

(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.

H04L 12/28 (2006.01)

H04L 12/56 (2006.01)

H04L 12/64 (2006.01)

(11) 공개번호 10-2006-0108709

(43) 공개일자 2006년10월18일

(21) 출원번호 10-2006-7011429

(22) 출원일자 2006년06월09일

번역문 제출일자 2006년06월09일

(86) 국제출원번호 PCT/US2004/038002

(87) 국제공개번호 WO 2005/048562

국제출원일자 2004년11월12일

국제공개일자 2005년05월26일

(30) 우선권주장 60/519,833 2003년11월12일 미국(US)

(71) 출원인 칼컴 인코포레이티드
미국 캘리포니아 샌디에고 모어하우스 드라이브5775 (우 92121-1714)(72) 발명자 앤더슨, 존 제임스
미국 80301 콜로라도 보울더 아우구스타 드라이브 7436
스틸, 브라이언
미국 80026 콜로라도 라파예트 일리아드 웨이 1074
위레이, 조지, 에이.
미국 92130 캘리포니아 샌디에고 브리태니 포레스트 레인 5740
셰크하트, 샤생크
미국 80516 인디애나 에리에 홀맨 드라이브 1597

(74) 대리인 남상선

심사청구 : 있음

(54) 향상된 링크 제어를 제공하는 고속 데이터 레이트인터페이스

요약

미리-선택된 세트의 디지털 제어 및 표현 데이터를 전달하기 위한 통신 프로토콜을 형성하기 위해 함께 링크된 패킷 구조들을 사용하여 통신 경로를 통해 호스트와 클라이언트 사이에서 디지털 데이터를 전달하기 위한 데이터 인터페이스가 제시된다. 신호 프로토콜은 통신 프로토콜을 형성하는 패킷들을 생성, 전송 및 수신하고 디지털 데이터를 하나 이상의 데이터 패킷 타입들로 형성하도록 구성된 링크 제어기들에 의해 사용되며, 링크 제어기들 중 적어도 하나는 호스트 장치에 위치하며 통신 경로를 통해 클라이언트로 연결된다. 상기 인터페이스는 짧은-범위의 "직렬" 타입 데이터 링크를 통해 비용-효율적, 저전력, 양방향성 및 고속 데이터 전달 메커니즘을 제공하며, 상기 "직렬" 타입 데이터 링크는 착용가능한 마이크로-디스플레이들을 휴대용 컴퓨터들 및 무선 통신 장치들로 연결하는데 특히 유용한 소형 커넥터들과 얇고 플렉서블한 케이블들을 사용하여 구현한다.

대표도

도 5

명세서

기술분야

본 특허출원은 "스위칭 가능 임계 차동 인터페이스"라는 명칭으로 2003년 9월 10일자 제출된 예비 출원 60/519,833호에 대한 우선권을 주장하며, 이는 본원의 양수인에게 양도되었으며 이로써 본원에 명백히 통합된다.

본 발명의 실시예들은 호스트 장치와 클라이언트 장치 사이에 높은 데이터 전송률로 신호를 교환 또는 전송하기 위한 디지털 신호 프로토콜 및 프로세스에 관한 것이다. 보다 구체적으로, 본원은 내부 및 외부 장치 애플리케이션을 갖는 저전력 고속 데이터 전송 메커니즘을 이용하여 최종 사용자에게 프리젠테이션 또는 디스플레이하기 위해 호스트 또는 제어기 장치로부터 클라이언트 장치로 멀티미디어나 다른 종류의 디지털 신호들을 전송하기 위한 기술에 관한 것이다.

배경기술

컴퓨터, 전자오락 관련 제품 및 다양한 비디오 기술(예를 들어 DVD 및 고화질 VCR)은 지난 수년간 상당히 진보하여, 어떤 종류의 텍스트를 포함하더라도 점점 더 높은 해상도의 스틸, 비디오, 주문형 비디오 및 그래픽 이미지 표현을 이러한 기기의 최종 사용자들에게 제공해왔다. 이러한 진보는 고화질 비디오 모니터, HDTV 모니터 또는 특수 이미지 투사 엘리먼트와 같은 고해상도 전자 시청 장치의 사용을 요구했다. CD 타입 음향 재생, DVD, 서라운드 음향, 및 관련 오디오 신호 출력을 갖는 다른 장치들을 이용하는 경우와 같이 이러한 시각적 이미지와 고화질 또는 고품질 오디오 데이터의 결합이 사용되어 최종 사용자에게 보다 사실적이고 콘텐츠가 풍부한 또는 정확한 멀티미디어 체험을 일으킨다. 또한, 최종 사용자에게 대한 오디오 전용 프리젠테이션을 위해 MP3 플레이어와 같은 고도의 이동성, 고품질 사운드 시스템 및 음악 전송 메커니즘이 개발되었다. 그 결과, 컴퓨터에서부터 텔레비전, 심지어 전화기까지 현재 익숙한 그리고 높은 또는 프리미엄급의 출력을 기대하는 시판용 전자 기기에 대한 일반 사용자들의 기대치가 증가하게 되었다.

전자공학 제품을 포함하는 통상의 비디오 프리젠테이션 시나리오에서, 비디오 데이터는 통상적으로 느린 또는 중간이라고 할 수 있는 초당 1 내지 10 킬로비트 정도의 속도로 현재의 기술을 이용하여 전송된다. 이 데이터는 원하는 시청 장치에서의 지연(나중에) 재생을 위해 버퍼링되거나 일시적인 또는 장기 메모리 장치에 저장된다. 예를 들어, 이미지들은 이미지의 디지털 표현에 유용한 데이터를 수신 또는 송신하기 위한 모델이나 다른 종류의 인터넷 접속 장치를 갖는 컴퓨터에 상주하는 프로그램을 이용하여 "전역으로" 또는 인터넷을 이용하여 전송될 수 있다. 무선 모델을 장착한 휴대용 컴퓨터나, 무선 개인 휴대 단말(PDA) 또는 무선 전화 등의 무선 장치를 이용하여 비슷한 전송이 이루어질 수 있다.

데이터가 수신되면, 재생을 위해 소형 하드드라이브 등의 내부 또는 외부 저장 장치를 포함하는, RAM이나 플레이 메모리와 같은 메모리 엘리먼트, 회로 또는 장치에 로컬 저장된다. 데이터량 및 이미지 해상도에 따라 재생은 비교적 빠르게 시작할 수도 있고 장기간 지연될 수도 있다. 즉, 어떤 경우에는 이미지 프리젠테이션은 많은 데이터를 요구하지 않는 매우 작은 또는 저해상도 이미지에 대해, 또는 약간의 지연 후 더 많은 자료가 전송되는 동안 일부 자료가 프리젠테이션되도록 어떤 종류의 버퍼링을 이용하여 어느 정도의 실시간 재생을 허용한다. 전송 링크에 인터럽트가 없거나, 사용되고 있는 전송 채널에 대한 다른 시스템이나 사용자들로부터의 간섭이 없을 경우, 프리젠테이션이 시작되면 전송은 시청 장치의 최종 사용자에게 상당히 투명하다. 물론, 다수의 사용자가 무선 인터넷 접속과 같은 단일 통신 경로를 공유하는 경우, 전송은 인터럽트될 수도 있고 원하는 것보다 느려질 수도 있다.

스틸 이미지나 모션 비디오를 생성하는데 사용되는 데이터는 흔히 통신 링크를 통한 데이터 전송 속도를 높이기 위해 JPEG(Joint Photographic Experts Group), MPEG(Motion Picture Experts Group), 및 미디어, 컴퓨터, 통신 산업에서 잘 알려진 다른 표준화 기관이나 회사에 의해 지정된 것과 같이 잘 알려진 여러 기술 중 하나를 이용하여 압축된다. 이는 소정량의 정보를 전송하는데 더 적은 수의 비트를 사용함으로써 이미지 또는 데이터 전송을 더욱 빠르게 한다.

메모리나 자기 또는 광 저장 엘리먼트 등의 저장 메커니즘을 갖는 컴퓨터와 같은 "로컬" 장치에 또는 다른 수신측 장치에 데이터가 전송되면, 결과적인 정보가 압축 해제되고(또는 특별한 디코딩 플레이어를 이용하여 재생되고), 필요하다면 디코딩되어, 해당하는 유효 프리젠테이션 해상도 및 제어 엘리먼트에 기반한 적절한 프리젠테이션이 준비된다. 예를 들어, 원하는 대로 또는 필요에 따라 다양한 다른 해상도가 가능한 것이 일반적이지만, X 대 Y 픽셀들의 스크린 해상도에 관한 통상의 컴퓨터 비디오 해상도는 통상적으로 480×640 픽셀의 낮은 해상도에서부터 600×800을 거쳐 1024×1024 범위이다.

