예를 들어, 도 7에 도시되는 바와 같이, 화소 데이터 속도는 0.5 Gb/sec이고, 따라서 화소 데이터의 한 비트가 2ns마다 전송된다. 본 예에서, 링크 속도는 1.25 Gb/sec로 설정되었고, 따라서 화소 데이터의 한 비트가 0.8 ns마다 전송된다. 여기서, 송신기(102)는 도 8에 도시되는 바와 같이 화소 데이터 간에 특수 문자를 산개(intersperse)시킨다. 두 특수 문자가 화소 데이터 P1의 제 1 비트와 화소 데이터 P2의 제 2 비트 사이에 배치된다. 특수 문자들은 수신기가 화소 데이터의 각 비트를 구분할 수 있게 한다. 화소 데이터의 비트들 사이에 특수 문자들을 산개하게 함으로서, 링크가 동기화를 유지할 수 있도록 하는 일정한 스트림의 데이터가 생성된다. 본 예에서, 특수 문자들은 널 문자들이다. 이러한 방법을 위해 어떤 라인 버퍼도 필요하지 않으며, 단지 작은 FIFO만이 필요하다. 왜냐하면 링크 속도가 충분히 빠르기 때문이다. 그러나, 비디오 신호를 재구성하기 위해 수신측에 더 많은 로직이 요구된다. 수신기는 특수 문자가 시작하고 종료되는 시기를 인지할 필요가 있다.

산개 방법에 대한 대안은 널 값처럼 일련의 화소 데이터 비트들을 특수 문자들과 교대로 나타나게 하는 것이다. 예를 들어, P1 – P4가 송신기(104)에 포함된 라인 버퍼에 공급될 수 있고, 그후 더 많은 화소 데이터가 사용해질 때까지 한 개 이상의 널 값이 상기 버퍼에 공급될 수 있다. 이러한 구현은 상술한 산개 방법보다 비교적 큰 버퍼 공간을 요한다. 이러한 여러 구현에서, 라인 버퍼를 채우는 데 요구되는 시간은 라인 버퍼가 채워진 후 데이터를 송신하는 데 걸리는 시간보다 길다. 왜냐하면 링크 속도가 비교적 빠르기 때문이다.

도 5A를 참고하여 설명한 바와 같이, 발명에 따른 인터페이스의 장점 중 한가지는 여러 데이터 스트림들을 멀티플렉싱할 수 있을 뿐 아니라, 다수의 서브 패킷들 중 일부 또는 전부를 특정 메인 링크 데이터 패킷에 인클로징 할 수 있다는 것이다. 도 9A는 발명의 한 실시 예에 따른 표본 서브패킷(900)을 도시한다. 서브패킷(900)은 서브패킷 헤더(902)를 포함한다. 이 헤더(902)는 본 실시 예에서 2바이트이고 SPS(서브패킷시작) 특수 문자를 동반한다. 서브패킷(900)을 인클로징하는 메인 링크 데이터 패킷이 서브패킷(900)에 추가하여 패킷 페이로드를 지니는 경우에, 서브패킷(900)의 종점은 SPE(서브패킷 종료) 특수 문자에 의해 표시되어야 한다. 그렇지 않을 경우, 메인 패킷의 종점(도 9B에 도시되는 예에서 COM 문자에 의해 표시됨)은 서브패킷(902)의 종점과, 서브패킷을 인클로징하는 메인 패킷을 표시한다. 그러나, 서브패킷은 그 인클로징 메인 패킷이 페이로드를 구비하지 않을 때 SPE로 끝날 필요가 없다. 도 9B는 발명의 한 실시 예에 따른 메인 링크 패킷 내의 일례의 서브패킷 포맷을 도시한다. 헤더 필드와 서브패킷 페이로드의 형성은 서브패킷(902)을 이용하는 특정 애플리케이션 프로파일에 따라 좌우된다.

서브패킷 인클로저 이용의 특별히 유용한 예는 도 10에 도시되는 비압축 그래픽 이미지(1000)의 선택적 리프레시에 있다. 전체 프레임(1002)의 속성들은 보조 채널(224)을 통해 전송될 것이다. 왜냐하면 스트림이 유효하게 유지되는 한 이 속성들이 일정 상태로 머무르기 때문이다. 선택적 리프레시 동작에서, 이미지(1000)의 일부분(1004)만이 비디오 프레임마다 업데이트된다. 업데이트된 사각형(즉, 1004)의 네 개의 X-Y 좌표는 매 프레임마다 전송되어야 한다. 왜냐하면, 사각형의 값들이 프레임마다 변하기 때문이다. 또 다른 예는 256 칼라 그래픽 데이터에 요구되는 칼라 탐색표(CLUT) 데이터의 전송이다. 이 때, 8비트 화소 데이터가 256-엔트리 CLUT에 대한 입력이고, CLUT의 콘텐트는 동적으로 업데이트되어야 한다.

단일 양방향 보조 채널(224)은 USB 트래픽과 같은 보조 애플리케이션 데이터를 운반하면서 메인 링크 동작을 지원하기 위한, 그리고 링크 설정에 유용한 여러 지원 기능들에 대한 도관을 제공한다. 예를 들어, 보조 채널(224)로, 디스플레이 장치는 동기화 손실, 누락된 패킷, 그리고 트레이닝 세션의 결과와 같은 이벤트들을 소스 장치에 알릴 수 있다. 예를 들어, 특정 트레이닝 세션이 실패할 경우, 송신기(102)는 실패한 트레이닝 세션의 지정된 결과를 바탕으로 메인 링크 속도를 조정한다. 이 방식으로, 조정 가능한 고속 메인 링크를 비교적 저속의 신뢰도 높은 보조 채널과 조합함으로서 생성되는 페루프에 의해, 다양한 링크 조건에서 견고한 동작이 가능해진다. 일부 경우에, 소스 장치(202)로부터 싱크 장치(204)까지 데이터를 이송하기 위해 메인 링크(222)의 대역폭의 일부분(522)을 이용하여 도직 양방향 보조 채널(520)이 구축될 수 있으며, 싱크 장치(204)로부터 소스 장치(202)까지 일방향 백 채널(524)이 구축될 수 있다. 일부 애플리케이션의 경우에, 이로직 양방향 보조 채널을 이용하는 것이 도 5A에 기술된 해프-듀플렉스 양방향 채널을 이용하는 것보다 바람직 할 수 있다.

실제 패킷 데이터 스트림의 전송을 개시하기 전에 송신기(102)는 모뎀의 링크 설정의 개념 측면에서 유사한 링크 트레이닝 세션을 통해 안정한 링크를 구축한다. 링크 트레이닝 중, 메인 링크 송신기(102)는 지정 트레이닝 패턴을 전송하여, 수신기(104)는 솔리드 비트/문자 락을 구현할 수 있는지를 결정할 수 있게 한다. 본 실시 예에서, 송신기(102)와 수신기(104) 간의 트레이닝 관련 핸드쉐이킹(handshaking)은 보조 채널에서 실행된다. 링크 트레이닝 패턴의 한 예가 발명의 한 실시 예에 따라 도 11에 도시되어 있다. 도시되는 바와 같이, 트레이닝 세션 중에, 위상 1은 최단 동작 길이를 나타내고, 위상 2는 등화기(equalizer) 최적화를 위해 수신기에 의해 사용되는 최장 동작 길이를 나타낸다. 위상 3에서는, 링크 품질이 합리적이기만 하다면 비트 락과 문자 락이 모두 구현된다. 일반적으로, 트레이닝 주기는 약 10ms이고, 이 시간동안 대략 107비트의 데이터가 전송된다. 수신기(104)가 솔리드 락(solid lock)을 구현하지 않을 경우, 이는 보조 채널(224)을 통해 송신기(102)에 연락하고, 송신기(102)는 링크 속도를 감소시켜서 트레이닝 세션을 반복한다. 트레이닝 세션 도관을 제공함에 추가하여, 보조 채널(224)은 메인 링크 패킷 스트림 세부사항을 운반하는 데 사용될 수 있어서, 메인 링크(222)에서의 패킷 전송의 오버헤드를 크게 감소시킬 수 있다. 더욱이, 보조 채널(224)은 모든 모니터에서 발견되는 디스플레이 데이터 채널(DDC)을 대체하는 확장 디스플레이 식별 데이터(EDID)를 운반하도록 설정될 수 있다(EDID는 모니터에 관한 기본 정보와 그 기능을 지닌 VESA 표준 데이터 포맷이다). 그 예로는 벤더 정보, 최대 이미지 크기, 칼라 특성, 공장 프리셋 타이밍, 주파수 범위 한도, 모니터 명칭 및 시리얼 넘버에 대한 문자 스트링 등이 있다. 이 정보는 디스플레이에 저장되며, 모니터와 PC 그래픽 어댑터

사이에 자리하는 DDC를 통해 시스템과 통신하는 데 사용된다. 이 시스템은 이 정보를 구성 용도로 사용하여, 모니터와 시스템이 함께 동작할 수 있다). 확장 프로토콜 모드라 불리는 경우에, 보조 채널은 비동기 및 동기 패킷들을 모두 운반하여, 키보드, 마우스, 마이크로폰 등의 추가적인 데이터 타입을 지원할 수 있다.

도 12는 발명의 한 실시예에 따라 데이터 스트림 스케줄러(1202)를 구비한 표본 시스템(1200)의 도면이다. 시스템(1200)은 도 5A와 5B를 참고하여 설명한 시스템(500)을 바탕으로 한다. 따라서 발명의 여러 구현 중 한가지에 해당할 따름이다. 따라서, 스트림 스케줄러(1202)는 비디오 소스(202)에 포함되거나 비디오 소스(202)에 연결되며, 다시 멀티플렉서(1204)와 데이터 버퍼(1206)에 연결된다. 이는 스케줄러 사이클 시간 T_{sched} 로 불리는 시간 주기 동안 링크 데이터 스트림(1208)을 채우는 데 사용되는 입력 데이터 스트림(S1, S2, S3)의 일부분만을 저장하는 데 적합한 것이다. 본 실시예에서, 데이터 버퍼(1206)는 FireWall 같은 동기식 비디오 트랜스퍼 프로토콜을 일반적인 60K바이트에 반해 수십바이트 크기에 지나지 않는다. 이 방식으로, (가용 링크 대역폭에 의해 나누어지는 데 데이터 스트림들의 페이로드 대역폭의 총괄을 비교함에 바탕하여) 링크 효율이 90% 이상이 된다.

본 실시예에서, 링크 데이터 스트림(1208)은 각각의 데이터 스트림 S1, S2, S3로부터 크기 데이터 패킷 P1, P2, P3를 조합하기 위해 시간 분할 멀티플렉싱을 이용함으로서 스케줄러(1202)에 의해 형성된다. 패킷 P1, P2, P3의 크기는 링크 대역폭에 관련된 각각의 데이터 스트림에 대한 상대적 대역폭을 반영한다. 상술한 바와 같이, 각각의 데이터 패킷의 크기는 특정 데이터 스트림 비트 속도(SBR)와 링크 비트 속도(LBR)의 함수이다. 특히, 특정 데이터 스트림 비트 속도가 클수록, 특정 데이터 패킷 크기가 크다(표 1 참조). 예를 들어, 링크 비트 속도 LBR이 2.5 Gbps이고 최대 패킷 크기가 80 링크 심벌이며 표 1의 가정을 이용할 경우(한개의 링크 심벌은 링크 클럭당 데이터 유닛으로 규정되며, 통상적으로 4ns이다), 스트림 S1에 관련된 데이터 패킷 P1은 32 링크 심벌이고 스트림 S2 및 S3에 관련된 데이터 패킷 P2와 P3는 10 링크 심벌과 8 링크 심벌에 각각 해당한다. 각각의 스케줄러 사이클 시간 T_{sched} 의 시점(beginning)에서, 스케줄러(1202)는 2개의 링크 심벌의 크기인 수신기(204)에 대한 정렬 툴을 제공하는 인터레이크 정렬 패킷(Interlane Alignment Packet; ILA)을 삽입한다. 따라서 본 예에서, 스케줄러 사이클 시간 T_{sched} 은 52개의 링크 심벌(즉, 32+ 10+ 8+ 2)에 해당한다. 이는 각각의 링크 심벌이 대략 4ns를 나타날 때 대략 208 ns로 변환된다.

각각의 데이터 패킷 P가 스트림 비트 속도 SBR과 링크 비트 속도 LBR에 관련된 데이터 심벌 D와 스터핑 심벌 N의 수에 대한 관련 액티브 데이터 비율을 가진다(아래 참조).

$$SBR/LBR = D/(D+N)$$

따라서, 스케줄러 사이클 시간 T_{sched} 의 증가(또는 감소)를 불러오도록 데이터 스트림이 추가(또는 삭제)된 경우에, 패킷 크기 P는 데이터 심벌 D의 수에 관련하여 스터핑 심벌 N의 수를 변화시킴으로서 일정하게 유지된다. 데이터 스트림 추가(삭제) 이후, 다른 데이터 스트림에 대해 특정 패킷 크기가 일정하게 유지되고 스케줄러 사이클 시간이 증가(또는 감소)하기 때문에, 스터핑 심벌 N의 수는 T_{sched} 의 변화와 같은 양으로 증가(또는 감소)한다. 단일 데이터 스트림나이 유지되는 "디제너레이트(degenerate)"의 경우에는 어떤 스터핑 심벌 N도 존재하지 않는다.

도 13은 한 실시예에 따라 데이터 스트림(1210)의 상세 부분(1300)을 도시한다. 특히 도 13은 표 1에 도시되는 값들을 이용하여 데이터 스트림(1208)에 대한 데이터 심벌 D와 스터핑 심벌 N의 배열을 도시한다. 특정 데이터 스트림을 동적으로 추가하거나 삭제함으로서, 나머지 데이터 스트림에 대한 특정 패킷 크기들이 불변으로 유지된다. 따라서 도 14는 20개의 링크 심벌들의 패킷 크기 P4에 대응하는 0.625 Gbps의 스트림 비트 속도를 가진 제 4 데이터 스트림 S4의 추가를 도시한다. 이로 인해 T_{sched} 가 52에서 72까지의 링크 심벌 수로 증가한다. 이는 4ns와 같은 링크 심벌의 경우 288 ns에 대응한다. 그러나, 특정 패킷 크기 P1, P2, P3를 일정하게 유지하기 위해, 각각의 데이터 패킷에 대하여 스터핑 심벌 N의 수는 증가한다. 역으로, 데이터 스트림이 삭제되는 경우에(가령, S3), 스터핑 심벌 N의 수에 관련하여 데이터 심벌 D의 수의 증가와 같은 양으로 스케줄러 사이클 시간 T_{sched} 가 감소할 것이다. 모든 데이터 스트림이 한 개씩 절감하는 "디제너레이트"의 경우에, 나머지 데이터 패킷은 어떤 스터핑 심벌 N도 가지지 않으며, 따라서, 어떤 버퍼도 필요하지 않다. 결과적으로, 상술한 라스터 스캔 트랜스포트 프로토콜을 시뮬레이션한다.

또한, 링크 데이터 스트림(1208)이 단 한 개의 비압축 비디오 스트림일 때(도 15 참조) 디제너레이트 연결의 경우에, ILA 패킷들은 S1의 아이들 주기(수평 블랭킹 영역)에 위치하고, 액티브 디스플레이 영역은 데이터 심벌 D와 스터핑 심벌 N의 믹스처에 의해 표현된다(도 16 참조).

내포된 시간 스텝프를 데이터 심벌 D의 상대적 수치가 제공한다는 점에 주목하여야 한다. 즉, 특정 데이터 스트림에 대한 데이터 심벌 D의 수를 특정 데이터 스트림에 관련없는 데이터 심벌들의 수에 관련하여 카운팅함으로서, 관심 대상인 데이터 스트림에 대한 스트림 클럭을 제공한다. 예를 들어, 도 13에서의 경우에, 특정 데이터 스트림에 대한 스트림 클럭 $F_{\text{stream_clk}}$ 를 복원하기 위해, 스터핑 심벌과 스트림 데이터 심벌(P)의 총 숫자에 대한 스트림 데이터 심벌(M)의 수를 단순히 결정함으로서 결정할 수 있다. 특히 스트림 클럭 $F_{\text{stream_clk}}$ 는 다음과 같이 결정된다.

$$F_{\text{stream_clk}} = (M/P) * F_{\text{link_clk}}$$

이때, M과 P는 수신기(204)에 의해 측정될 수 있다.

도 17은 발명의 한 실시예에 따라 다수의 데이터 스트림들을 스케줄링하는 과정(1700)의 순서도이다. 이 과정(1700)은 수신기에 스트림 속성 데이터를 전송하는 송신기에 의해 시작된다(1702). 본 실시예에서, 속성 데이터는 송신기를 수신기에 연결하는 보조 링크를 통해 전송된다. 그후 단계 1704에서, 수신기는 입력 데이터 스트림을 수신하여 처리하기 위해 충분한 리소스들을 할당한다. 송신기가 데이터 스트림을 수신한 후(단계 1706), 스트림 트랜스포트 스케줄러는 링크 데이터 스트림을 형성하고(1708), 링크 데이터 스트림은 송신기에 의해 메인 링크를 통해

수신기에 전달된다(1710). 그후, 추가적인 데이터 스트림이 추가되고(1712), 수신기가 추가 데이터 스트림을 수신하고 처리할 수 있을 때(1714), 새로운 데이터 스트림이 투명하게 추가되며(1716), 앞서 전송한 데이터 스트림의 데이터 패킷 크기가 일정하게 유지된다.

도 18은 발명의 한 실시예에 따라 링크 데이터 스트림을 형성하는 과정(1800)의 순서도이다. 이 과정(1800)은 처리 과정(1700)의 형성 동작(1708)의 특징 구현에 해당한다. 따라서, 트랜스포트 스케줄러는 메인 링크에 대한 링크 비트 속도를 결정하고(1802), 송신기로부터 수신기까지 상기 링크를 통해 이송될 각각의 데이터 스트림에 대한 스트림 비트 속도를 결정한다(1804). 그후 1806에서, 각각의 데이터 스트림들에 대한 패킷 크기가 스트림 비트 속도, 링크 비트 속도, 그리고 지정 최대 패킷 크기를 바탕으로 결정된다. 그후 단계 1808에서, 각각의 데이터 스트림에 대한 데이터 패킷이 형성되며, 각각의 데이터 패킷은 다수의 데이터 심벌들과 스터핑 심벌들을 포함한다. 데이터 패킷들이 형성되고 단일 트랜스포트 스케줄러 사이를 시간동안, 트랜스포트 스케줄러는 수신기에 전송될 데이터 스트림들 각각으로부터의 데이터 패킷을 연결하고(1810), 연결된 데이터 패킷에 인터레인 정렬 패킷(ILA)을 추가한다(1812). 단계 1814에서, 트랜스포트 스케줄러는 연결된 데이터 패킷들을 수신기에 송신할 것을 송신기에 신호하고, 단계 1816에서 결정되는 바와 같이 송신 완료시까지 단계 1808-1814를 반복한다.

도 19는 발명의 한 실시예에 따른 시스템(200)의 로직 총배열(1900)을 도시한다. 정확한 구현이 애플리케이션마다 다를 수 있지만, 소스(가령, 비디오 소스(202))는 소스 물리층(1902)으로 형성되고, 송신기 하드웨어, 소스 링크층(1904), 그리고 데이터 스트림 소스(1906)를 포함한다. 소스 링크층(1904)은 하드웨어 및 상태 머신(또는 펌웨어)의 멀티플렉싱을 포함하고, 그리고 데이터 스트림 소스(1906)로는 오디오/비디오/그래픽 하드웨어 및 관련 소프트웨어 등등이 있다. 마찬가지로 디스플레이 장치는 물리층(1908)(여러 수신기 하드웨어를 포함), 싱크 링크층(1910)(하드웨어 및 상태 머신(또는 펌웨어)의 디멀티플렉싱), 그리고 스트림 싱크(1912)(디스플레이/타이밍 컨트롤러 하드웨어 및 부가적 펌웨어를 포함)를 포함한다. 소스 애플리케이션 프로파일층(1914)은 소스가 링크층(1904)과 통신하는 포맷을 규정하며, 마찬가지로, 싱크 애플리케이션 프로파일층(1916)은 싱크(1912)가 싱크 링크층(1910)과 통신하는 포맷을 규정한다.

여러 다양한 층들이 아래에서 보다 상세하게 설명될 것이다.

소스 장치 물리층

본 실시예에서, 소스 장치 물리층(1902)은 전기 서브층(1902-1)과 로직 서브층(1902-2)를 포함한다. 전기 서브층(1902-1)은 핫플러그/언플러그 검출 회로, 드라이버/수신기/종료 저항, 패럴렐 투 시리얼/시리얼 투 패럴렐 컨버터, 그리고 스프래드스펙트럼 가능 PLL 등과 같은 인터페이스 초기화 및 동작을 위한 모든 회로들을 포함한다. 로직 서브층(1902-2)은 메인 링크(222)용 256개의 링크 데이터 문자와 12개의 제어 문자(도 13 참조)와 보조 채널(224)용의 Manchester II(도 21 참조)를 제공하는 8B/10B(ANSI X3.230-1994, 11절 참조)과 같은 데이터 인코딩/디코딩, 시간 베이스 복원 회로, 링크 트레이닝용 패턴 발생, 데이터 스크램블링/디스크램블링, 패킷화-탈패킷화를 위한 회로들을 포함한다.

8B/10B 인코딩 알고리즘은 미국특허 4,486,739 호에 기재되어 있고 그 내용이 본원에서 참고로 인용된다. 당분야에 잘 알려진 바와 같이, 8B/10B 코드는 시리얼 전송을 위해 8비트 데이터 블록들을 10비트 코드 워드로 인코딩하는 블록 코드이다. 추가적으로, 8B/10B 전송 코드는 임의적인 1과 0들의 바이트 폭 데이터 스트림을 최대 동작 길이가 5인 DC 균형의 스트림(1과 0)으로 변환한다. 이러한 코드는 송수신기(110)와 같은 수신기에 의한 신뢰성 있는 클럭 복원이 가능하도록 충분한 신호 전이를 제공한다. 더욱이, DC 균형의 데이터 스트림은 광섬유 및 전자기 도선 연결에 장점을 보인다는 것이 드러났다. 시리얼 스트림에서 1과 0의 평균 수치들이 동일하거나 거의 동일한 수준으로 유지된다. 8B/10B 전송 코드는 1과 0들의 수 사이의 불균형(disparity)을 6과 4비트 블록 경계 사이에서 -2, 0, 또는 2가 되도록 제약한다. 코딩 기법은 명령 코드라 불리는 신호처리를 위한 추가 코드를 또한 구현한다.

비암축 디스플레이 데이터에 의해 나타나는 반복 비트 패턴들을 피하기 위해(그래서 EMI를 줄이기 위해), 메인 링크(222) 상에서 전송되는 데이터는 8B/10B 인코딩 이전에 먼저 스크램블링된다. 트레이닝 패킷과 특수 문자들을 제외한 모든 데이터가 스크램블링될 것이다. 스크램블링 기능은 선형 피드백 시프트 레지스터(LFSR)로 구현된다. 데이터 인크립션에 구현되면, LFSR 시드의 초기값이 인크립션 키 세트에 따라 좌우된다. 인크립션없는 데이터 스크램블링의 경우, 초기값은 고정될 것이다.

데이터 스트림 속성들이 보조 채널(224) 상에서 전송되기 때문에, 메인 링크 패킷 헤더들은 스트림 식별 수치로 기능하며, 따라서, 오버헤드를 감소시키고 링크 대역폭을 최대로 한다. 메인 링크(222)나 보조 링크(224)가 개별적인 클럭 신호 라인을 가지지 않는 점에 주목하여야 한다. 이 방식으로, 메인 링크(222)와 보조 링크(224) 상의 수신기들은 데이터를 샘플링하여 입력 데이터 스트림으로부터 클럭을 추출한다. 수신기 전기 서브층에서 위상 락 루프(PLL)에 회로에 대한 고속 위상 라킹은 중요하다. 왜냐하면, 보조 채널(224)이 해프-듀플렉스 양방향이고, 트래픽의 방향이 자주 변하기 때문이다. 따라서, 보조 채널 수신기 위상 락에서의 PLL은 Manchester II (MII) 코드의 빈번한 그리고 균일한 신호 전송 덕택에 16 데이터 주기만큼이나 작다.

링크 설정 시간에서, 메인 링크(222)의 데이터 속도는 보조 채널(224)에서의 핸드쉐이크를 이용하여 조율된다. 이 처리 중, 공지된 트레이닝 패킷들의 세트가 최고 링크 속도로 메인 링크(222) 상에서 전송된다. 보조 채널(224)을 통해 송신기(102)에 성공 또는 실패가 다시 전달된다. 트레이닝이 실패할 경우, 메인 링크 속도가 감소하고 성공시까지 트레이닝이 반복된다. 이 방식으로, 소스 물리층(1902)은 케이블 문제에 보다 내성을 지니며, 따라서, 애플리케이션들에 대한 외부 호스트의 모니터링에 대해 보다 적합하다. 그러나, 기존 디스플레이 인터페이스들과는 달리, 메인 채널 링크 데이터 속도가 화소 클럭 속도와 분리된다. 링크 데이터 속도는 전송된 스트림들의 총과 대역폭을 링크 대역폭이 뛰어넘도록 설정된다.

소스 장치 링크층

소스 링크총(1904)은 링크 초기화 및 관리를 취급한다. 예를 들어, 소스 물리총(1902)으로부터 모니터 케이블의 연결이나 모니터 전력 연결시 발생되는 핫플러그 검출 이벤트를 수신하였을 때, 소스 장치 링크총(1904)은 보조 채널(224)에서의 상호교환을 통한 수신기의 기능을 평가하여, 트레이닝 세션에 의해 결정되는 바와 같이 최대 메인 링크 데이터 섹도를 결정하고, 수신기에서의 시간 베이스 복원 유닛들의 수를 결정하며, 양 측에서의 사용 범위를 결정하고, USB 확장의 사용성을 결정하며, 그리고 관련 핫플러그 이벤트를 스트림 소스(1906)에 알린다. 추가적으로, 스트림 소스(1906)로부터 요청에 따라, 소스 링크총(1904)은 보조 채널(224)을 통해 수신기(104)에 스트림 속성들을 전송하고, 요청한 데이터 스트림들을 취급할 만한 리소스들을 메인 링크(222)가 충분히 가지는지를 스트림 소스(1904)에 알리며, 동기화 손실 및 범위 오버플로 같은 링크 실패 이벤트들을 스트림 소스(1904)에 알리고, 그리고 스트림 소스(1904)에 의해 제안되는 MCCS 명령들을 보조 채널(224)을 통해 수신기(1904)에 전달한다. 소스 링크총(1904)과 스트림 소스/싱크 간의 모든 통신들은 애플리케이션 프로파일총(1914)에서 규정된 포맷들을 이용한다.

애플리케이션 프로파일총(소스 및 싱크)

일반적으로, 애플리케이션 프로파일총은 스트림 소스(또는 싱크)가 관련 링크총과 면할 포맷을 규정한다. 이 포맷들은 애플리케이션 프로파일총에 의해 규정되어 다음의 분류로 나누어진다. 즉, 애플리케이션 독립형 포맷(링크 상태 질의용 링크 메시지)과, 애플리케이션 의존형 포맷(메인 링크 데이터 매핑, 수신기용 시간 베이스 복원 방정식, 싱크 기능/스트림 속성 메시지 서브패킷 포맷). 애플리케이션 프로파일총은 다음의 칼라 포맷들을 지원한다. 즉, 24비트 RGB, 16비트 RG2565, 18비트 RGB, 30비트 RGB, 256칼라 RGB(CLUT기반), 16비트 CbCr422, 20비트 YCbCr422, 그리고 24비트 YCbCr444를 지원한다.

예를 들어, 디스플레이 장치 애플리케이션 프로파일총(APL)(1914)은 인터페이스(100)에 전달되거나 인터페이스(100)로부터 수신되는 데이터에 대한 제시 포맷을 포함하는 메인 링크(222) 상의 스트림 소스/싱크 통신을 위한 포맷을 기술하는 애플리케이션-프로그래밍 인터페이스(API)이다. APL(1914)의 일부 태양이 베이스라인 모니터 기능이기 때문에, 이들은 인터페이스(100)의 모든 이용에 공통이다. 반면, 다른 비-베이스라인 모니터 기능(가령, 데이터 매핑 포맷 및 스트림 속성 포맷)은 전송될 동기식 스트림의 종류나 애플리케이션에 대해 독자적이다. 애플리케이션에 관계없이, 스트림 소스(1904)는 소스 링크총(1914)에 질의하여, 메인 링크(222)에서 어떤 패킷 스트림 송신을 시작하기 전에 계류중인 데이터 스트림을 메인 링크(222)가 취급할 수 있는지를 확인한다.

메인 링크(222)가 계류중인 패킷 트리밍을 지원할 수 있다고 결정되면, 스트림 소스(1906)는 스트림 속성을 소스 링크총(1914)에 전달하고, 이는 다시 보조 채널(224) 상에서 수신기에 전달된다. 이 속성들은 수신기에 의해 사용되는 정보로서, 특정 스트림의 패킷들을 식별하고, 상기 스트림으로부터 원본 데이터를 복원하며, 그리고 스트림의 원본 데이터 속도로 이를 다시 포매팅한다. 데이터 스트림의 속성은 애플리케이션에 의존한다.

요망 대역폭이 메인 링크(222) 상에서 사용하지 않은 경우에, 스트림 소스(1914)는 이미지 리프레시 속도나 칼라 깊이를 감소시킴으로서(일례에 불과) 정확한 액션을 취할 수 있다.

디스플레이 장치 물리총

디스플레이 장치 물리총(1916)은 디스플레이 장치 링크총(1910)과 디스플레이 장치 APL(1916)을 링크 데이터 송신/수신에 사용되는 신호처리 기술로부터 분리시킨다. 메인 링크(222)와 보조 채널(224)은 독자적인 물리총들을 가지며, 각각의 물리총은 커넥터 명세를 포함하는 전기서브총과 로직서브총으로 구성된다. 예를 들어, 해프-듀플렉스 양방향 보조 채널(224)은 도 22에 도시되는 바와 같이 링크 양쪽 말단에 송신기와 수신기를 각각 구비한다. 보조 링크 송신기(2902)에는 로직서브총(1908-1)에 의해 링크 문자들이 제공되며, 이들은 시리얼(직렬)화되어 대응하는 보조 링크 수신기(2904)에 계로 송신된다. 수신기(2904)는 보조 링크(224)로부터 시리얼 링크 문자를 수신하여, 링크 문자 클럭 속도로 데이터를 탈-시리얼화한다. 소스 로직서브총들의 주된 기능들은 신호 인코딩, 패킷화, 데이터 스크램블링, 그리고 송신 포트에 대한 트레이닝 패턴 발생을 포함한다. 수신 포트의 경우, 수신기 로직서브총의 주된 기능들은 신호 디코딩, 탈-패킷화, 데이터 디스크램블링, 그리고 시간 베이스 복원을 포함한다.

보조 채널 로직서브총의 주된 기능들은 데이터 인코딩과 디코딩, 데이터의 프레임 형성 및 프레임 해제(de-framing)를 포함하며, 보조 채널 프로토콜에는 두 가지 옵션이 있다. 즉, 독립형 프로토콜은 링크총 상태 머신이나 펌웨어에 의해 관리될 수 있는 경량 프로토콜이고, 확장 프로토콜은 데이터-체인 싱크 장치 같은 구조와 USB 트래픽 같은 다른 데이터 종류를 지원할 수 있다. 데이터 인코딩 및 디코딩 기법은 프로토콜에 상관없이 동일하지만, 데이터의 프레임 형성은 둘 사이에 차이가 있다.

도 29에서, 보조 채널 전기서브총은 송신기(2902)와 수신기(2904)를 포함한다. 송신기(2902)에는 로직서브총에 의해 링크 문자들이 제공되고, 이는 직렬화되어 송신된다. 수신기(2904)는 링크총으로부터 시리얼화된 링크 문자를 수신하여, 이를 링크 문자 클럭 속도로 탈-시리얼화한다. 보조 채널(224)의 양의 신호 및 음의 신호들은 도시되는 바와 같이 링크의 각각의 말단에서 50오옴 단자 저항을 통해 접지부에서 종료된다. 본 구현에서, 구동 전류는 대략 8mA에서 24mA까지 범위로 링크 조건에 따라 프로그래밍 가능하고, 이는 대략 400mV에서 1.2V의 $V_{\text{differential_pp}}$ 의 범위를 도출한다. 전기 아이들 모드에서는 양의 신호도 음의 신호도 구동되지 않는다. 전기 아이들 상태로부터 송신을 시작할 때, SYNC 패턴이 송신되어야 하고 링크가 재구축되어야 한다. 본 실시예에서, SYNC 패턴은 보조 채널 차동 상 신호들을 클럭 속도 28배에 이어지는 네 개의 1(Manchester II 코드)로 토클링하는 구성을 이룬다. 소스 장치의 보조 채널 마스터는 보조 채널(224)의 양의 신호 및 음의 신호를 주기적으로 구동시키거나 측정함으로서 핫플러그(hot-plug)와 핫언플러그(hot-unplug)를 검출한다.