또한, 이미지 프리젠테이션은 미리 정해진 특정 색상 레벨 또는 색상 심도(색상 생성에 사용되는 픽셀 당 비트) 및 명암에 관해 이미지를 조정하며 임의의 추가 오버헤드 비트가 사용되는 주어진 비디오 제어기의 성능 및 이미지 콘텐츠에 영향을 받는다. 예를 들어, 다른 값들이 사용될 수도 있지만, 통상의 컴퓨터 프리젠테이션은 다양한 색상(명암 및 색조)을 표현하는데 픽셀 당 약 8 내지 32, 또는 그 이상의 비트를 예상하게 된다.

상기 값들로부터 소정의 스크린 이미지는 최저에서부터 최고의 통상 해상도 및 심도 범위에 걸쳐 각각 2.45 메가비트(Mb) 내지 약 33.55 Mb의 데이터 전송을 요구할 것을 알 수 있다. 초당 30 프레임 속도의 비디오 또는 모션 타입 이미지를 볼 때, 필요한 데이터량은 초당 약 73.7 내지 1,006 메가비트의 데이터(Mbps), 또는 초당 약 9.21 내지 125.75 메가바이트(MBps)이다. 추가로, 멀티미디어 프리젠테이션을 위해, 또는 CD 음질 음악과 같은 개별 고해상도 오디오 프리젠테이션 등 이미지와 함께 오디오 데이터를 프리젠테이션하는 것이 바람직할 수도 있다. 또한, 대화식 명령, 제어 또는 신호와 관계하는 추가 신호가 사용될 수도 있다. 이들 각각은 전송될 훨씬 더 많은 데이터의 추가를 옵션으로 한다. 더욱이, 고화질(HD) 텔레비전을 포함하는 더욱 새로운 전송 기술들 및 영화 레코딩은 훨씬 많은 데이터 및 제어 정보를 추가할 수 있다. 어떤 경우에도, 고품질 또는 고해상도 이미지 데이터 및 고품질 오디오 정보 또는 데이터 신호를 최종 사용자에게 전송하여 풍부한 콘텐츠 체험을 일으키는 것이 바람직할 경우, 프리젠테이션 엘리먼트와 이러한 타입의 데이터를 제공하도록 구성된 소스 또는 호스트 장치 사이에 고속 데이터 링크가 필요하다.

초당 약 115 킬로바이트(KBps) 또는 920 킬로비트(Kbps)의 데이터 전송률은 일상적으로 어떤 최신 직렬 인터페이스에 의해 처리될 수 있다. USB 직렬 인터페이스와 같은 다른 인터페이스들은 12 MBps의 높은 속도의 데이터 전송을 제공할 수 있으며, 전기 전자 학회(IEEE) 1394 표준을 이용하여 구성된 것과 같은 특수 고속 전송은 100 내지 400 MBps 정도의 속도로 일어날 수 있다. 공교롭게도, 이러한 속도는 미래의 무선 데이터 장치 및 고해상도, 풍부한 콘텐츠, 휴대용 비디오 디스플레이나 오디오 장치를 구동하기 위한 출력 신호를 제공하는 다른 서비스에서의 사용이 기대되는 상술한 바람직한 높은 데이터 전송률에 미치지 않는다. 이는 사업용 컴퓨터 및 다른 프리젠테이션, 게임기 등을 포함한다. 추가로, 이들 인터페이스는 작동을 위해 상당량의 호스트 또는 시스템 및 클라이언트 소프트웨어의 사용을 요구한다. 또한, 이들의 소프트웨어 프로토콜 스택은 특히 이동 무선 장치나 전화 애플리케이션에 관련하여 바람직하지 않게 많은 양의 오버헤드를 생성한다. 이러한 장치는 이미 부과된 연산 용량은 물론, 엄밀한 메모리 및 전력 소비 한계를 갖는다. 더욱이, 이들 인터페이스의 일부는 비용이 추가되거나 단순히 너무 많은 전력을 소비하는 매우 심미 지향적인 모바일 애플리케이션, 복합 커넥터에는 만족스럽지 않고 너무 무거운 부피가 큰 케이블을 이용한다.

아날로그 비디오 그래픽 어댑터(VGA), 디지털 비디오 인터랙티브(DVI) 또는 기가비트 비디오 인터페이스(GVIF)와 같은 다른 공지된 인터페이스들이 있다. 이들 중 처음 두 개는 보다 높은 전송률로 데이터를 처리하지만 무거운 케이블을 이용하고 몇 와트 정도의 상당량의 전력을 소비하는 병렬형 인터페이스이다. 이들 특성 중 어느 것도 휴대용 전자 장치에 사용할 여지가 없다. 세 번째 인터페이스 또한 너무 많은 전력을 소비하며 고가의 또는 부피가 큰 커넥터를 사용한다.

상기 인터페이스들 중 일부, 및 고정식 컴퓨터 기기에 대한 데이터 전송과 관련된 다른 초고속 데이터 시스템/프로토콜 또는 전송 메커니즘에 대해 다른 중요한 결점이 있다. 원하는 데이터 전송률을 제공하기 위해서는 상당량의 전력 및/또는 높은 전력 레벨에서의 동작이 요구된다. 이는 매우 모바일 소비자 지향적인 제품에 대해 이러한 기술들의 유용성을 크게 감소시킨다.

일반적으로, 광섬유형 접속 및 전송 엘리먼트 등의 대안을 이용하여 이러한 데이터 전송률을 제공하기 위해서는, 사실상 민간 소비자 지향 제품에 대해 원하는 것보다 훨씬 높은 복잡도 및 비용을 이끄는 다수의 추가 변환기 및 엘리먼트를 필요로 한다. 광학 시스템의 일반적인 고가 특성은 차치하고, 그 전력 요건 및 복잡도는 경량이고 저전력인 휴대용 애플리케이션에 대한 일반적인 사용을 방해한다.

휴대용, 무선 또는 모바일 애플리케이션 산업에서 부족했던 것은 고도의 모바일 최종 사용자들에게 오디오 기반이든, 비디오 기반이든, 멀티미디어 기반이든 고품질의 프리젠테이션 체험을 제공하는 기술이다. 즉, 휴대용 컴퓨터, 무선 전화, PDA 또는 다른 고도의 이동 통신 장치나 기기를 이용할 때, 현재 사용되고 있는 비디오 및 오디오 프리젠테이션 시스템 또는 장치는 단순히 원하는 고품질 레벨로 출력을 전달할 수 없다. 흔히, 부족한 것으로 인식되는 품질은 고품질 프리젠테이션 데이터 전송에 필요한 얻기 어려운 높은 데이터 전송률의 결과이다. 이는 최종 사용자에게 프리젠테이션하기 위한 보다 효율적이고 진보된 또는 특정 부과된 외부 장치로의 전송, 또는 호스트와 컴퓨터, 게임기, 및 이동 전화와 같은 무선 장치 등의 휴대용 장치 내부의 클라이언트 사이의 전송을 모두 포함할 수 있다.

후자의 경우, 내부 비디오 스크린에 점점 더 높은 해상도를 부가하는데 있어서, 그리고 소위 3세대 전화와 같은 무선 장치 및 소위 랩탑 컴퓨터에 대한 다른 특수한 입력 및/또는 출력 장치 및 접속에 있어서 장족의 발전이 이루어졌다. 그러나 내

부 데이터 버스 및 접속이 호스트 및/또는 각종 다른 제어 엘리먼트 및 출력 컴포넌트들이 상주하는 메인 하우징에 비디오 스크린 또는 다른 엘리먼트를 장착 또는 접속하는 회전 또는 슬라이딩 힌지나 힌지형 구조와 교차하는 브리징을 포함할 수 있다. 일례로, 무선 전화에 대해 원하는 스루풋을 얻기 위해 90개까지 또는 그 이상의 컨덕터를 필요로 할 수 있는 종래의 기술을 이용하여 높은 스루풋의 데이터 전송 인터페이스를 구성하는 것은 매우 어렵다. 이는 극복해야 하는 많은 제조, 상당한 비용 및 신뢰성 면에 있어 도전적인 이슈를 나타낸다.

이러한 이슈 및 요건은 설비나 다른 소비자 장치에 통신 또는 연산형 장치가 추가되어 진보된 데이터 용량, 인터넷 및 데이터 전송 접속 또는 내장형 엔터테인먼트를 제공하는 고정 위치 설치에서 나타나고 있다. 다른 예는 개별 비디오 및 오디오 프리젠테이션 스크린이 시트백에 장착된 비행기나 버스가 된다. 그러나 이러한 상황에서는 정보의 프리젠테이션을 위한 상호 접속 링크 또는 채널을 가진 시각적 스크린 또는 오디오 출력으로부터 거리를 두고 메인 저장, 처리 또는 통신 제어 엘리먼트를 배치하는 것이 종종 더 편리하고 효율적이며 확실히 실용적이다. 이 링크는 상술한 바와 같이 원하는 스루풋을 달성하기 위해 상당량의 데이터를 처리할 필요가 있을 것이다.

따라서 데이터를 제공하는 호스트 장치와 최종 사용자에게 출력을 제시하는 클라이언트 디스플레이 장치 또는 엘리먼트 사이의 데이터 스루풋을 증가시키기 위한 새로운 전송 메커니즘이 필요하다.

미국 특허출원 10/020,520호 및 10/236,657호에 이러한 새로운 전송 메커니즘이 제안되었으며, 상기 출원은 모두 "고속 데이터 신호 전송을 위한 통신 프로토콜 및 인터페이스 생성 및 구현"이라는 명칭으로 현재 허여되었으며, 본 발명의 양수인에게 할당되어 본원에 참조로 포함된다. 또한, 출원 10/860,116호는 "고속 데이터용 신호 프로토콜 및 인터페이스 생성 및 구현"이라는 명칭이다. 이들 출원에 개시된 기술은 고속 데이터 신호에서 상당량의 데이터 전송 속도를 크게 향상시킬 수 있다. 그러나 데이터 전송률을 증가시키라는, 특히 비디오 프리젠테이션에 관련된 요구가 계속해서 증가하고 있다. 데이터 신호 기술에서 다른 진행중인 개발에도, 더욱 빠른 전송 속도, 개선된 통신 링크 효율 및 더욱 강력한 통신 링크를 얻으려는 노력이 여전히 필요하다. 따라서 호스트와 클라이언트 장치 사이의 데이터 스루풋 증가가 필요한 새로운 또는 개선된 전송 메커니즘을 개발하는 것이 계속해서 필요하다.

발명의 상세한 설명

상기 결점 및 종래 기술 기존의 다른 결점들은 호스트 장치와 수신측 클라이언트 장치 사이에 높은 데이터 전송률로 데이터를 전송하기 위한 새로운 프로토콜 및 데이터 전송 수단, 방법 및 메커니즘이 개발된 본 발명의 실시예에 의해 해결된다.

본 발명의 실시예들은 서로 연결되어 호스트와 클라이언트 장치 간 미리 선택된 디지털 제어 및 프리젠테이션 데이터 세트를 전달하는 통신 프로토콜을 형성하는 다수의 또는 일련의 패킷 구조를 이용하는 통신 경로를 통해 호스트 장치와 클라이언트 장치 간에 고속으로 디지털 데이터를 전송하기 위한 이동 데이터 디지털 인터페이스(MDDI)에 관련된다. 신호 통신 프로토콜 또는 링크 계층이 호스트 또는 클라이언트 링크 제어기의 물리 계층에 의해 사용된다. 호스트 장치에 상주하는 적어도 하나의 링크 제어기가 통신 경로 또는 링크를 통해 클라이언트 장치에 연결되어, 통신 프로토콜을 형성하는 패킷들을 생성, 송신 및 수신하고, 디지털 프리젠테이션 데이터를 하나 이상의 타입의 데이터 패킷으로 형성하도록 구성된다. 인터페이스는 호스트와 클라이언트 사이의 양방향 정보 전송을 제공하고, 이는 공통 전체 하우징 또는 지지 구조 내에 상주할 수 있다.

디지털 CMOS 칩에 쉽게 구현될 수 있는 차동 드라이버 및 수신기를 제외하고, 구현은 본래 일반적으로 모두 디지털이고, 6개와 같이 적은 신호를 필요로 하며, 시스템 설계자에게 편리한 거의 임의의 데이터 전송률로 동작한다. 간단한 물리 및 링크 계층 프로토콜은 통합이 쉬워지게 하고, 이러한 단순성 및 절전 상태는 휴대용 시스템이 매우 낮은 시스템 전력 소비를 갖게 할 수 있다.

사용 및 승인에 도움이 되도록, 인터페이스는 장치 비용이 거의 추가되지 않고, 표준 배터리 전압을 이용하여 인터페이스를 통해 디스플레이에 전력을 공급하는 동안 전력 소비가 거의 없으며, 포켓용 폼-팩터를 갖는 장치를 제공할 수 있다. 인터페이스는 HDTV 이상의 해상도를 지원하도록 스케일링될 수 있고, 디스플레이 장치에 동시 스테레오 비디오 및 7.1 오디오를 지원하며, 임의의 스크린 영역에 대해 조건부 업데이트를 수행하고, 양방향으로 다중 데이터 타입을 지원한다.

본 발명의 부가적인 양상들에서, 적어도 하나의 클라이언트 링크 제어기 또는 클라이언트 수신기가 클라이언트 장치에 배치되고 통신 경로 또는 링크를 통해 호스트 장치에 연결된다. 또한, 클라이언트 링크 제어기는 통신 프로토콜을 형성하는 패킷을 생성, 송신 및 수신하고, 디지털 프리젠테이션 데이터를 하나 이상의 타입의 데이터 패킷으로 형성하도록 구성된다. 일반적으로, 호스트 또는 링크 제어기는 명령 또는 특정 타입의 신호 준비 및 문의 처리에 사용되는 데이터 패킷을 처

리하기 위해 상태 머신을 이용하지만, 통신 프로토콜에 사용되는 더 적은 복합 패킷의 일부 및 데이터를 조종하기 위해 더 저속의 범용 프로세서를 사용할 수 있다. 호스트 제어기는 하나 이상의 차동 라인 드라이버를 포함하는 한편, 클라이언트 수신기는 통신 경로에 연결된 하나 이상의 차동 라인 수신기를 포함한다.

패킷은 상이한 가변 길이를 갖는 미리 결정된 개수의 패킷으로 미리 정해진 고정 길이를 가지며 호스트와 클라이언트 장치 사이에 통신하는 미디어 프레임 내에서 함께 그룹화된다. 패킷은 각각 패킷 길이 필드, 하나 이상의 패킷 데이터 필드 및 순환 중복 검사 필드를 포함한다. 서브 프레임 헤더 패킷은 호스트 링크 제어기로부터 다른 패킷들의 최초 전송시 전송 또는 위치가 정해진다. 통신 프로토콜에 의해 하나 이상의 비디오 스트림 타입 패킷 및 오디오 스트림 타입 패킷이 사용되어 클라이언트 장치 사용자에게 프리젠테이션하기 위해 순방향 링크를 통해 호스트에서 클라이언트로 각각 비디오 타입 데이터 및 오디오 타입 데이터를 전송한다. 통신 프로토콜에 의해 하나 이상의 역방향 링크 캡슐화 타입 패킷이 사용되어 클라이언트 장치로부터 호스트 링크 제어기로 데이터를 전송한다. 일부 실시예에서의 이러한 전송은 적어도 하나의 MDDI 장치를 갖는 내부 제어기로부터 내부 비디오 스크린으로의 데이터 전송을 포함한다. 다른 실시예들은 내부 사운드 시스템으로의 전송 및 조이스틱 및 복합 키보드를 포함하는 각종 입력 장치로부터 내부 호스트 장치로의 전송을 포함한다.

호스트 링크 제어기에 의해 채움 타입 패킷이 생성되어 데이터를 갖지 않는 순방향 링크 전송 기간을 차지한다. 통신 프로토콜에 의해 다수의 다른 패킷이 사용되어 비디오 정보를 전송한다. 이러한 패킷은 컬러맵, 비트 블록 전송, 비트맵 영역 채움, 비트맵 패턴 채움 및 투명색 인에이블 타입 패킷을 포함한다. 사용자 정의 스트림 타입 패킷이 통신 프로토콜에 의해 사용되어 인터페이스-사용자 정의 데이터를 전송한다. 키보드 데이터 및 포인팅 장치 데이터 타입 패킷이 통신 프로토콜에 의해 사용되어 상기 클라이언트 장치와 관련된 사용자 입력 장치에 또는 사용자 입력 장치로부터 데이터를 전송한다. 링크 섀다운 타입 패킷이 통신 프로토콜에 의해 사용되어 상기 통신 경로를 통해 어느 한 방향으로의 데이터 전송을 종료한다.