메인 링크

본 실시예에서, 메인 링크(222)는 국부 결정 주파수의 정수배인 개별적인 가변 링크 속도를 지원한다(도 3의 24-MHz의 국부 결정 주파수와 조화되는 표본 세트의 링크 속도 참조). 도 16에 도시되는 바와 같이, 메인 링크(222)(일방향 채널)는 소스 장치에서 송신기(1602)만을, 그리고 디스플레이 장치에서 수신기(1604)만을 구비한다.

도시되는 바와 같이, 케이블(2304)은 꼬인 도선 쌍들의 세트를 포함한다. R, G, B 비디오 신호들에 대한 각각은 전형적인 RGB 칼라 기반 비디오 시스템(가령, PAL 기반 TV 시스템)에서 제공된다. 당 분야에 잘 알려진 바와 같이, 꼬인 쌍의 케이블은 서로의 둘레로 꼬인, 독립적으로 절연된 두 도선들로 구성되는 종류의 케이블이다. 한 개의 도선은 신호를 운반하고 다른 하나의 도선은 접지되어 신호 간섭을 흡수한다. 일부 다른 시스템에서, 신호들은 NTSC 비디오 TV 시스템들에 사용되는 성분 기반 신호들(Pb, Pr, Y)일 수도 있다. 케이블 내에서 각각의 꼬인 쌍은 개별적으로 차폐된다. +12V 전력 및 접지를 위한 두 핀들이 제공된다. 전체 케이블이 또한 차폐된다. 이 외부 실드와 개별 실드들은 두 말단에서 커넥터 헬에 쇼트된다. 커넥터 헬은 소스 장치에서 접지부로 쇼트된다. 커넥터(2400)(도 24)는 소스 장치의 커넥터와 디스플레이 장치의 커넥터에 대해 모두 동일한 편아웃을 가지는, 한 개의 행에 13개의 핀을 가진다. 소스 장치는 전력을 공급한다.

메인 링크(222)는 두 말단에서 종료된다. 메인 링크(222)가 AC 연결되기 때문에, 0V(접지)부터 +3.6볼트 사이에 말단 전압이 위치할 수 있다. 본 구현에서, 구동 전류는 8mA부터 24mA 까지의 범위로 링크 조건에 따라 프로그래밍 가능하며, 따라서 $V_{\text{differential,pp}}$ 가 400 mV에서 1.2V까지의 범위를 보인다. 트레이닝 패턴을 이용하여 각각의 연결에 대해 최소 전압 스윙이 선택된다. 전력 관리 모드동안 전기 아이들 상태가 제공된다. 전기 아이들 상태에서, 양의 신호도 음의 신호도 구동되지 않는다. 전기 아이들 상태로부터 송신을 개시할 때, 송신기는 수신기와의 링크를 재구축하기 위해 트레이닝 세션을 실행하여야 한다.

상태 다이어그램

발명은 도 25와 도 26에 제시된 상태 다이어그램을 참고하여 이제부터 설명될 것이다. 따라서 도 15는 아래의 소스 상태 다이어그램을 도시한다. 오프 상태(2502)에서, 시스템은 오프로서 소스가 동작하지 않는다. 소스가 동작할 경우, 시스템은 전력 절감 및 수신기 검출에 적합한 스템바이 상태(2504)로 전이한다. 수신기의 존재(즉, 핫플러그/플레이) 여부를 검출하기 위해, 보조 채널이 주기적으로 펄스화되고(가령, 10ms마다 1us동안) 구동 중 말단 저항들 사이의 전압 강하가 측정된다. 측정된 전압 강하를 바탕으로 수신기가 존재한다고 결정될 경우, 시스템은 수신기가 검출되었음(즉, 핫플러그 이벤트가 검출되었음)을 표시하는 수신기 검출 상태(2506)로 전이한다. 그러나 수신기 검출이 없는 경우에, 수신기 검출은 수신기가 검출되거나 시간이 초과될 때까지 계속된다. 일부 경우에 소스 장치는 어떤 추가적인 디스플레이 검출도 시도되지 않는 오프 상태로의 진행을 선택할 수 있다.

상태 2506에서 디스플레이 핫 언플러그 이벤트가 검출되면, 시스템은 스템바이 상태(2504)로 되돌아간다. 그렇지 않을 경우, 소스는 수신기를 깨우기 위해 양의 신호와 음의 신호로 보조 채널을 구동하고, 수신기의 차후 응답을 확인한다. 수신한 응답이 없을 경우, 수신기는 깨어나지 않으며 소스는 상태 2506을 유지한다. 그러나 신호가 디스플레이 장치로부터 수신되면, 디스플레이에는 깨어나게 되고 소스는 수신기 링크 기능(가령, 최대 링크 속도, 버퍼 크기, 시간 베이스 복원 유닛 수)을 판독할 준비를 한다. 그리고 시스템은 메인 링크 초기화 상태(2508)로 전이하고, 트레이닝 시점 통지 단계를 개시할 준비를 한다.

이 때, 트레이닝 세션은 세트 링크 속도로 메인 링크 상태에서 트레이닝 패턴을 전송함으로서 시작되며, 관련 트레이닝 상태를 점검한다. 수신기는 세 단계 각각에 대해 통과/차단(pass/fail) 비트를 설정하고, 송신기는 통과가 검출될 때 메인 링크가 상기 링크 속도로 준비되도록 통과 검출시에 다음 단계로 진행할 것이다. 이 때, 인터페이스는 정상 동작 상태(2210)로 전이하고, 그렇지 않을 경우, 링크 속도는 감소하고 트레이닝 세션이 반복된다. 정상 동작 상태(2510) 중, 소스는 링크 상태 인덱스를 계속하여 주기적으로 모니터링하며, 실패할 경우, 핫 언플러그 이벤트가 검출되고 시스템은 스템바이 상태(2504)로 전이하여 핫 플러그 검출 이벤트를 기다린다. 그러나 동기 손실이 검출될 경우, 시스템은 메인 링크 재-개시 이벤트를 위한 상태 2508로 전이한다.

도 26은 디스플레이 상태 다이어그램(2600)을 도시한다. 상태 2602에서는 전압이 검출되지 않고 디스플레이는 오프 상태이다. 스템바이 모드 상태 2604에서, 메인 링크 수신기와 보조 채널 슬레이브는 전기적으로 아이들 상태이고, 보조 채널 슬레이브 포트의 말단 저항들 간 전압 강하는 지정 전압에 대해 감시된다. 전압이 검출되면 보조 채널 슬레이브 포트가 온 상태로 변화한다. 이는 핫 플러그 이벤트를 표시하는 것으로서 시스템은 디스플레이 상태 2606으로 진행한다. 그렇지 않을 경우 디스플레이는 스템바이 상태 2604에 머무른다. 상태 2606에서(메인 링크 초기화 단계), 디스플레이가 검출되면, 보조 슬레이브 포트는 완전하게 온 상태가 되며, 송신기는 수신기 링크 기능 판독 명령에 응답하고, 디스플레이 상태는 2608로 전이한다. 그렇지 않을 경우 지정 시간 주기보다 긴 시간동안 보조 채널에 활동이 없으면, 보조 채널 슬레이브 포트가 스템바이 상태(2604)로 들어간다.

트레이닝 시작 통지 단계동안, 디스플레이는 트레이닝 패턴들을 이용하여 등화기를 조정함으로서 송신기에 의해 트레이닝 개시에 응답하고, 각각의 단계에 대한 결과를 업데이트한다. 트레이닝이 실패할 경우, 또 다른 트레이닝 세션을 기다리며, 트레이닝이 통과될 경우, 정상 동작 상태(2610)로 진행한다. 보조 채널이나 메인 링크에 지정 시간(가령, 10ms)보다 큰 시간동안 아무 활동이 없을 경우, 보조 채널 슬레이브 포트는 스템바이 상태(2604)로 설정된다.

도 27-31은 크로스 플랫폼 디스플레이 인터페이스의 특정 구현을 도시한다.

도 27은 본 발명에 따라 송신기(2704)를 통합하는 온-보드 그래픽 엔진(2002)을 구비한 PC 마더보드(2000)를 도시한다. 송신기(2704)는 도 1의 송신기(102)의 특정 예에 해당한다. 본 실시예에서, 송신기(2704)는 마더보드(2000) 상에 장착된 커넥터(2706)에 연결되며, 다시 커넥터(2706)는 디스플레이 장치(2708)에 연결된 꼬인 쌍 케이블(2710)을 이용하여 디스플레이 장치(2708)에 연결된다.

당 분야에 잘 알려진 바와 같이, PCI 익스프레스(미국, 캘리포니아 Santa Clara 소재의 Intel 사에 의해 개발됨)는 고대역폭, 로우 편 카운트, 시리얼, 상호연결 기술로서, 기존 PCI 인프러스트럭처와 소프트웨어 호환성을 유지하는 제품이다. 본 구성에서, PCI 익스프레스 포트는 크로스 플랫폼 인터페이스의 요건에 부합하도록 증강되었으며, 마더보드에 장착된 커넥터를 이용하여 디스플레이 장치를 직접 구동할 수 있다.

마더보드에 커넥터를 장착하는 것이 적합하지 않은 경우에, 신호들은 PCI 익스프레스 마더보드의 SDVO 슬롯을 통해 전달될 수 있고, 도 28의 패시브 카드 커넥터를 이용하여 PC의 후방에 연결될 수 있다. 애드-인 그래픽 카드의 현 생성에서처럼, 애드-인 그래픽 카드는 도 30에서처럼 온보드 그래픽 엔진을 지원할 수 있다.

노트북 애플리케이션의 경우에, 마더보드 그래픽 엔진에서의 송신기는 내부 케이블링을 통해 일체형 수신기/TCON을 구동하며, 이는 패널을 직접 구동시킬 것이다. 대부분의 합리적 가격 구현에서, 수신기(TCON)은 패널에 장착되어 도 31에 제시되는 바와 같이 상호연결 와이어의 수를 8개나 10개로 감소시킨다.

모든 예들은 일체형 송신기를 가정한 것이다. 그러나 AGP나 SDVO 슬롯을 통해 PCI 및 PCI 익스프레스 환경에 통합되는 독립형 송신기로 구현하는 것도 역시 가능하다. 독립형 송신기는 그래픽 하드웨어나 소프트웨어에 어떤 변화도 필요없이 출력 스트림들을 발생시킬 것이다.

순서도 실시예

발명의 방법은 발명을 구현하는 특정 처리과정을 설명하는 순서도를 참고하여 설명될 것이다. 도 32-36은 발명의 여러 태양들에 대한 단일한 태양 또는 조합 태양에 사용될 때 다수의 관련된 처리과정들을 설명한다.

도 32는 발명의 한 실시예에 따른 인터페이스(100)의 동작 모드를 결정하는 처리과정(3200)의 순서도이다. 이 과정에서, 동작 모드는 비디오 소스와 디스플레이 장치가 모두 디지털인 경우 디지털 모드로 설정될 것이다. 그렇지 않을 경우 동작 모드는 아날로그 모드로 설정될 것이다. 본 명세서에서 "아날로그 모드"란 기존의 VGA 모드와 개선형 아날로그 모드를 모두 포함할 수 있다. 개선형 아날로그 모드란 내포된 정렬 신호와 양방향 측파대를 갖춘 차동 아날로그 비디오를 가지는 경우를 말한다. 이 개선형 아날로그 모드는 아래에서 설명될 것이다.

단계 3202에서, 비디오 소스가 아날로그 데이터나 디지털 데이터를 지원하는지 여부를 결정하기 위해 비디오 소스에 질의가 이루어진다. 비디오 소스가 아날로그 데이터만을 지원할 경우, 연결 장치(100)의 동작 모드는 아날로그로 설정될 것이고(단계 3208) 처리과정이 종료될 것이다(단계 3212).

비디오 소스가 디지털 데이터를 출력할 수 있다면, 처리과정은 단계 3206으로 진행된다. 디스플레이 장치가 디지털 데이터를 수신하도록 구성되었는지를 결정하기 위해 디스플레이 장치에 질의가 이루어진다. 디스플레이 장치가 아날로그 데이터만을 지원할 경우 연결 장치의 동작 모드는 아날로그로 설정될 것이며(3208), 처리과정이 종료될 것이다(단계 3212). 그렇지 않을 경우, 연결 장치의 동작 모드가 디지털로 설정된다(단계 3210). 예를 들어, 동작 모드를 디지털로 설정하도록 연결 장치 내의 스위치를 프로세서가 제어할 수 있다. 일반적으로, 연결 장치는 비디오 소스와 비디오 싱크 모두가 대응하는 디지털 모드로 동작하는 경우에만 풀 디지털 모드로 동작하도록 설정된다.

도 33은 발명의 일부 태양에 따라 실시간 비디오 이미지 품질 점검을 제공하는 처리과정(3300)의 순서도이다. 본 예에서, 처리과정(3300)의 모든 결정사항은 디스플레이 인터페이스에 연결된 프로세서에 의해 구현된다.

단계 3300에서, 비디오 신호가 비디오 소스로부터 수신된다. 그후, 수신한 비디오 신호에 관련된 비디오 소스로부터 신호 품질 테스트 패턴이 제공된다(단계 3302). 단계 3304에서, 비트 오류율에 대한 결정이 이루어진다. 이는 품질 테스트 패턴을 바탕으로 이루어진다. 그후, 비트오류율이 한도값보다 큰지가 결정된다(단계 3306). 비트 오류율이 한도값 이하라고 결정될 경우, 추가적인 비디오 프레임들이 있는지 여부가 결정된다(단계 3314). 추가적인 비디오 프레임들이 있다고 결정될 경우, 처리과정은 단계 3300으로 진행한다. 그렇지 않을 경우 처리과정이 종료된다.

그러나 비트 오류율이 한도값보다 크다고 결정될 경우(단계 3306), 비트속도가 최소 비트 속도보다 큰지가 결정된다(단계 3308). 비트 속도가 최소 비트 속도보다 클 경우, 비트 속도가 저하되고(단계 3310), 처리 과정은 단계 3306으로 되돌아간다. 비트 속도가 최소 비트 속도보다 크지 않을 경우 동작 모드는 아날로그 모드로 변경되고(단계 3312) 처리과정이 종료된다.

도 34는 발명의 한 실시예에 따른 링크 설정 과정(3400)의 순서도이다. 이 과정(3400)은 핫 플러그 검출 이벤트 통지를 수신함으로서 개시된다(3402). 단계 3404에서, 최대 데이터 속도, 수신기에 포함된 시간 베이스 복원 유닛들의 수, 그리고 사용 범위 크기를 결정하기 위해 관련 보조 채널을 이용하여 메인 링크 질의가 이루어진다. 그후 3406에서, 최대 링크 데이터 속도가 트레이닝 세션에 의해 확인되고, 단계 3408에서, 데이터 스트림 소스에 핫 플러그 이벤트가 통지된다. 단계 3410에서, 보조 채널을 이용하여 디스플레이(가령, EDID 이용)의 기능이 결정되고, 3412에서 디스플레이가 질의에 응답한다. 그 결과 3414에서 메인 링크 트레이닝 세션의 협력처리가 이루어진다.

그후 3416에서, 스트림 소스는 보조 채널을 이용하여 수신기에 스트림 속성들을 전송하고, 3418에서, 요청한 데이터 스트림들의 수를 메인 링크가 지원(3420)할 수 있는지 여부를 스트림 소스에게 통지한다. 단계 3422에서, 관련 패킷 헤더들을 추가함으로서 다양한 데이터 패킷들이 형성되며, 다수의 소스 스트림들에 대한 멀티플렉싱이 단계 3424에서 스케줄링된다. 3426에서, 링크 상태가 양호한지 여부를 결정한다. 링크 상태가 양호하지 않을 경우, 3428 단계에서 링크 실패 이벤트가 소스에게 통지된다. 그렇지 않을 경우, 링크 데이터 스트림이 원본 스트림으로 재구성된다. 이는 다양한 패킷 헤더들을 바탕으로 3430 단계에서 이루어진다. 단계 3432에서, 재구성된 원본 데이터 스트림들이 디스플레이 장치에 전달된다.