통신 경로는 일반적으로 연속된 4개 이상의 컨덕터 및 실드를 갖는 케이블을 포함 또는 이용한다. 추가로, 원하는 대로 일부가 플렉서블 기판상에 있는 인쇄 배선 또는 컨덕터가 사용될 수 있다.

호스트 링크 제어기는 클라이언트가 상기 인터페이스를 통해 어떤 데이터 및 데이터 전송률을 제공할 수 있는지를 결정하기 위해 상기 클라이언트로부터의 디스플레이 성능 정보를 요구한다. 클라이언트 링크 제어기는 적어도 하나의 클라이언트 용량 타입 패킷을 이용하여 호스트 링크 제어기에 디스플레이 또는 프리젠테이션 성능을 전달한다. 통신 프로토콜에 의해 다수의 전송 모드가 사용되며, 각각 소정 시간대에 걸쳐 상이한 최대 개수의 데이터 비트의 병렬 전송을 가능하게 하고, 각 모드는 호스트와 클라이언트 링크 제어기 사이의 협상에 의해 선택할 수 있다. 전송 모드는 데이터 전송 도중 동적으로 조정할 수 있으며, 역방향 링크에 순방향 링크에 사용된 것과 동일한 전송 모드가 사용될 필요는 없다.

본 발명의 일부 실시예의 다른 양상에 있어서, 호스트 장치는 무선 전화, 무선 PDA, 또는 무선 모뎀이 배치된 휴대용 컴퓨터 등의 무선 통신 장치를 포함한다. 통상의 클라이언트 장치는 마이크로 디스플레이 장치 및/또는 휴대용 오디오 프리젠테이션 시스템 등의 휴대용 비디오 디스플레이를 포함한다. 더욱이, 호스트는 클라이언트 장치 사용자에게 제시하기 위해 전송될 프리젠테이션 또는 멀티미디어 데이터를 저장하는 저장 수단 또는 엘리먼트를 사용할 수 있다.

일부 실시예들의 또 다른 양상에 있어서, 호스트 장치는 무선 전화, 무선 PDA 또는 휴대용 컴퓨터 등의 무선 통신 장치와 같은 휴대용 전자 장치 내에 있는 후술하는 드라이버와 함께 제어기 또는 통신 링크 제어 장치를 포함한다. 이러한 구성에서 통상의 클라이언트 장치는 호스트에 연결되며 동일 장치 내에 있는, 그리고 이동 전화의 고해상도 스크린과 같이 내부 비디오 디스플레이에 연결되는 클라이언트 회로나 집적 회로 또는 모듈, 및/또는 휴대용 오디오 프리젠테이션 시스템 또는 대안으로 어떤 종류의 입력 시스템 또는 장치를 포함한다.

도면의 간단한 설명

본 발명의 각종 실시예의 구조 및 동작은 물론, 본 발명의 추가적인 특징 및 이점은 첨부 도면을 참조로 하기에 상세히 설명한다. 도면에서 동일 참조 부호는 일반적으로 동일하고 기능적으로 유사하며 그리고/또는 구조적으로 유사한 엘리먼트 또는 처리 단계들을 나타내며, 엘리먼트가 처음 나타나는 도면은 참조 부호에서 맨 왼쪽 숫자(들)로 지시된다.

도 1a는 휴대용 컴퓨터 또는 다른 데이터 처리 장치와 관련한 마이크로 디스플레이 장치 또는 프로젝터의 사용을 포함하여, 본 발명의 실시예들이 작동할 수 있는 기본 환경을 나타낸다.

도 1b는 무선 트랜시버와 함께 사용되는 마이크로 디스플레이 장치 또는 프로젝터 및 오디오 프리젠테이션 엘리먼트의 사용을 포함하여, 본 발명의 실시예들이 작동할 수 있는 기본 환경을 나타낸다.

도 1c는 휴대용 컴퓨터에 사용되는 내부 디스플레이 또는 오디오 프리젠테이션 장치의 사용을 포함하여, 본 발명의 실시예들이 작동할 수 있는 기본 환경을 나타낸다.

도 1d는 무선 트랜시버에 사용되는 내부 디스플레이 또는 오디오 프리젠테이션 엘리먼트의 사용을 포함하여, 본 발명의 실시예들이 작동할 수 있는 기본 환경을 나타낸다.

도 2는 호스트 및 클라이언트 상호 접속에 의한 이동 디지털 데이터 인터페이스의 전체 개념을 설명한다.

도 3은 클라이언트 장치로부터 호스트 장치로의 데이터 전송을 실현하는데 유용한 패킷 구조를 나타낸다.

도 4는 MDDI 링크 제어기의 사용 및 타입 1 인터페이스에 대한 물리 데이터 링크 컨덕터를 통해 호스트와 클라이언트 사이에 전달되는 신호 타입을 나타낸다.

도 5는 MDDI 링크 제어기의 사용 및 타입 2, 3, 4 인터페이스에 대한 물리 데이터 링크 컨덕터를 통해 호스트와 클라이언트 사이에 전달되는 신호 타입을 나타낸다.

도 6은 인터페이스 프로토콜의 구현에 사용되는 프레임 및 서브 프레임 구조를 나타낸다.

도 7은 인터페이스 프로토콜의 구현에 사용되는 패킷의 일반 구조를 나타낸다.

도 8은 서브 프레임 헤더 패킷의 포맷을 나타낸다.

도 9는 채움 패킷의 포맷 및 내용을 나타낸다.

도 10은 비디오 스트림 패킷의 포맷을 나타낸다.

도 11a-11e는 도 10에 사용된 비디오 데이터 포맷 기술자의 포맷 및 내용을 나타낸다.

도 12는 데이터의 압축 및 비압축 포맷의 사용을 나타낸다.

도 13은 오디오 스트림 패킷의 포맷을 나타낸다.

도 14는 데이터의 바이트 정렬 및 압축된 PCM 포맷의 사용을 나타낸다.

도 15는 사용자 정의 스트림 패킷의 포맷을 나타낸다.

도 16은 컬러맵 패킷의 포맷을 나타낸다.

도 17은 역방향 링크 캡슐화 패킷의 포맷을 나타낸다.

도 18은 클라이언트 용량 패킷의 포맷을 나타낸다.

도 19는 키보드 데이터 패킷의 포맷을 나타낸다.

도 20은 포인팅 장치 데이터 패킷의 포맷을 나타낸다.

도 21은 링크 셧다운 패킷의 포맷을 나타낸다.

도 22는 클라이언트 요청 및 상태 패킷의 포맷을 나타낸다.

도 23은 비트 블록 전송 패킷의 포맷을 나타낸다.

도 24는 비트맵 영역 채움 패킷의 포맷을 나타낸다.