도 35는 발명의 한 실시예에 따라 트레이닝 세션을 실행하는 과정(3500)의 순서도를 도시한다. 트레이닝 세션 과정(3500)은 도 32에 설명된 동작(3206)의 한 구현에 지나지 않는다. 트레이닝 세션은 세트 링크 속도로 메인 링크 상에서 트레이닝 패턴을 수신기에 전송함으로서 시작된다(3502). 전형적인 링크 트레이닝 패턴이 도 11에 도시된다. 이는 발명의 한 실시예에 따른다. 도시되는 바와 같이, 트레이닝 세션 중 단계 1은 최단 동작 길이를 나타내고, 단계 2는 최장 동작 길이를 나타낸다. 수신기는 등화기(equalizer)를 최적화시키기 위해 이 두 단계들을 이용한다. 단계 3에서는 링크 품질이 합리적이기만 하다면 비트 랙과 문자 랙이 모두 구현된다. 3504에서 수신기는 관련 트레이닝 상태를 점검하고, 트레이닝 상태 점검을 바탕으로, 수신기는 세 단계 각각과 송신기에 대한 통과/차단 비트를 설정한다(3506). 각각의 단계에서, 수신기는 통과 검출시에만 다음 단계로 진행할 것이며, 수신기가 통과를 검출하지 못할 경우, 수신기는 링크 속도를 감소시키고(단계 3510) 트레이닝 세션을 반복할 것이다. 메인 링크는 통과가 검출된 링크 속도에서 동작할 준비가 되어 있다(단계 3512).

도 36은 발명의 구현에 사용되는 컴퓨터 시스템(3600)을 도시한다. 컴퓨터 시스템(3600)은 발명의 구현될 수 있는 그래픽 시스템의 한 예에 지나지 않는다. 컴퓨터 시스템(3600)은 중앙 처리 장치(CPU)(3610), RAM(3620), ROM(3525), 한 개 이상의 주변 장치(3630), 그래픽 컨트롤러(3660), 주기억 장치(3640, 3650), 디지털 디스플레이 유닛(3670)을 포함한다. 당분야에 잘 알려진 바와 같이, ROM은 데이터와 명령을 CPU(3610)에 일방향으로 전송하는 기능을 하며, RAM은 데이터와 명령을 양방향으로 전송하는 데 사용된다. CPU(3610)는 다수의 프로세서들을 포함할 수 있다. 주기억 장치(3640, 3650)는 적절한 컴퓨터-판독형 매체를 포함할 수 있다. 보조기억 매체(880)는 대용량 기억 장치로서, 역시 CPU(3610)에 양방향으로 연결되어 추가적인 데이터 기억 용량을 제공한다. 대량 기억 장치(880)는 컴퓨터 코드, 데이터 등을 포함한 프로그램들을 저장하는 데 사용될 수 있는 컴퓨터-판독형 매체이다. 일반적으로, 대량 기억 장치(880)는 하드디스크나 테이프와 같은 기억 매체로서, 주기억 장치(3640, 3650) 보다 느린다. 대량 기억 장치(880)는 자기/종이 테이프 리더나 그 외 다른 잘 알려진 장치의 형태를 취할 수 있다. 대량 기억 장치(880) 내에 보유된 정보는 적절한 경우에 RAM(3620)의 일부분으로 표준 방식에 따라 가상 메모리로 통합될 수 있다.

CPU(3610)는 한 개 이상의 입/출력 장치(890)에 연결되며, 상기 입/출력 장치(890)는 비디오 모니터, 트랙볼, 마우스, 키보드, 마이크, 터치식 디스플레이, 트랜스듀서 카드 리더, 자기/종이 테이프 리더, 태블릿, 스타일러스, 음성/수기 인식기, 또는 그 외 다른 잘 알려진 입력 장치 등을 포함할 수 있으나 이에 제한되지는 않는다. 마지막으로 CPU(3610)는 컴퓨터나 통신망에 연결될 수 있다. 가령, 인터넷이나 인트라넷에 네트워크 연결(3695)을 이용하여 연결될 수 있다. 이러한 네트워크 연결을 이용하여, 상술한 방법 단계들을 수행하는 과정에서 CPU(3610)는 네트워크로부터 정보를 수신할 수 있고, 네트워크에 정보를 출력할 수도 있다. 이러한 정보는 CPU(3610)를 이용하여 실행될 수 있는 명령들의 시퀀스로 나타나는 경우가 있는데, 반송파로 구현되는 컴퓨터 데이터 신호 형태로 네트워크로부터 수신되거나 네트워크에 출력될 수 있다. 상술한 장치 및 재료들은 컴퓨터 하드웨어 및 소프트웨어 분야의 통상의 지식을 가진 자에게 친숙한 것들이다.

그래픽 컨트롤러(3660)는 아날로그 이미지 데이터 및 이에 대응하는 기준 신호를 발생시키고, 디지털 디스플레이 유닛(3670)에 이 모두를 제공한다. 아날로그 이미지 데이터는 외부 인코드로부터 또는 CPU(3610)로부터 수신한 화소 데이터를 바탕으로 발생될 수 있다(일례에 불과함). 한 실시예에서, 아날로그 이미지 데이터는 RGB 포맷으로 제공되며 기준 신호는 당분야에 잘 알려진 VSYNC와 HSYNC 신호를 포함한다. 그러나, 아날로그 이미지, 데이터, 기준 신호를 다른 포맷으로 구현할 수도 있다. 예를 들어, 아날로그 이미지 데이터는 대응하는 시간 기준 신호를 가진 비디오 신호 데이터를 포함할 수 있다.

발명의 효과

스케줄러 시간 분할은 다중 스트림들의 패킷을 대응하는 링크 데이터 스트림으로 멀티플렉싱(송신기측)하고 디멀티플렉싱(수신기측)한다. 본 실시예에서, 송신기는 최대 데이터 링크 속도, 사용 범위 크기, 그리고 시간 베이스 복원(TBR) 유닛의 수 등을 이용하여 수신기의 능력을 판독한다. 이 지식으로, 송신기는 가장 효율적인 트랜스포트 구성을 결정할 수 있고, 수신기에 주가적인 질의를 전송할 필요없이 수신기에 의해 차후 데이터 스트림이 수용될 수 있는지 여부를 결정할 수 있다. 데이터 스트림 트랜스포트를 시작하기 전에, 송신기는 비디오 데이터의 경우에 각각의 데이터 스트림에 관련된 칼라 포맷 및 깊이, 형태, 그리고 패킷 크기 같은 스트림 속성을 수신기에 알린다. 속성 데이터를 통신함으로서, 패킷 헤더의 크기는 단일 스트림 ID만이 요구되는 포인트까지 감소될 수 있다. 이 방식으로, 패킷 헤더들이 크기 때문에 보다 큰 오버헤드를 요구하는 기존 패킷 기반 트랜스포트 프로토콜에서 제공되는 트랜스포트 효율에 비해 이 경우의 트랜스포트 효율이 훨씬 증가한다.

(57) 청구의 범위

청구항 1.

멀티미디어 소스 장치를 멀티미디어 싱크 장치에 연결하도록 배열되는 패킷 기반 디스플레이 인터페이스로서, 상기 인터페이스는,

- 소스 장치에 연결되어 원본 스트림 속도에 따라 소스 패킷 데이터 스트림을 수신하는 송신기 유닛,
- 싱크 장치에 연결되는 수신기 유닛,
- 송신기 유닛과 수신기 유닛을 연결하여, 송신기 유닛과 수신기 유닛간의 원본 스트림 속도에 독립적인 링크 속도에 따라 소스 패킷 데이터 스트림을 바탕으로 다수의 멀티미디어 데이터 패킷들로 구성되는 멀티미디어 데이터 패킷 스트림을 이송하는 링크 유닛, 그리고

- 상기 링크 유닛에 연결되어, 데이터 스트림들 중 선택된 데이터 스트림들에 대해 링크 유닛에서 이송을 위한 데이터 패킷들을 스케줄링하는 데이터 패킷 스케줄러 유닛

을 포함하며, 이때, 각각의 데이터 스트림에 대한 데이터 패킷 크기는 데이터 스트림 비트 속도와 링크 비트 속도 간의 비에 따라 좌우되는 연계된 고정 크기인 것을 특징으로 하는 패킷 기반 디스플레이 인터페이스.

청구항 2.

제 1 항에 있어서, 상기 멀티미디어 데이터 패킷 스트림은 다수의 멀티미디어 데이터 패킷 스트림들 중 하나이고, 각각의 멀티미디어 데이터 패킷 스트림은 원본 스트림 속도에 독립적인, 연계된 조정가능형 데이터 스트림 링크 속도를 가지는 것을 특징으로 하는 패킷 기반 디스플레이 인터페이스.

청구항 3.

제 1 항에 있어서, 상기 링크 유닛은,

- 송신기 유닛으로부터 수신기 유닛까지 멀티미디어 데이터 패킷들을 운반하는 일방향 메인 링크, 그리고
- 송신기 유닛과 수신기 유닛 사이에서 양방향으로 정보를 전달하는 양방향 보조 채널

을 추가로 포함하는 것을 특징으로 하는 패킷 기반 디스플레이 인터페이스.

청구항 4.

제 3 항에 있어서, 상기 양방향 보조 채널은 싱크 장치로부터 소스 장치까지 정보를 운반하도록 구성된 일방향 백 채널과, 상기 백 채널과 조율하여, 소스 장치로부터 싱크 장치까지 정보를 운반하도록 메인 채널의 일부분으로 포함되는 일방향 포워드 채널로 형성되는 것을 특징으로 하는 패킷 기반 디스플레이 인터페이스.

청구항 5.

제 2 항에 있어서, 상기 메인 링크 유닛은,

- 멀티미디어 데이터 패킷 스트림들 중 특정한 하나에 각기 연계된 다수의 가상 링크들
- 을 추가로 포함하며, 이때, 각각의 가상 링크는 연계된 가상 링크 대역폭과 가상 링크 속도를 가지는 것을 특징으로 하는 패킷 기반 디스플레이 인터페이스.

청구항 6.

제 5 항에 있어서, 메인 링크 대역폭은 가상 링크 대역폭의 총합보다 크거나 같은 것을 특징으로 하는 패킷 기반 디스플레이 인터페이스.

청구항 7.

멀티미디어 소스 장치를 멀티미디어 싱크 장치에 연결하는 방법으로서, 이 방법은,

- 송신기 유닛이 연결된 소스 장치를 제공하고,
- 수신기 유닛이 연결된 싱크 장치를 제공하며,
- 원본 스트림 속도에 따라 소스 데이터 스트림을 송신기 유닛에 의해 수신하고,
- 링크 유닛을 이용하여 송신기 유닛과 수신기 유닛을 연결하며,
- 다수의 멀티미디어 데이터 패킷들로 구성되는 멀티미디어 데이터 패킷 스트림을 형성하고, 그리고

- 송신기 유닛과 수신기 유닛 간에 링크 속도에 따라 멀티미디어 데이터 패킷 스트림을 이송하는 이송 스케줄을 빌드시키며, 이때, 멀티미디어 데이터 패킷들이 링크 속도와 데이터 스트림 비트 속도에 따라 각기 고정 크기를 가지는

단계들을 포함하는 것을 특징으로 하는 멀티미디어 소스 장치를 멀티미디어 싱크 장치에 연결하는 방법.

청구항 8.

제 7 항에 있어서,

- 송신기 유닛으로부터 수신기 유닛까지 멀티미디어 데이터 패킷들을 운반하도록 배열되는 일방향 메인 링크를 제공하고, 그리고

- 송신기 유닛과 수신기 유닛 사이에서 정보를 양방향으로 이송하도록 배열되는 양방향 보조 채널을 제공하는

단계들을 추가로 포함하는 것을 특징으로 하는 멀티미디어 소스 장치를 멀티미디어 싱크 장치에 연결하는 방법.

청구항 9.

제 8 항에 있어서, 상기 양방향 보조 채널은 싱크 장치로부터 소스 장치까지 정보를 운반하도록 설정된 일방향 백 채널과, 상기 백채널과 조화하여, 소스 장치로부터 싱크 장치까지 정보를 운반하기 위해 메인 채널의 일부분으로 포함되는 일방향 포워드 채널을 포함하는 것을 특징으로 하는 멀티미디어 소스 장치를 멀티미디어 싱크 장치에 연결하는 방법.

청구항 10.

제 9 항에 있어서, 상기 메인 링크 유닛은,

- 멀티미디어 데이터 패킷 스트림들 중 특정한 하나에 각기 연계된 다수의 가상 링크들

을 추가로 포함하며, 이때, 각각의 가상 링크는 연계된 가상 링크 대역폭과 가상 링크 속도를 가지는 것을 특징으로 하는 멀티미디어 소스 장치를 멀티미디어 싱크 장치에 연결하는 방법.

청구항 11.

제 10 항에 있어서, 상기 메인 링크 대역폭이 가상 링크 대역폭들의 총합보다 크거나 같은 것을 특징으로 하는 멀티미디어 소스 장치를 멀티미디어 싱크 장치에 연결하는 방법.

청구항 12.

데이터 소스와 데이터 싱크 간에 데이터 링크를 통한 다수의 데이터 패킷들의 이송을 스케줄링하는 방법으로서, 이 방법은,

- 데이터 패킷 소스로부터 데이터 패킷 싱크까지 데이터 패킷 속성들을 전송하고,

- 상기 소스로부터 상기 싱크까지 전송될 다수의 데이터 스트림 각각에 대해 스트림 비트 속도를 데이터 링크 비트 속도와 비교하며,

- 상기 비교를 바탕으로 데이터 스트림 각각에 대해 패킷 크기를 설정하고, 이때, 상기 패킷 크기가 고정 패킷 크기이며,

- 각각의 데이터 패킷 중 한 개 이상을 조합하며, 그리고

- 조합된 데이터 패킷들을 상기 소스로부터 상기 싱크까지 이송하는

단계들을 포함하는 것을 특징으로 하는 데이터 소스와 데이터 싱크 간에 데이터 링크를 통한 다수의 데이터 패킷들의 이송을 스케줄링하는 방법.

청구항 13.

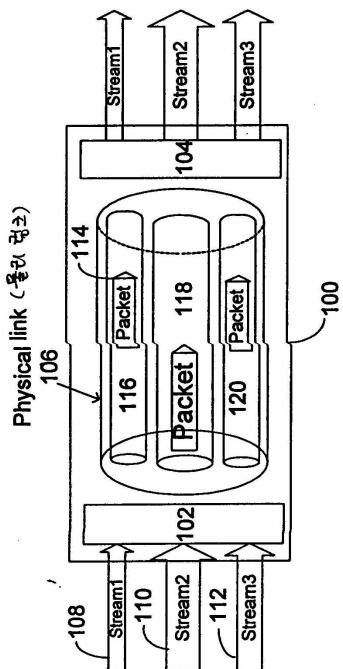
데이터 소스와 데이터 싱크 간에 데이터 링크를 통한 다수의 데이터 패킷들의 이송을 스케줄링하는 컴퓨터 프로그램 프로덕트로서, 이 프로덕트는,

- 데이터 패킷 소스로부터 데이터 패킷 싱크까지 데이터 패킷 속성들을 전송하는 컴퓨터 코드,
- 상기 소스로부터 상기 싱크까지 전송될 다수의 데이터 스트림 각각에 대해 스트림 비트 속도를 데이터 링크 비트 속도와 비교하는 컴퓨터 코드,
- 상기 비교를 바탕으로 데이터 스트림 각각에 대해 패킷 크기를 설정하는 컴퓨터 코드로서, 이때, 상기 패킷 크기가 고정 패킷 크기인, 이러한 컴퓨터 코드
- 각각의 데이터 패킷 중 한 개 이상을 조합하는 컴퓨터 코드, 그리고
- 조합된 데이터 패킷들을 상기 소스로부터 상기 싱크까지 이송하는 컴퓨터 코드

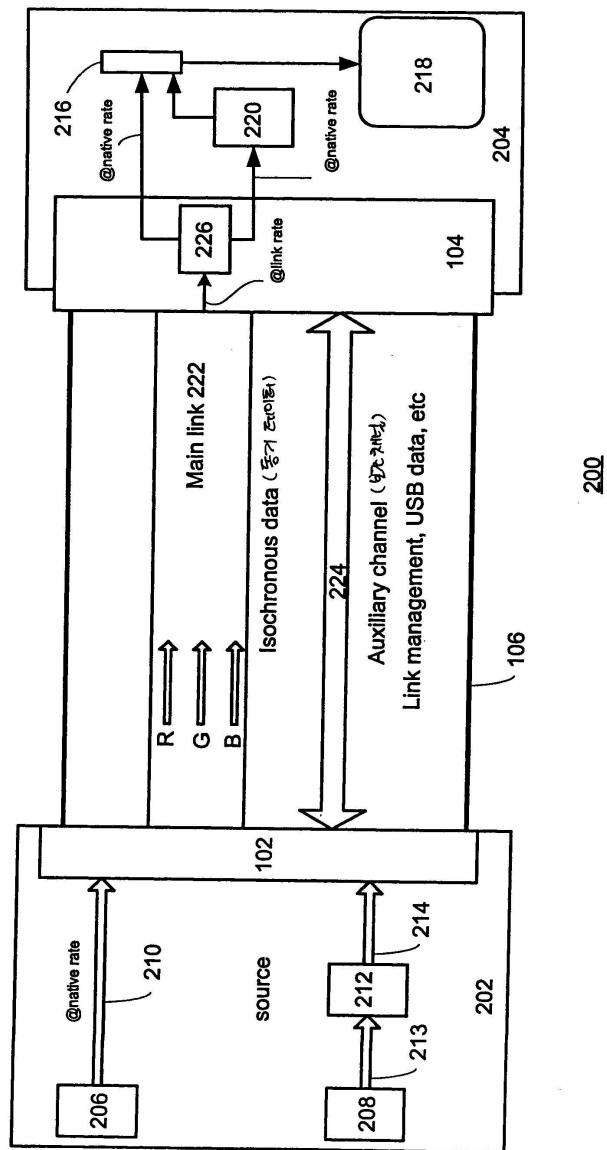
를 포함하는 것을 특징으로 하는 데이터 소스와 데이터 싱크 간에 데이터 링크를 통한 다수의 데이터 패킷들의 이송을 스케줄링하는 컴퓨터 프로그램 프로덕트.