- 도 25는 비트맵 패턴 채움 패킷의 포맷을 나타낸다.
- 도 26은 통신 링크 데이터 채널 패킷의 포맷을 나타낸다.
- 도 27은 인터페이스 타입 핸드오프 요청 패킷의 포맷을 나타낸다.
- 도 28은 인터페이스 타입 확인 응답 패킷의 포맷을 나타낸다.
- 도 29는 실행 타입 핸드오프 패킷의 포맷을 나타낸다.
- 도 30은 순방향 오디오 채널 인에이블 패킷의 포맷을 나타낸다.
- 도 31은 역방향 오디오 샘플 레이트 패킷의 포맷을 나타낸다.
- 도 32는 디지털 콘텐츠 보호 오버헤드 패킷의 포맷을 나타낸다.
- 도 33은 투명색 인에이블 패킷의 포맷을 나타낸다.
- 도 34는 왕복 지연 측정 패킷의 포맷을 나타낸다.
- 도 35는 왕복 지연 측정 패킷 도중의 이벤트 타이밍을 나타낸다.
- 도 36은 본 발명의 구현에 유용한 CRC 생성기 및 검사기의 샘플 구현을 나타낸다.
- 도 37a는 데이터 패킷 전송시 도 36의 장치에 대한 CRC 신호의 타이밍을 나타낸다.
- 도 37b는 데이터 패킷 수신시 도 36의 장치에 대한 CRC 신호의 타이밍을 나타낸다.
- 도 38은 회선 경합이 없는 통상의 서비스 요청 처리 단계들을 나타낸다.
- 도 39는 링크 재기동 시퀀스가 시작한 후 선언되어 링크 기동과 경합하는 통상의 서비스 처리 단계들을 나타낸다.
- 도 40은 데이터 시퀀스가 DATA-STB 인코딩을 이용하여 어떻게 전송될 수 있는지를 나타낸다.
- 도 41은 호스트에서 입력 데이터로부터 DATA 및 STB 신호를 생성한 다음 클라이언트에서 데이터를 복원하는데 유용한 회로를 나타낸다.
- 도 42는 일 실시예의 구현에 유용한 드라이버 및 종단 저항을 나타낸다.
- 도 43은 호스트로부터의 서비스를 확보하기 위해 클라이언트에 의해 이용되는, 그리고 이러한 서비스를 제공하기 위해 호스트에 의해 이용되는 단계들 및 신호 레벨들을 나타낸다.
- 도 44는 Data0, 다른 데이터 라인(DataX) 및 스트로브 라인(Stb) 상에서의 트랜지션 사이의 상대적 간격을 나타낸다.
- 도 45는 호스트가 패킷 전송 후 호스트 드라이버를 디세이블(disable)할 때 일어날 수 있는 응답 지연의 출현을 나타낸다.
- 도 46은 호스트가 호스트 드라이버로 하여금 패킷을 전송할 수 있게 할 때 일어날 수 있는 응답 지연의 출현을 나타낸다.
- 도 47은 호스트 수신기 입력에서, 전송되고 있는 데이터의 타이밍과 스트로브 펄스의 선단 및 후미 에지 사이의 관계를 나타낸다.
- 도 48은 역방향 데이터 타이밍에 의해 전개되는 스위칭 특성 및 해당 클라이언트 출력 지연을 나타낸다.

- 도 49는 상태 머신을 이용하여 동기화가 구현될 수 있는 신호 처리 단계 및 조건의 고 레벨도를 나타낸다.
- 도 50은 MDDI를 이용하는 시스템에서 순방향 및 역방향 경로 상에서의 신호 처리시 부딪히는 통상의 지연량을 나타낸다.
- 도 51은 한계 왕복 지연 측정을 나타낸다.
- 도 52는 역방향 링크 데이터 전송률 변화를 나타낸다.
- 도 53은 역방향 레이트 제수 대 순방향 링크 데이터 레이트 값들의 그래픽 표현을 나타낸다.
- 도 54a 및 도 54b는 인터페이스 동작시의 단계들을 나타낸다.
- 도 55는 패킷을 처리하는 인터페이스 장치의 개요를 나타낸다.
- 도 56은 순방향 링크 패킷의 포맷을 나타낸다.
- 도 57은 타입 1 링크 인터페이스에서의 전파 지연 및 왜곡의 통상 값들을 나타낸다.
- 도 58은 인터페이스를 통한 전형적인 신호 처리에 대한 타입 1 링크 상에서의 데이터, 스트로브 및 클록 복원 타이밍을 나타낸다.
- 도 59는 타입 2, 타입 3 또는 타입 4 링크 인터페이스에서의 전파 지연 및 왜곡의 통상 값들을 나타낸다.
- 도 60a, 도 60b 및 도 60c는 2개의 데이터 신호 및 MDDI_Stb의 서로에 대한 타이밍이 각각 이상적일 확률, 이를 확률 및 늦을 확률을 나타낸다.
- 도 61은 타입 1/타입 2 인터페이스에 사용되는 전형적인 커넥터들의 인터페이스 핀 할당을 나타낸다.
- 도 62a 및 도 62b는 타입 1 및 타입 2 인터페이스 각각에 대해 가능한 MDDI_Data 및 MDDI_Stb 파형을 나타낸다.
- 도 63은 상태 머신을 이용하여 동기화가 구현될 수 있는 다른 신호 처리 단계 및 조건의 고 레벨도를 나타낸다.
- 도 64는 일련의 클록 사이클과 각종 역방향 링크 패킷 비트 및 제수값들의 타이밍 간의 전형적인 상대 타이밍을 나타낸다.
- 도 65는 전형적인 에러 코드 전송 처리를 나타낸다.
- 도 66은 에러 코드 전송 처리에 유용한 장치를 나타낸다.
- 도 67a는 코드 오버로드를 위한 에러 코드 전송 처리를 나타낸다.
- 도 67b는 코드 수신을 위한 에러 코드 전송 처리를 나타낸다.
- 도 68a는 호스트가 웨이크업을 시작하는 처리 단계를 나타낸다.
- 도 68b는 클라이언트가 웨이크업을 시작하는 처리 단계를 나타낸다.
- 도 68c는 회선 경합에 의해 호스트 및 클라이언트가 웨이크업을 시작하는 처리 단계를 나타낸다.
- 도 69는 요청 VCP 피쳐 패킷의 포맷을 나타낸다.
- 도 70은 VCP 피쳐 응답 패킷의 포맷을 나타낸다.
- 도 71은 VCP 피쳐 응답 리스트의 포맷을 나타낸다.

- 도 72는 세트 VCP 피쳐 패킷의 포맷을 나타낸다.
- 도 73은 요청 유효 파라미터 패킷의 포맷을 나타낸다.
- 도 74는 유효 파라미터 응답 패킷의 포맷을 나타낸다.
- 도 75는 알파 커서 이미지 용량 패킷의 포맷을 나타낸다.
- 도 76은 알파 커서 투명 맵 패킷의 포맷을 나타낸다.
- 도 77은 알파 커서 이미지 오프셋 패킷의 포맷을 나타낸다.
- 도 78은 알파 커서 비디오 스트림 패킷의 포맷을 나타낸다.
- 도 79는 스케일 비디오 스트림 용량 패킷의 포맷을 나타낸다.
- 도 80은 스케일 비디오 스트림 셋업 패킷의 포맷을 나타낸다.
- 도 81은 스케일 비디오 스트림 확인 응답 패킷의 포맷을 나타낸다.
- 도 82는 스케일 비디오 스트림 패킷의 포맷을 나타낸다.
- 도 83은 요청 지정 상태 패킷의 포맷을 나타낸다.
- 도 84는 유효 상태 응답 리스트 패킷의 포맷을 나타낸다.
- 도 85a는 패킷 처리 지연 파라미터 패킷의 포맷을 나타낸다.
- 도 85b는 지연 파라미터 리스트 아이템의 포맷을 나타낸다.
- 도 86은 개인용 디스플레이 용량 패킷의 포맷을 나타낸다.
- 도 87a는 클라이언트 에러 보고 패킷의 포맷을 나타낸다.
- 도 87b는 에러 보고 리스트 아이템의 포맷을 나타낸다.
- 도 88은 클라이언트 식별 패킷의 포맷을 나타낸다.
- 도 89는 대안 디스플레이 용량 패킷의 포맷을 나타낸다.
- 도 90은 레지스터 액세스 패킷의 포맷을 나타낸다.
- 도 91a~91c는 가시물을 줄이기 위한 2개의 디스플레이 버퍼 사용을 나타낸다.
- 도 92는 디스플레이 리프레시가 이미지 전송보다 빠른 2개의 버퍼를 나타낸다.
- 도 93은 디스플레이 리프레시가 이미지 전송보다 느린 2개의 버퍼를 나타낸다.
- 도 94는 디스플레이 리프레시가 이미지 전송보다 훨씬 빠른 2개의 버퍼를 나타낸다.
- 도 95는 디스플레이 리프레시가 이미지 전송보다 빠른 3개의 버퍼를 나타낸다.
- 도 96은 디스플레이 리프레시가 이미지 전송보다 느린 3개의 버퍼를 나타낸다.

도 97은 디스플레이 리프레시가 이미지 전송보다 빠른 하나의 버퍼를 나타낸다.

도 98은 페이지 체인 및 허브에 의한 호스트-클라이언트 접속을 나타낸다.

도 99는 허브 및 페이지 체인의 조합에 의해 접속된 클라이언트 장치들을 나타낸다.

도 100은 컬러맵을 나타낸다.

도 101은 누설 전류 분석을 나타낸다.