도면

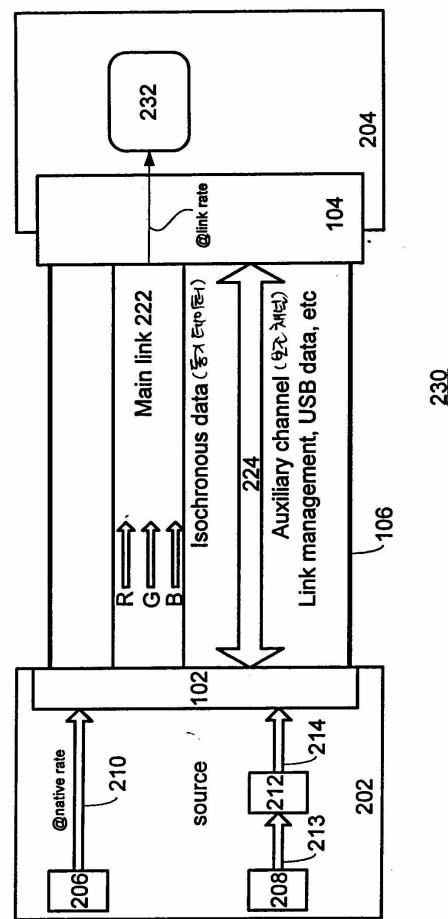
도면1



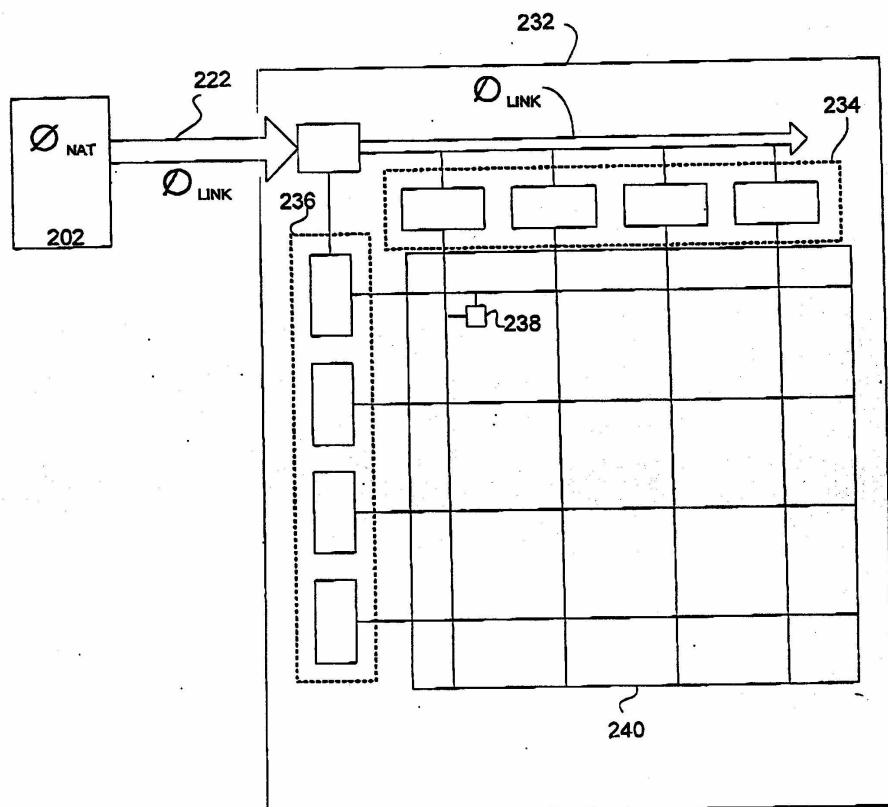
도면2a



도면2b



도면2c

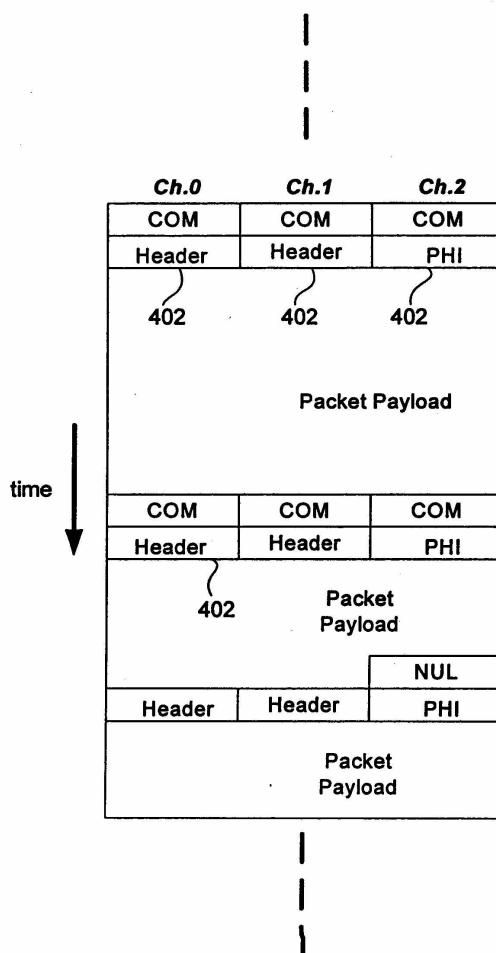


도면3

대역 평균 데이터 속도

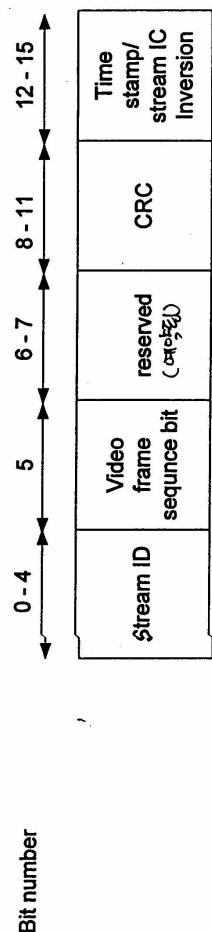
채널당 데이터 속도 (Gbits/second)	채널당 실제 속도 (Gbits/second)	24채널 결합의 금속 송신 인수
1.0	0.960	x40
1.35	1.344	x56
1.7	1.728	x72
2.1	2.112	x88
2.5	2.496	x104

도면4a

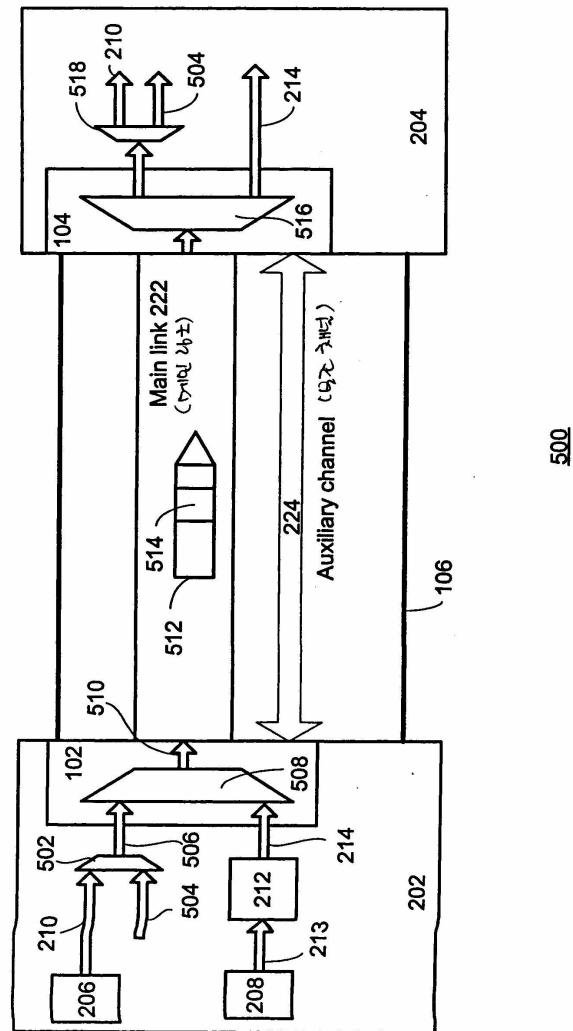


Main Link Packet Format (대역 평균 데이터 속도)

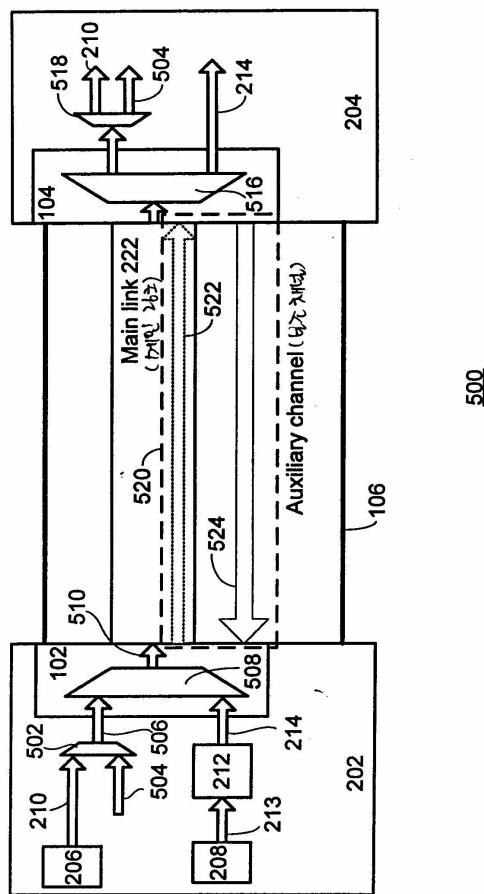
도면4b



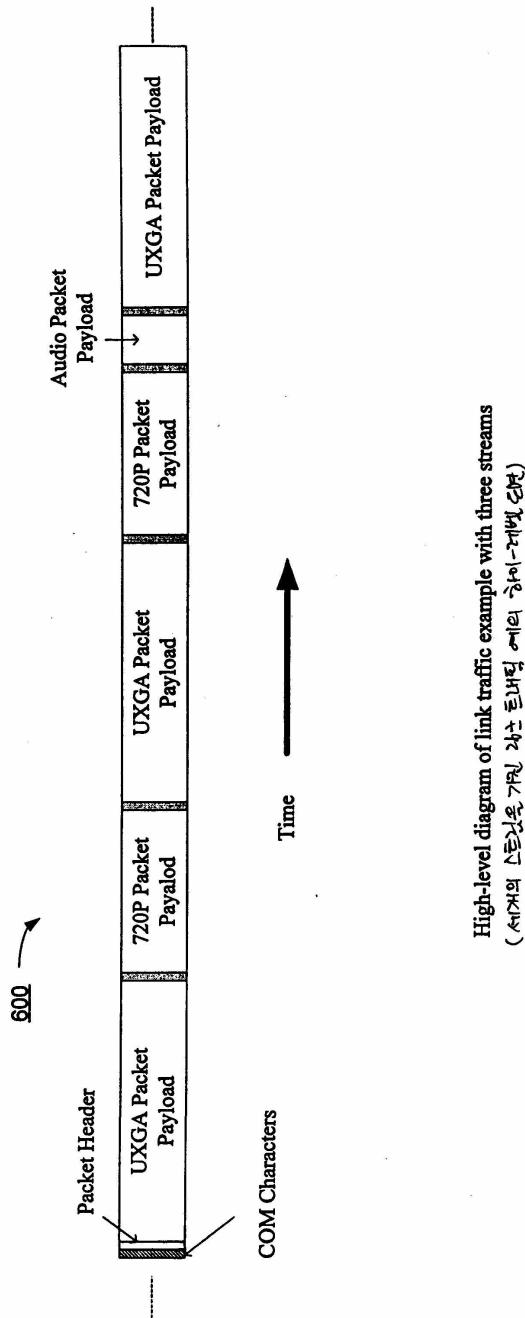
도면5a



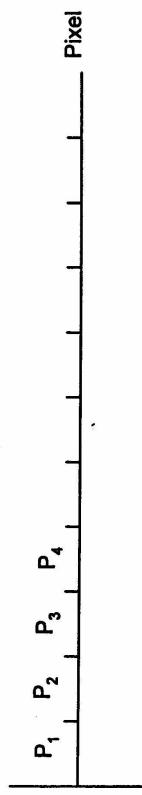
도면5b



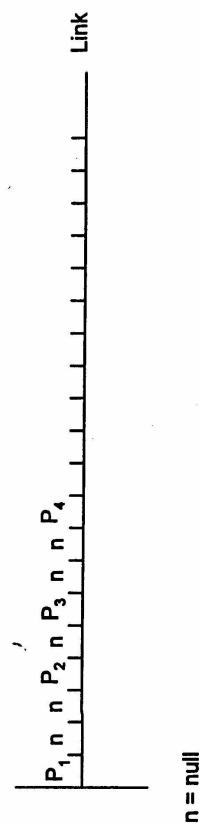
도면6



도면7



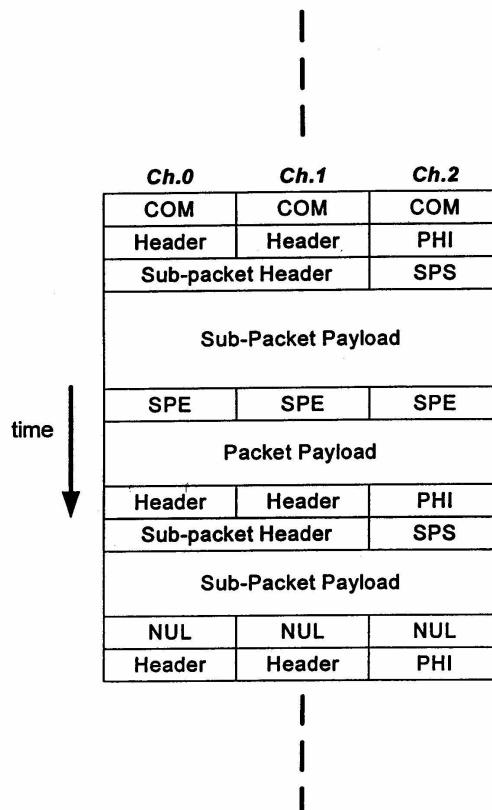
도면8



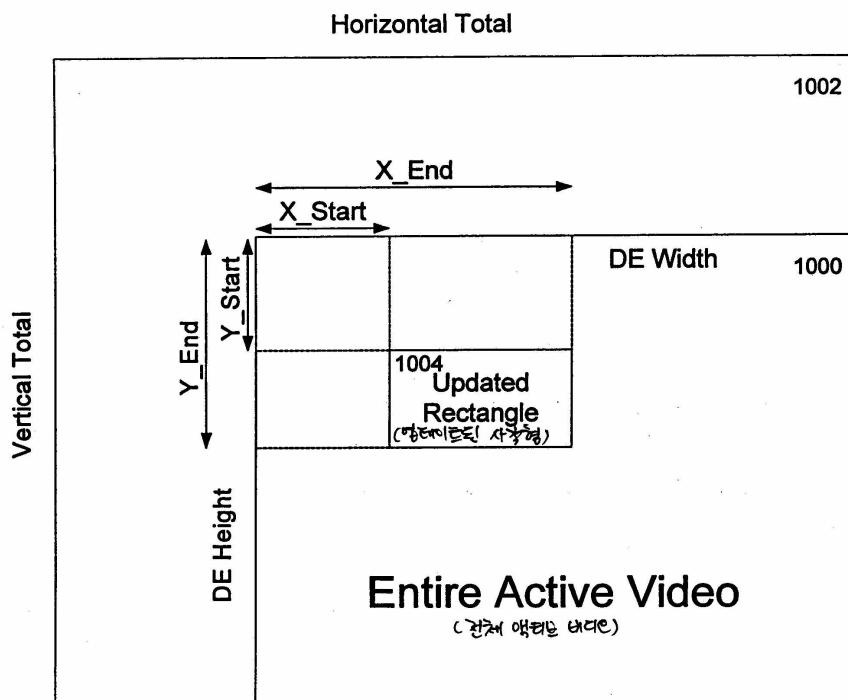
도면9a

SID = 1	0	0	0	CRC	TSP19-16	PHI	
Sub-packet Header	902	Sub-packet Header	902	SPS	904		
Sub-packet Payload							
SID = 1	1	0	0	CRC	TSP33-0	PHI	
Packet Payload for SID 1							

도면9b



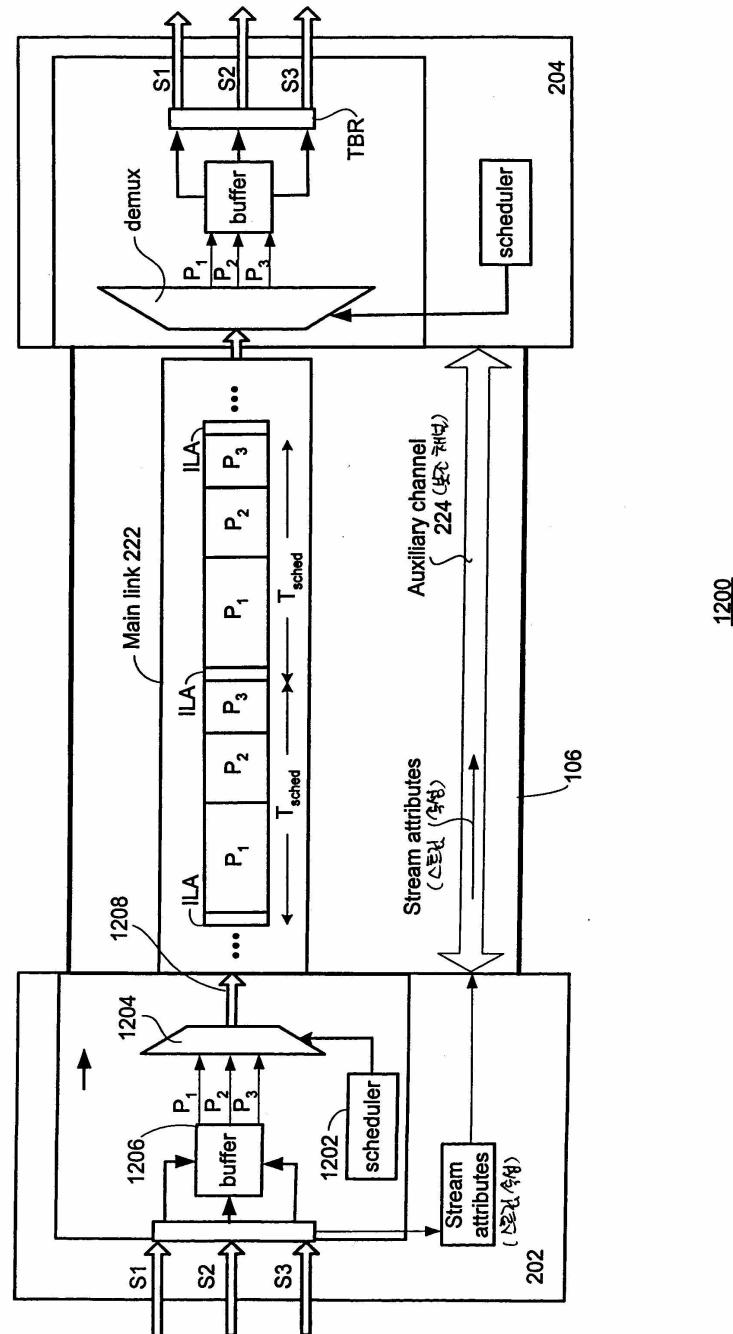
도면10



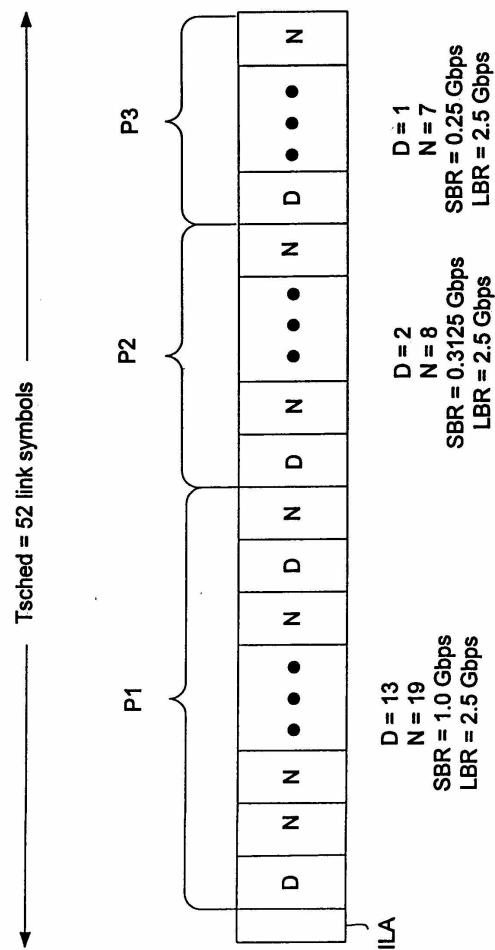
도면11

위상	동상화 결과 문자	0101 패턴
1	D10.2	0101010101 0101010101 0101010101 0101010101 0101010101
2	K28.7	0011111000 0011111000 0011111000 0011111000 0011111000
3	K28.5, and three D10.2	0011111010 0101010101 0101010101 0101010101 1100000101