실시예

I. 개요

본 발명의 전체적인 목적은 후술하는 바와 같이 이동 디스플레이 디지털 인터페이스(MDDI)를 제공하는 것이며, 이는 "직렬"형 데이터 링크 또는 채널을 이용하여 호스트 장치와 디스플레이 엘리먼트와 같은 클라이언트 장치 사이의 단거리 통신 링크를 통한 고속 또는 초고속 데이터 전송을 가능하게 하는 비용 효율적이고 전력 소비가 낮은 전송 메커니즘을 이루거나 제공한다. 이 메커니즘은 (하우징이나 지지 프레임의) 내부 디스플레이 엘리먼트나 입력 장치를 중앙 제어기에, 또는 착용 가능한 마이크로 디스플레이(고글이나 프로젝터)와 같은 외부 디스플레이 엘리먼트나 장치를 휴대용 컴퓨터, 무선 통신 장치 또는 엔터테인먼트 장치에 연결하는데 특히 유용한 소형 커넥터 및 얇은 플렉서블 케이블에 의한 구현에 적합하다.

이동 및 디스플레이라는 용어는 프로토콜 명칭과 관련되지만, 이는 편의상 인터페이스 및 프로토콜을 다루는 당업자들에 의해 쉽게 이해되는 표준 명칭을 갖는 것으로만 이해해야 한다. 그러나 많은 비-이동 및 비-디스플레이 관련 애플리케이션이 이 프로토콜 및 결과적인 인터페이스 구조의 적용으로부터 이익을 얻게 되는 후술하는 실시예들의 검토 후 쉽게 이해될 것이며, MDDI 라벨은 본 발명 또는 그 다양한 실시예의 특성이나 유용성에 대한 어떤 제한도 암시하는 것이 아니다.

본 발명의 실시예의 이점은 복잡도가 낮고 비용이 낮으며 신뢰성이 높고 사용 환경에 잘 맞춰지며 매우 확고한 동시에 여전히 매우 융통성 있는 데이터 전송 기술이 제공된다는 점이다.

본 발명의 실시예들은 일반적으로 오디오, 비디오 또는 멀티미디어 애플리케이션을 위한 상당량의 데이터를 이러한 데이터가 생성되거나 저장되는 호스트 또는 소스 장치로부터 클라이언트 디스플레이 또는 프리젠테이션 장치로 고속으로 전달 또는 전송하는 다양한 환경에 사용될 수 있다. 후술하는 통상의 애플리케이션은 휴대용 컴퓨터나 무선 전화 또는 모뎀으로부터 소형 비디오 스크린, 또는 소형 투사 렌즈 및 스크린을 포함하는 고글이나 헬멧 형태의 착용 가능한 마이크로 디스플레이 기기와 같은 시각적 디스플레이 장치로의, 또는 호스트에서 이러한 컴포넌트들 내의 클라이언트 장치로의 데이터 전송이다. 즉, 클라이언트를 이용하는 다양한 내부 또는 외부 입력 장치로부터 내부적으로 위치하는(동일 장치 하우징 또는 지지 구조 내에 배치된) 호스트로, 그리고 프로세서로부터 내부 스크린 또는 다른 프리젠테이션 엘리먼트로의 데이터 전송이다.

MDDI의 특징 또는 속성은 이들이 특정 디스플레이 또는 프리젠테이션 기술에 무관하다는 점이다. 이는 데이터의 내부 구조는 물론 데이터가 구현하는 데이터 또는 명령들의 기능적 양상에도 무관하게 고속으로 데이터를 전송하는 매우 융통성 있는 메커니즘이다. 이는 전송되는 데이터 패킷의 타이밍이 특정 장치 고유의 디스플레이 요구에 대한 것과 같이 특정 클라이언트 장치의 특이성에 맞게, 또는 일부 A-V 시스템에 대한, 또는 조이스틱, 터치패드 등의 특정 입력 장치에 대한 조합된 오디오 및 비디오 요건을 충족하도록 조정될 수 있게 한다. 인터페이스는 선택된 프로토콜이 다음과 같은 한 바로 디스플레이 엘리먼트 또는 클라이언트 장치 불가지론이다. 추가로, 집합 직렬 링크 데이터 또는 데이터 전송률은 통신 시스템 또는 호스트 장치 설계자가 비용, 전력 요건, 클라이언트 장치 복잡도 및 클라이언트 장치 업데이트 속도를 최적화할 수 있게 하는 범위의 여러 배에 걸쳐 달라질 수 있다.

데이터 인터페이스는 주로 "무선" 신호 링크 또는 소형 케이블을 통해 상당량의 고속 데이터를 전송하는데 사용하기 위해 제시된다. 그러나 일부 애플리케이션은 광 기반 링크를 포함하여, 인터페이스 프로토콜을 위해 개발된 동일 패킷 및 데이터 구조를 사용하도록 구성된 무선 링크를 이용할 수 있으며, 전력 소비 또는 복잡도가 여전히 현실적으로 충분히 낮게 원하는 레벨의 전송을 계속할 수 있다.

II. 환경

통상의 애플리케이션은 휴대용 또는 랩탑 컴퓨터(100) 및 무선 전화 또는 PDA 장치(102)가 오디오 재생 시스템(108, 112)과 함께 각각 디스플레이 장치(104, 106)와 데이터를 교환하는 것으로 도시된 도 1a 및 도 1b에서 알 수 있다. 또한, 도 1a는 더 큰 디스플레이나 스크린(114) 또는 이미지 프로젝터(116)에 대한 잠재적 접속을 나타내며, 이는 간결성을 위해 한 도면에만 도시하였지만 무선 장치(102)에도 접속 가능하다. 무선 장치는 현재 데이터를 수신하고 있을 수도 있고 무선 장치의 최종 사용자에게 의한 시청 및/또는 청취를 위한 나중의 프리젠테이션을 위해 메모리 엘리먼트나 장치에 일정량의 멀티미디어 타입 데이터를 미리 저장했을 수도 있다. 통상의 무선 장치는 대부분의 시간에 음성 및 간단한 텍스트 통신에 사용되기 때문에, 장치(102) 사용자에게 정보를 전달하기 위해 오히려 작은 디스플레이 스크린 및 간단한 오디오 시스템(스피커)을 갖는다.

컴퓨터(100)는 훨씬 큰 스크린을 갖지만, 여전히 외부 사운드 시스템에는 부적당하며, 고화질 텔레비전이나 영화 스크린과 같은 다른 멀티미디어 프리젠테이션 장치에 미치지 못한다. 컴퓨터(100)는 설명을 위해 사용된 것이며, 다른 종류의 프로세서, 대화식 비디오 게임 또는 소비자 전자제품이 본 발명에 사용될 수도 있다. 컴퓨터(100)는 이에 한정되는 것은 아니지만 무선 모뎀이나 다른 무선 통신용 내장 장치를 이용할 수도 있고, 또는 원하는 대로 케이블이나 무선 링크를 이용하여 이러한 장치에 접속될 수도 있다.

이는 보다 복잡한 또는 "풍부한" 데이터의 프리젠테이션이 유용한 또는 즐거운 경험에 미치지 않게 한다. 따라서 산업은 최종 사용자들에게 정보를 제시하고 최소 레벨의 원하는 즐거움 또는 긍정적인 체험을 제공하기 위한 다른 메커니즘 및 장치를 개발하고 있다.

상술한 바와 같이, 장치(100)의 최종 사용자들에게 정보를 제시하기 위해 여러 종류의 디스플레이 장치가 개발되어 왔으며 또는 현재 개발중이다. 예를 들어, 하나 이상의 회사가 장치 사용자 눈 앞에 이미지를 투사하여 시각적 디스플레이를 제시하는 착용 가능한 고글 세트를 개발했다. 이러한 장치가 정확하게 배치되면, 시각적 출력을 제공하는 엘리먼트보다 훨씬 큰 가상 이미지를 사용자 눈으로 인식되는 것처럼 "투사"한다. 즉, 매우 작은 투사 엘리먼트가 사용자 눈(들)이 통상의 LCD 스크린 등으로 가능한 것보다 훨씬 큰 규모로 이미지를 "볼 수 있게 한다". 또한, 더 큰 가상 스크린 이미지의 사용은 더욱 한정된 LCD 스크린 디스플레이로 가능한 것보다 훨씬 높은 해상도 이미지의 사용을 가능하게 한다. 다른 디스플레이 장치는 이에 한정되는 것은 아니지만 소형 LCD 스크린 또는 다양한 평판 디스플레이 엘리먼트, 투사 렌즈 및 표면에 이미지를 투사하는 디스플레이 드라이버 등을 포함할 수 있다.