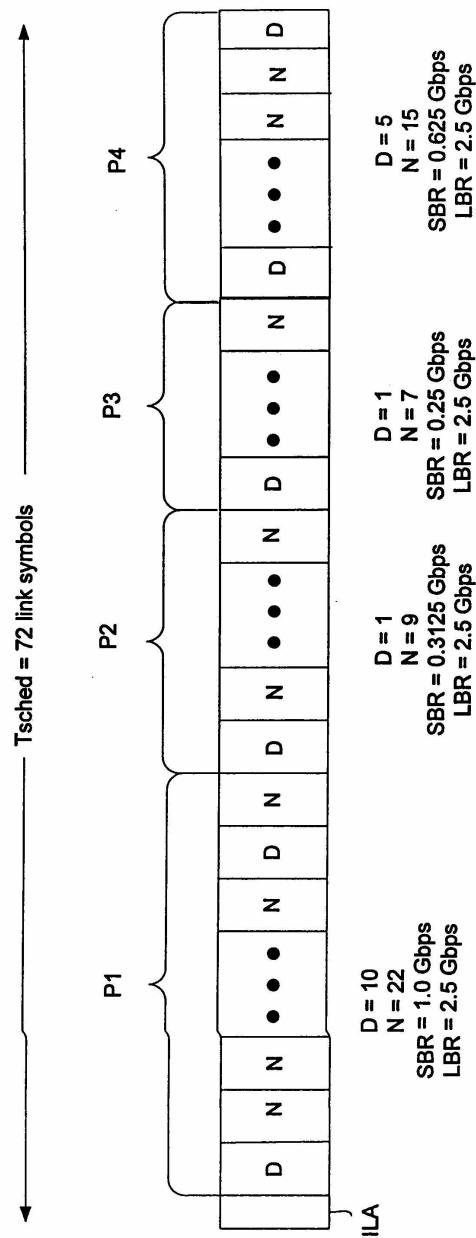
도면12



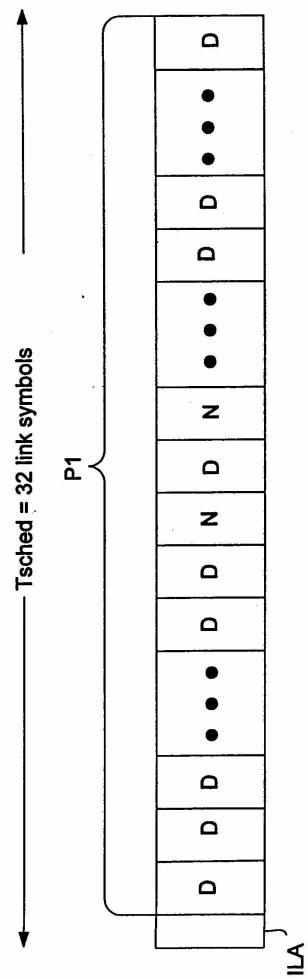
도면13



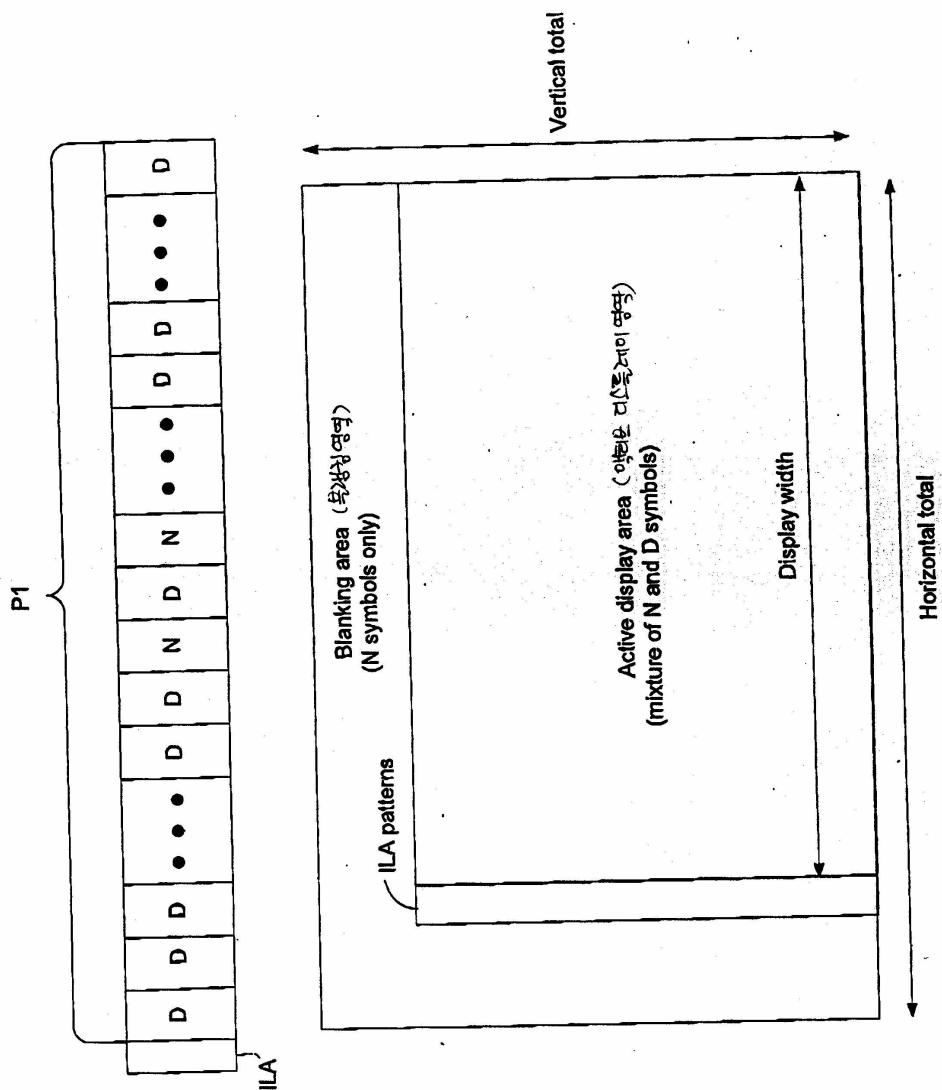
도면14



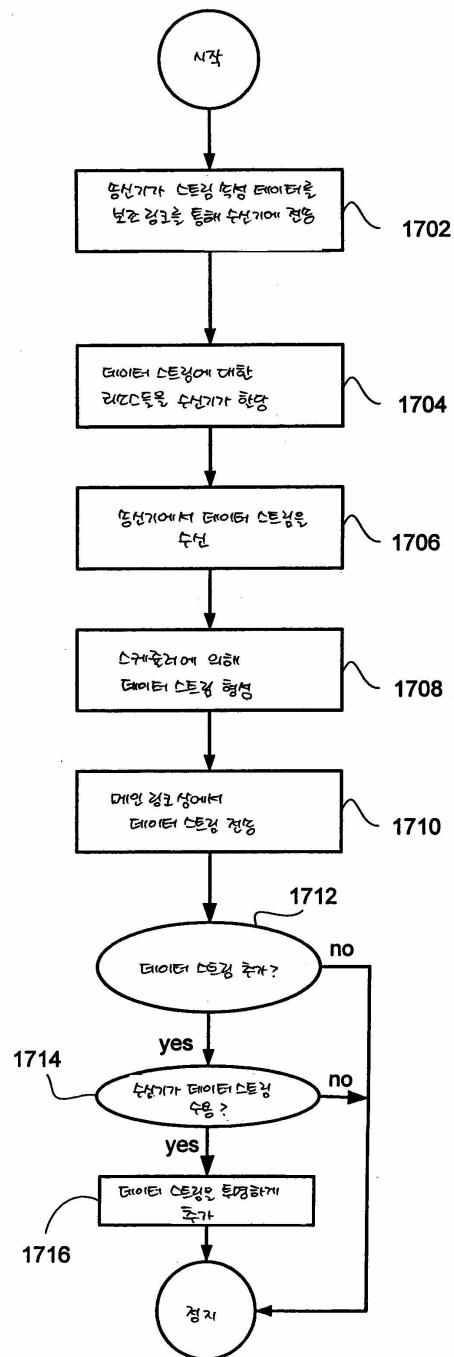
도면15



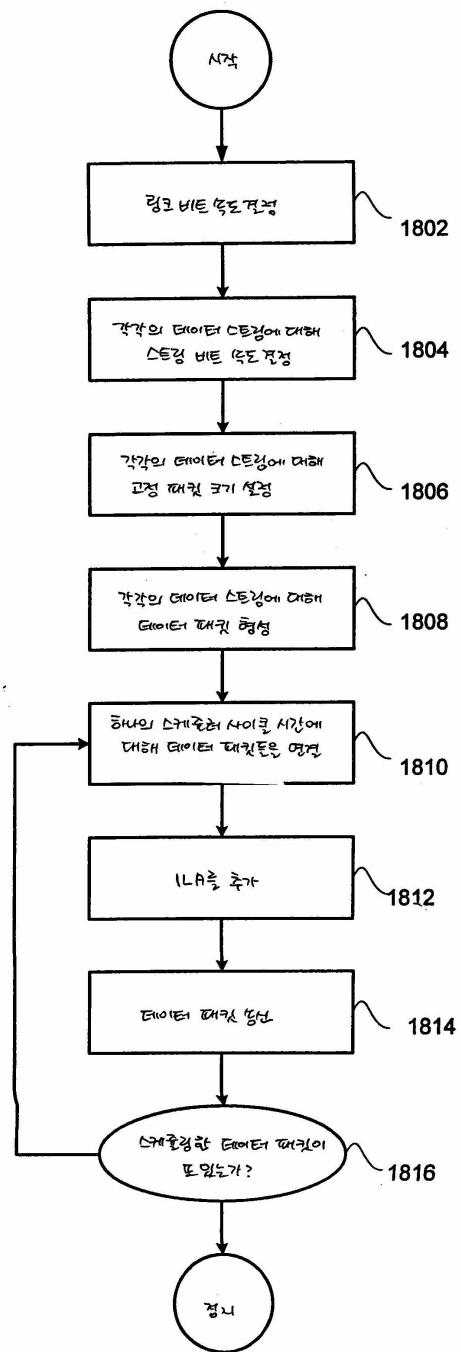
도면16



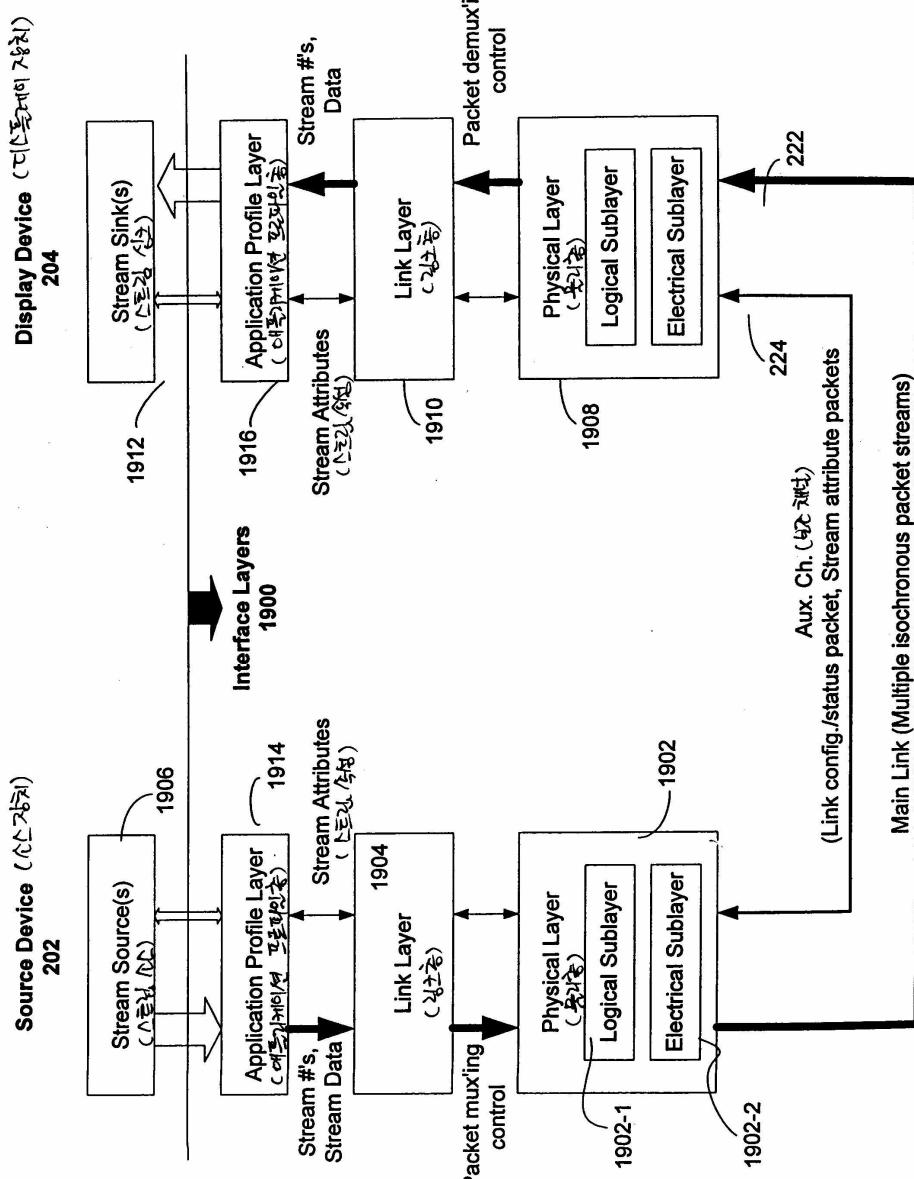
도면17



도면18



도면19

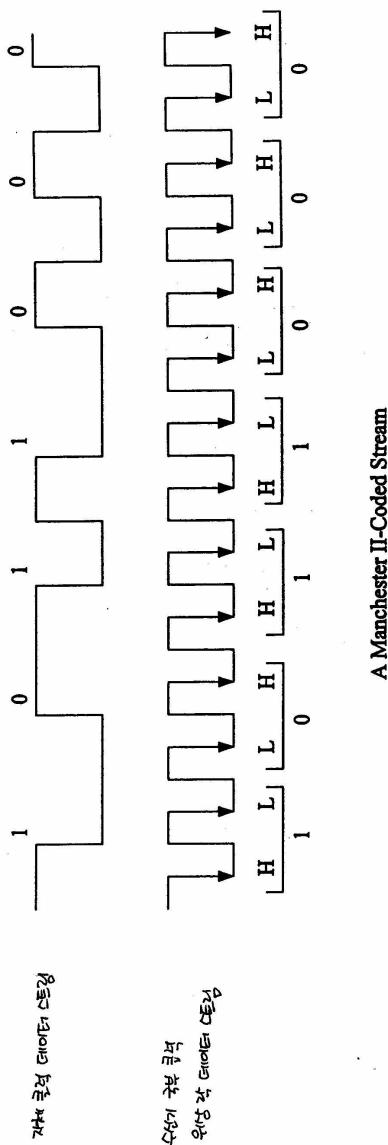


도면20

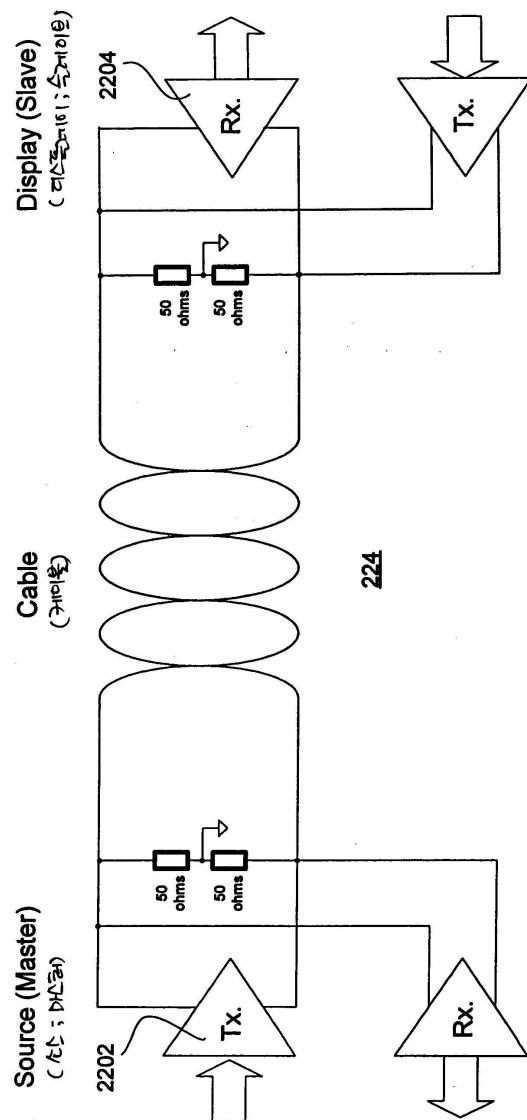
8B/10B 훈련 패턴

도면번호	설명	설명
K28.5	쉼표 Comma (COMI)	쉼표 쉼표가 있는 데이터 패턴
K28.7	TrainingPattern (TPN)	비트/비트로 구성된 훈련 패턴
K23.7	Null (NUL)	전송되는 데이터 없이 전송되는 데이터
K28.2	Sub-packet Start (SPS)	전송되는 데이터 의 시작 지점
K29.7	Sub-packet End (SPE)	전송되는 데이터 의 끝
K28.0	PacketHeaderIndicator (PHI)	전송되는 데이터 의 헤더 지점
K28.1		전송되는 데이터 의 끝
K28.3		전송되는 데이터 의 끝
K28.4		전송되는 데이터 의 끝
K28.6		전송되는 데이터 의 끝
K25.7		전송되는 데이터 의 끝
K27.7		전송되는 데이터 의 끝

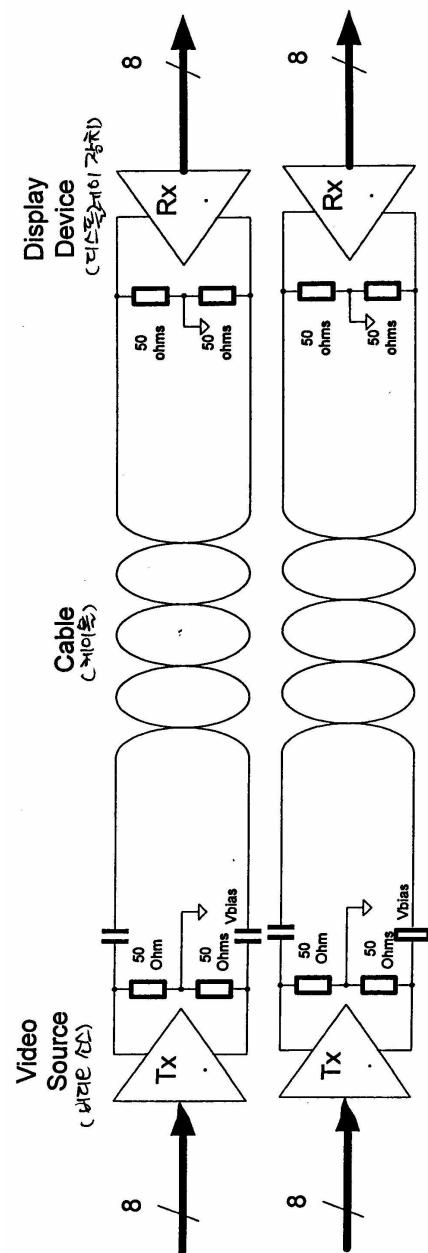
도면21



도면22



도면23

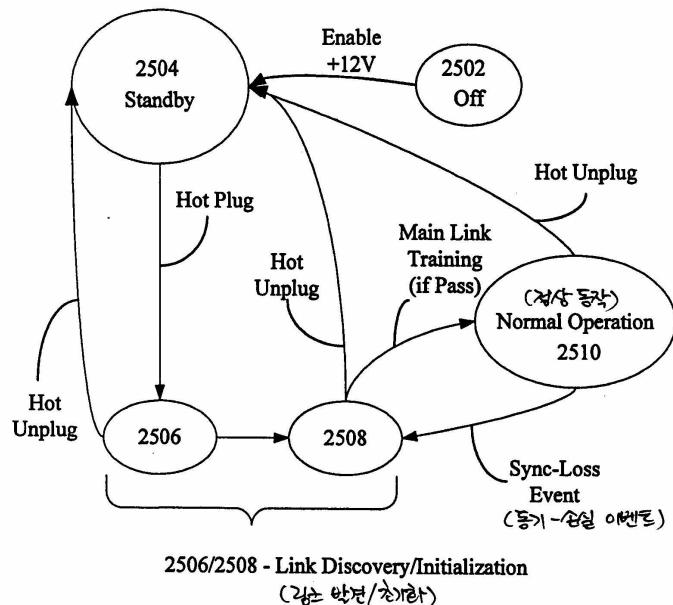


222

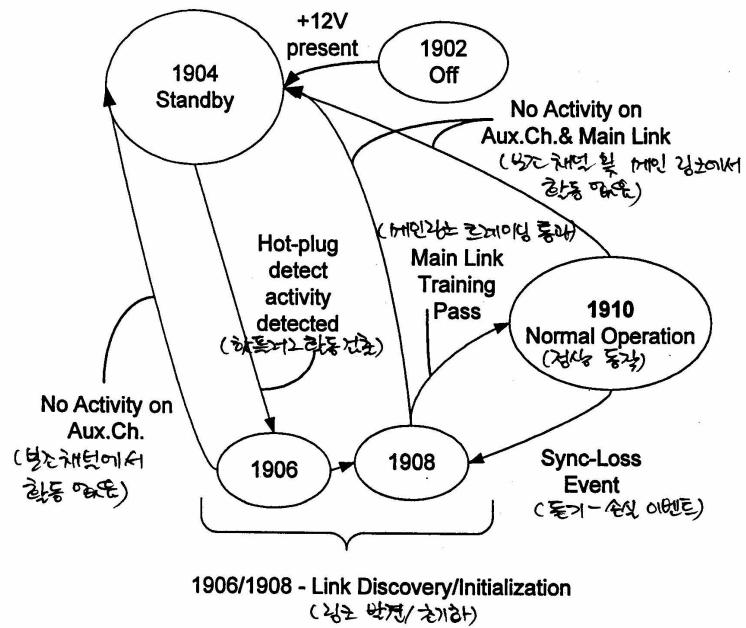
도면24

Connector Shell	
1	<input type="text"/>
2	<input type="text"/>
3	<input type="text"/>
4	<input type="text"/>
5	<input type="text"/>
6	<input type="text"/>
7	<input type="text"/>
8	<input type="text"/>
9	<input type="text"/>
10	<input type="text"/>
11	<input type="text"/>
12	<input type="text"/>
13	<input type="text"/>
2400	
	Main Link Ch0+
	Main Link Ch0-
	Ground
	Main Link Ch1+
	Main Link Ch1-
	Ground
	Main Link Ch2+
	Main Link Ch2-
	Ground
	Aux.Ch.+
	Aux.Ch.-
	Ground
	+12V Power

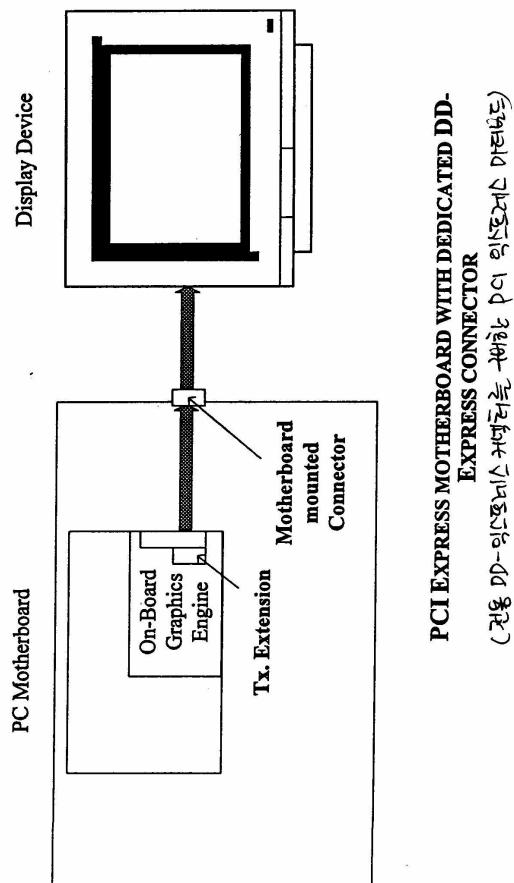
도면25



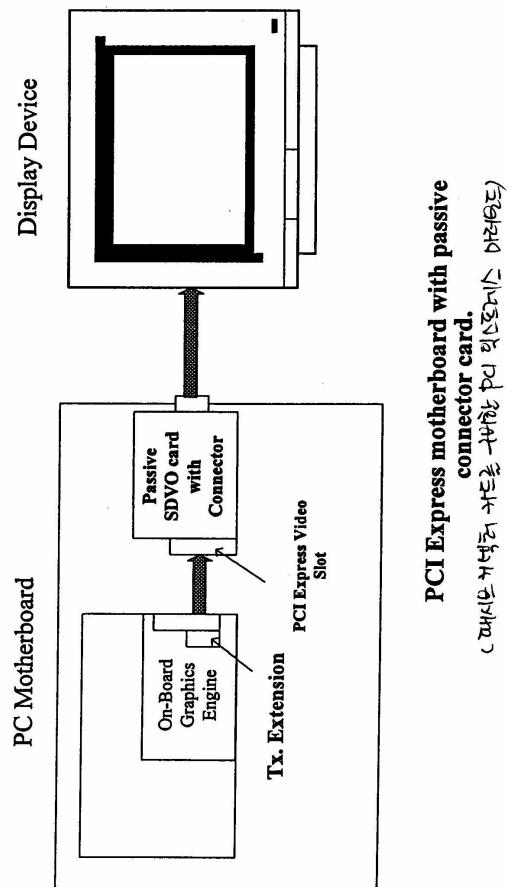
도면26



도면27



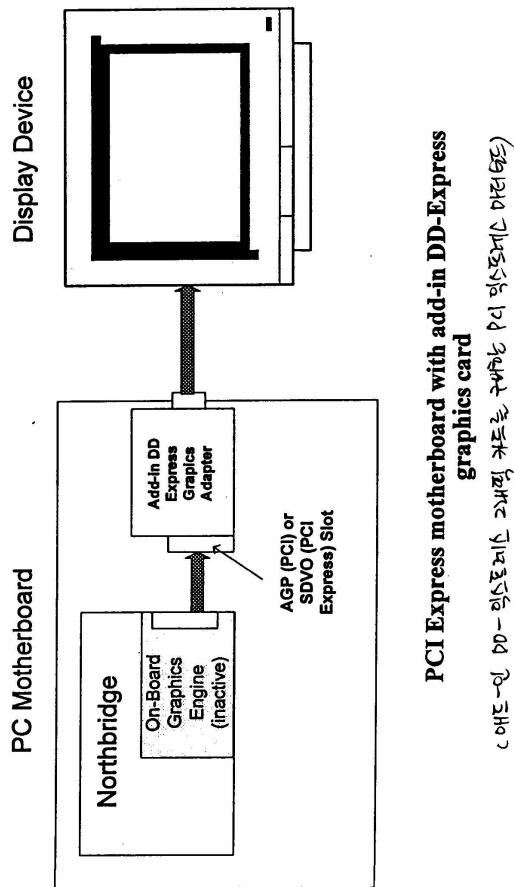
도면28



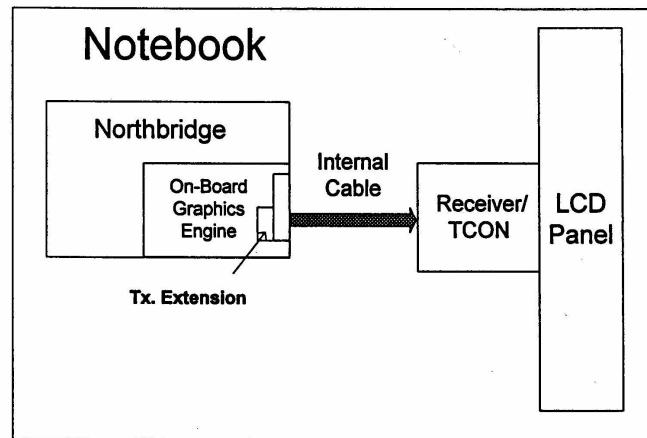
PCI Express motherboard with passive connector card.

(PC Motherboard with Passive Connector Card)

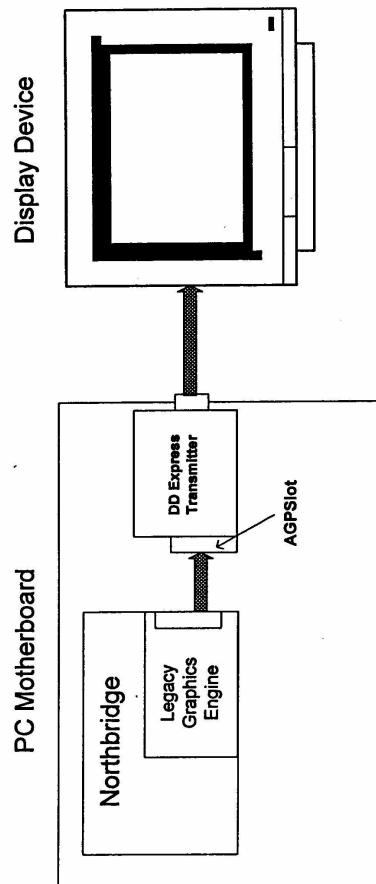
도면29



도면30

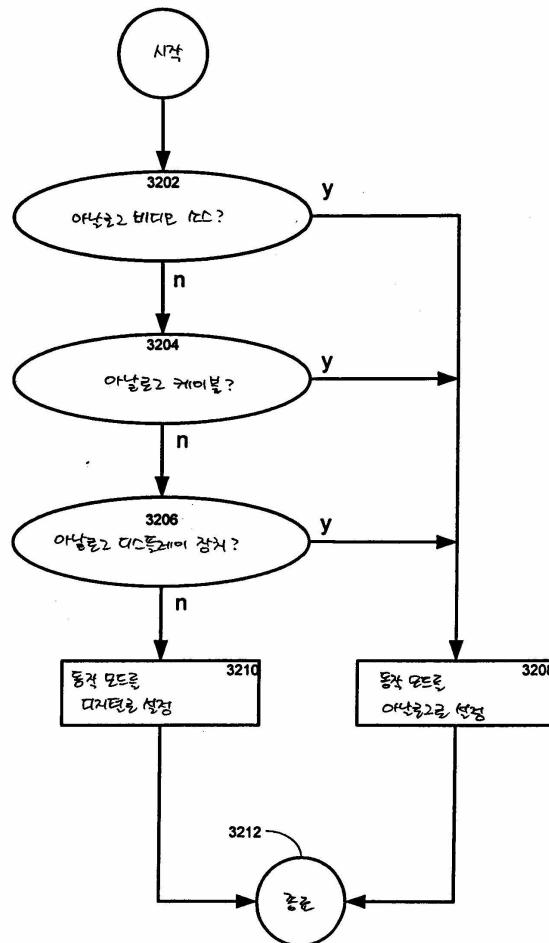


도면31

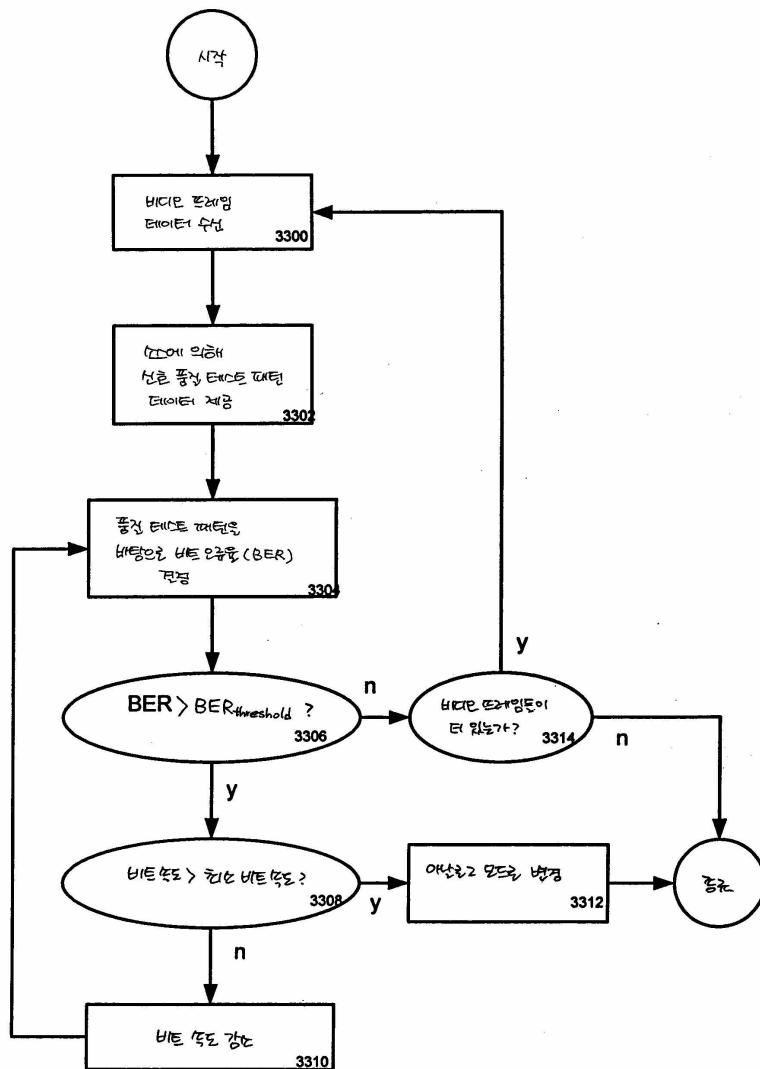


Legacy graphics accelerator bus transmitter mounted on a legacy bus card slot converts digital raster data/timing signals into main link streams

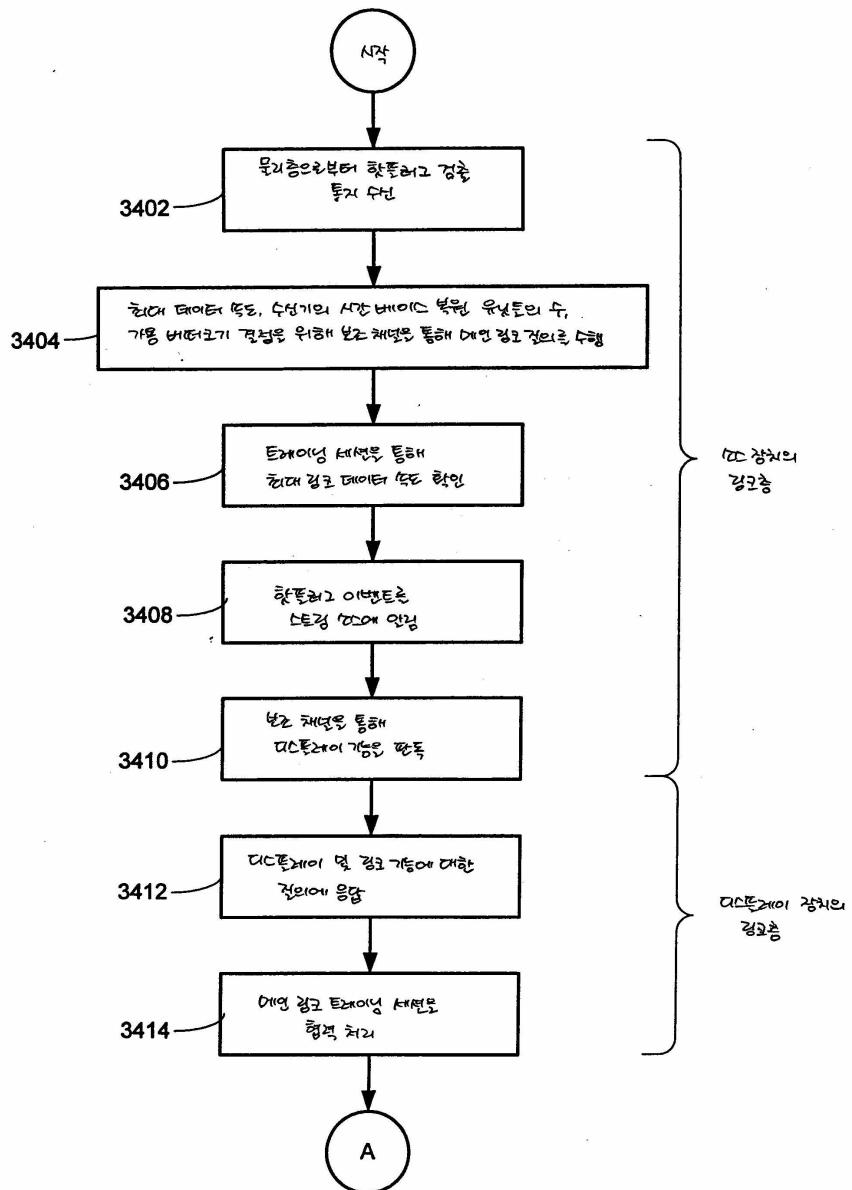
도면32



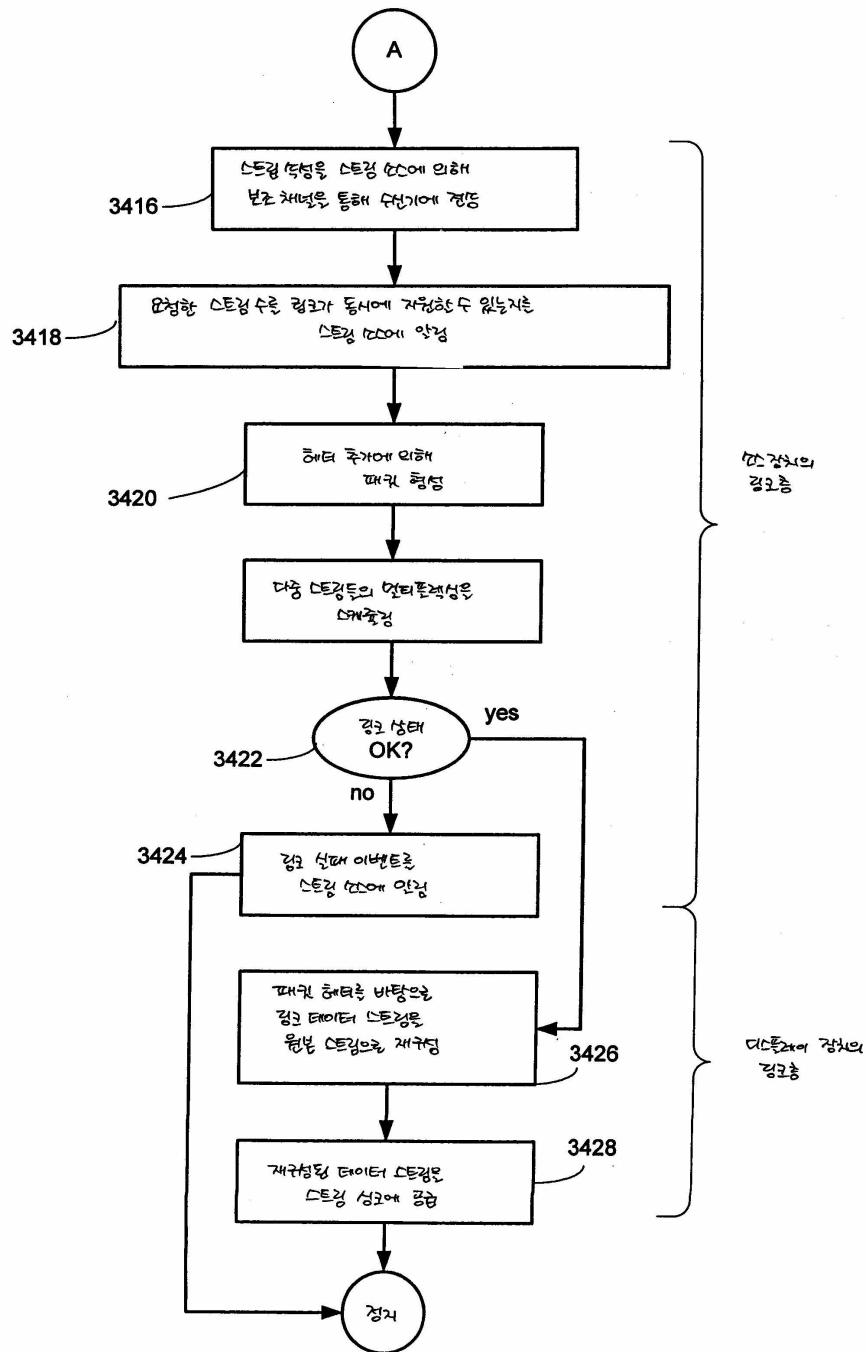
도면33



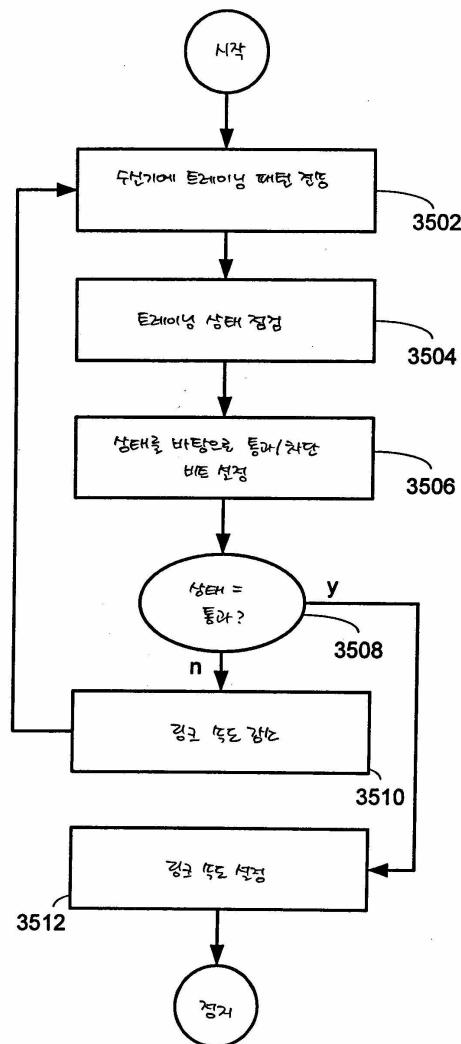
도면34a



도면34b



도면35



도면36