또한, 다른 사용자에게, 또는 다른 경우에 신호를 전송하거나 이를 저장하는 다른 장치에 출력을 제시하기 위해 무선 장치(102) 또는 컴퓨터(100)에 접속된 또는 이들의 사용과 관련된 추가 엘리먼트가 있을 수 있다. 예를 들어, 데이터는 나중에 사용하기 위해 예를 들어 쓰기 가능 CD 매체를 이용하여 플래시 메모리에 광 형태로, 또는 자기 테이프 레코더 및 비슷한 장치에서와 같이 자기 매체에 저장될 수 있다.

또한, 많은 무선 장치 및 컴퓨터는 내장형 MP3 음악 디코딩 성능 및 다른 진보된 사운드 디코더 및 시스템을 갖는다. 휴대용 컴퓨터는 일반적으로 CD 및 DVD 재생 능력을 이용하며, 일부는 미리 녹음된 오디오 파일을 수신하는 소형 전용 플래시 메모리 판독기를 갖는다. 이러한 능력의 결과는 디지털 음악 파일이 매우 증가한 특징의 풍부한 경험을 약속하지만, 디코딩 및 재생 프로세스가 페이스를 유지할 수 있을 경우만이라는 점이다. 이는 디지털 비디오 파일에 대해서도 사실이다.

사운드 재생을 돕기 위해, 도 1a에 외부 스피커(114)가 도시되며, 이는 서브 우퍼나 전방 및 후방 사운드 프로젝션을 위한 "서라운드 음향" 스피커와 같은 추가 엘리먼트가 동반될 수도 있다. 동시에, 스피커 또는 이어폰(108)은 도 1b의 마이크로 디스플레이 장치(106)의 지지 프레임 또는 메커니즘에 내장형으로 전용된다. 알려진 바와 같이, 전력 증폭 또는 사운드 변환 장치를 포함하는 다른 오디오 또는 사운드 재생 엘리먼트가 사용될 수 있다.

어떤 경우에도, 상술한 바와 같이 고품질 또는 고해상도 이미지 데이터 및 고품질 오디오 정보 또는 데이터 신호를 데이터 소스로부터 하나 이상의 통신 링크(110)를 통해 최종 사용자에게 전송하는 것이 바람직할 때, 높은 데이터 전송률이 요구된다. 즉, 현재 전송 메커니즘은 통상적으로 원하는 높은 데이터 전송률을 달성하지 않기 때문에, 전송 링크(100)는 상술한 바와 같이 데이터 통신에 있어 잠재적인 장애임이 명백하며, 시스템 성능을 제한하고 있다. 예를 들어, 상술한 바와 같이, 1024×1024 픽셀과 같이 높은 이미지 해상도에 대해, 픽셀 당 24~32 비트의 색상 심도 및 30 fps의 데이터 전송률로, 데이터 전송률은 755 Mbps 또는 그 이상을 초과하는 속도에 가까워질 수 있다. 또한, 이러한 이미지는 오디오 데이터 및 잠재적으로 대화식 게임 또는 통신을 다루는 추가 신호, 또는 각종 명령, 제어 또는 신호를 포함하는 멀티미디어 프리젠테이션의 일부로서 제시되어, 양 또는 데이터 및 데이터 전송률을 더 증가시킬 수도 있다.

또한, 데이터 링크를 확립하는데 더 적은 케이블 또는 상호 접속이 필요함이 명백하고, 이는 디스플레이 관련 모바일 장치들이 사용이 더 쉽고 더 큰 사용자 베이스에 의해 보다 채택되기 쉬움을 의미한다. 이는 특히 완전한 오디오-시각적 체험을 확립하는데 다수의 장치가 공통으로 사용되는 경우에, 더욱 특히 디스플레이 및 오디오 출력 장치의 품질 레벨이 증가할 때 정확하다.

비디오 화면 및 다른 출력 또는 입력 장치에 있어서의 상기 그리고 다른 많은 개선과 관련된 다른 통상의 애플리케이션은 휴대용 또는 랩탑 컴퓨터(130) 및 무선 전화 또는 PDA 장치(140)가 오디오 재생 시스템(136, 146)과 함께 "내부" 디스플레이 장치(134, 144)와 각각 데이터를 통신하는 것으로 나타낸 도 1c 및 도 1d에서 알 수 있다.

도 1c 및 도 1d에서는, 오늘날 전자 산업 전반에 사용되는 일부 알려진 종류의 회전 조인트와 교차하여 하나 이상의 내부 호스트 및 제어기를 해당 클라이언트를 갖는 비디오 디스플레이 엘리먼트 또는 스크린에 연결하는 일반화된 통신 링크(여기서는 각각 138, 148)를 갖는 장치의 일 부분에서 하나 이상의 내부 호스트 및 제어기의 위치를 나타내기 위해 전체 전자 장치 또는 제품들의 작은 절단면이 사용된다. 이들 전송에 수반되는 데이터량은 링크(138, 148)를 구성하기 위해 상당수의 컨덕터를 필요로 함을 알 수 있다. 병렬형 또는 이러한 데이터를 전송하는데 이용 가능한 다른 공지된 인터페이스 기술 때문에 이러한 장치에 진보된 컬러 및 그래픽 인터페이스, 디스플레이 엘리먼트를 이용하려는 오늘날의 성장하는 요구를 만족시키기 위해 이러한 통신 링크는 90개 또는 그 이상의 컨덕터에 이르고 있는 것으로 추정된다.

공교롭게도, 더욱 높은 데이터 전송률은 단위 시간당 전송되어야 하는 원(raw) 데이터량과 비용 효율적인 물리적 전송 매커니즘의 확실한 제조 면에서 데이터 전송에 이용 가능한 현재의 기술을 능가한다.

프리젠테이션 엘리먼트와 데이터 소스 사이의 데이터 전송 링크 또는 통신 경로에 대해 더욱 고속으로 데이터를 전송하는 기술, 구조, 수단 또는 방법이 필요하며, 이는 시종일관 (보다) 낮은 전력, 경량 및 가능한 한 간단하고 경제적인 케이블 구조를 가능하게 한다. 출원인들은 모바일, 휴대용 또는 심지어 고정 위치 장치들의 어레이가 원하는 디스플레이, 마이크로 디스플레이 또는 오디오 전송 엘리먼트에 원하는 저전력 소비 및 복잡도를 유지하면서 매우 높은 데이터 전송률로 데이터를 전송할 수 있게 하는 상기 및 다른 목적을 달성하는 새로운 기술, 또는 방법 및 장치를 개발해왔다.

III. 고속 디지털 데이터 인터페이스 시스템 구조

새로운 장치 인터페이스를 구성하고 효율적으로 이용하기 위해, 저전력 신호를 이용하여 매우 높은 데이터 전송률을 제공하는 신호 프로토콜 및 시스템 구조가 공식화되었다. 프로토콜은 패킷 및 공통 프레임 구조, 또는 인터페이스에 부과된 명령 또는 동작 구조와 함께 미리 선택된 데이터 또는 데이터 타입을 교환하는 프로토콜을 형성하도록 함께 연결된 구조들을 기반으로 한다.

A. 개요

MDDI 링크에 의해 연결된 또는 MDDI 링크를 통해 통신하는 장치들은 호스트 및 클라이언트라 하며, 클라이언트는 다른 출력 및 입력 장치를 생각할 수도 있지만 통상적으로 어떤 종류의 디스플레이 장치이다. 호스트에서 디스플레이로의 데이터는 (순방향 트래픽 또는 링크라 하는) 순방향으로 이동하고, 클라이언트에서 호스트로의 데이터는 호스트에 의해 인에이블될 때 역방향(역방향 트래픽 또는 링크)으로 이동한다. 이는 도 2에 나타낸 기본 구성으로 설명된다. 도 2에서 호스트(202)는 순방향 링크(208) 및 역방향 링크(210)를 포함하는 것으로 도시된 양방향 통신 채널(206)을 이용하여 클라이언트(204)에 접속된다. 그러나 이들 채널은 순방향 또는 역방향 링크 동작 사이에 데이터 전송이 효율적으로 전환되는 공통 컨덕터 세트에 의해 형성된다. 이는 컨덕터 수를 매우 감소시켜, 모바일 전자 장치에 대한 저전력 환경에서의 고속 데이터 전송에 관한 현재의 접근법이 당면한 많은 문제 중 하나를 즉시 해결한다.

다른 경우에 설명한 바와 같이, 호스트는 본 발명을 이용하여 이익을 얻을 수 있는 여러 종류의 장치 중 하나를 포함한다. 예를 들어, 호스트(202)는 휴대형, 랩탑 또는 비슷한 이동 연산 장치 형태의 휴대용 컴퓨터일 수 있다. 또한, 개인 휴대 단말(PDA), 페이지 장치, 또는 많은 무선 전화 또는 모뎀 중 하나일 수도 있다. 대안으로, 호스트(202)는 휴대용 DVD 또는 CD 플레이어와 같은 휴대용 오락 또는 프리젠테이션 장치, 또는 게임기일 수도 있다.

또한, 호스트는 클라이언트에 고속 통신 링크가 바람직한 다양한 그 밖의 널리 사용되는 또는 계획된 상품에 호스트 장치 또는 제어 엘리먼트로서 존재할 수 있다. 예를 들어, 비디오 레코딩 장치로부터 개선된 응답용 저장 기반 클라이언트로, 또는 프리젠테이션용 고해상도 대형 스크린에 데이터를 고속으로 전송하는데 호스트가 사용될 수 있다. 내장 목록 또는 연산 시스템 및/또는 블루투스 접속을 다른 가정용 장치에 통합한 냉장고 등의 가전제품은 인터넷 또는 블루투스 접속 모드로

동작할 때 디스플레이 성능을 향상시킬 수 있고, 또는 전자 컴퓨터나 제어 시스템(호스트)이 캐비닛에 있는 동안 실내 디스플레이(클라이언트) 및 키보드나 스캐너(클라이언트)에 필요한 배선을 줄일 수 있다. 일반적으로, 당업자들은 새로 추가된 또는 기존의 커넥터나 케이블에 이용 가능한 한정된 개수의 컨덕터를 이용한 보다 고속의 데이터 전송률 이동으로 구형 장치들을 개조하는 능력뿐 아니라 이러한 인터페이스의 사용으로부터 이익을 얻을 수 있는 광범위한 현대 전자 장치 및 가전 제품들을 인식할 것이다.

동시에, 클라이언트(204)는 최종 사용자에게 정보를 제시하거나 사용자로부터 호스트로 정보를 제시하는데 유용한 다양한 장치를 포함할 수 있다. 예를 들어, 고글이나 안경에 통합된 마이크로 디스플레이, 모자나 헬멧에 설치된 투사 장치, 창이나 전면 유리나 같이 차량에 설치된 소형 스크린 또는 심지어 입체 영상 엘리먼트, 또는 각종 스피커, 헤드폰, 또는 고품질 사운드나 음악을 제공하는 사운드 시스템이 있다. 다른 프리젠테이션 장치는 미팅, 또는 영화 및 텔레비전 이미지용 정보를 제공하는데 사용되는 프로젝터나 투사 장치를 포함한다. 다른 예는 장치 또는 시스템 사용자로부터의 상당량의 정보를 사용자로부터의 터치 또는 사운드 외의 거의 실제 "입력"으로 전송할 때 통신을 보낼 수 있는 터치 패드나 감지 장치, 음성 인식 입력 장치, 보안 스캐너 등의 사용이다. 컴퓨터용 도킹 스테이션 및 무선 전화용 카 키트 또는 데스크탑 키트 및 홀더가 최종 사용자 또는 다른 장치 및 기기에 대한 인터페이스 역할을 할 수 있으며, 클라이언트(마우스와 같은 출력 또는 입력 장치)나 호스트를 이용하여 데이터의 전송, 특별히 고속 네트워크가 수신되는 데이터 전송을 돕는다.

그러나 당업자들은 본 발명이 이들 장치에 한정되지 않으며, 저장 및 전송 면에서 또는 재생시 프리젠테이션 면에서 최종 사용자에게 고품질 이미지 및 사운드를 제공하도록 사용이 제안된 시장의 다른 많은 장치가 있음을 쉽게 인지할 것이다. 본 발명은 각종 엘리먼트 또는 장치들 간의 데이터 스트림을 증가시켜 원하는 사용자 체험을 실현하는데 필요한 높은 데이터 전송률을 이루는데 유용하다.

호스트 프로세서, 제어기 또는 회로 구성요소(예를 들어)와 (내부 모드라 하는) 장치 또는 장치 하우징 또는 구조 내의 디스플레이 사이의 상호 접속을 간소화하는데 발명의 MDD 인터페이스 및 통신 신호 프로토콜이 사용되어, 비용 또는 복잡도 및 관련 전력 및 제어 요건 또는 이들 접속의 강제를 감소시키고, (외부 모드라 하는) 외부 엘리먼트, 장치 또는 기기에 대한 접속뿐 아니라, 신뢰성을 향상시킬 수 있다.

이 인터페이스 구조에 의해 사용되는 각 신호 쌍에 대한 집합 직렬 링크 데이터 전송률은 여러 범위에 걸쳐 달라질 수 있으며, 이는 시스템 또는 장치 설계자가 소정 애플리케이션 또는 목적에 대해 비용, 전력, 구현 복잡도 및 디스플레이 업데이트 속도를 쉽게 최적화할 수 있게 한다. MDDI의 속성은 디스플레이나 다른 프리젠테이션 장치(타겟 클라이언트) 기술에 무관하다. 인터페이스를 통해 전송되는 데이터 패킷들의 타이밍은 디스플레이 장치, 사운드 시스템, 메모리 및 제어 엘리먼트와 같은 특정 클라이언트의 특이성, 또는 오디오-비디오 시스템의 조합 타이밍 요건에 맞게 쉽게 조정될 수 있다. 이는 시스템이 매우 적은 전력 소비를 갖게 할 수 있지만, 적어도 어떤 레벨로 MDDI 프로토콜을 사용하기 위해 다양한 클라이언트의 요건이 프레임 버퍼를 갖는 것은 아니다.

B. 인터페이스 타입

MDD 인터페이스는 통신 및 컴퓨터 산업에서 발견된 적어도 4개, 잠재적으로는 그 이상의 다소 명백한 물리적 타입의 인터페이스를 언급하는 것으로 고려된다. 사용되는 특정 애플리케이션 또는 관련된 산업에 따라 당업자들에 의해 다른 라벨이나 표시가 부착될 수도 있지만, 이들은 간단히 타입 1, 타입 2, 타입 3, 타입 4라 한다. 예를 들어, 간단한 오디오 시스템들은 보다 복잡한 멀티미디어 시스템보다 적은 접속을 이용하고, "채널"과 같은 특징들을 다르게 참조할 수 있다.

타입 1 인터페이스는 6-선 또는 다른 종류의 컨덕터 또는 도전 엘리먼트로 구성되며, 이 인터페이스는 이동 또는 무선 전화, PDA, 전자 오락, 및 CD 플레이어나 MP3 플레이어와 같은 휴대용 미디어 플레이어, 및 이와 비슷한 장치나 비슷한 종류의 전자 소비자 기술에 사용되는 장치에 적합하다. 일 실시예에서, 인터페이스는 랩탑, 노트북 또는 데스크탑 개인용 컴퓨터 및 비슷한 장치나 애플리케이션에 더욱 적합하고, 고속 데이터 업데이트를 필요로 하지 않으며, 내장형 MDDI 링크 제어기를 갖지 않는 8-선 (컨덕터) 인터페이스로 구성될 수 있다. 이 인터페이스 타입은 추가 2-선 범용 직렬 버스(USB) 인터페이스의 사용에 의해 구별될 수도 있으며, 이는 기존의 운영 시스템 또는 대부분의 개인용 컴퓨터에서 발견되는 소프트웨어 지원을 제공하는데 매우 유용하다.

타입 2, 타입 3, 타입 4 인터페이스는 고성능 클라이언트 또는 장치에 적합하며, 추가 꼬임 쌍 타입의 컨덕터를 갖는 훨씬 더 복잡한 배선을 이용하여 데이터 신호에 적절한 차폐 및 저손실 전송을 제공한다.

타입 1 인터페이스는 디스플레이, 오디오, 제어 및 한정된 시그널링 정보를 포함할 수 있는 신호들을 전달하고, 통상적으로 고해상도 고속 비디오 데이터를 필요로 하지 않는 모바일 클라이언트나 클라이언트 장치에 사용된다. 타입 1 인터페이스

스는 30 fps 및 5.1 채널 오디오의 SVGA 해상도를 쉽게 지원할 수 있으며, 최소한의 구성에서는 단지 총 3쌍의 배선, 즉 데이터 전송용 2쌍 및 전력 전송용 1쌍만을 이용할 수 있다. 이러한 타입의 인터페이스는 주로 이동 무선 장치와 같은 장치를 위한 것이며, USB 호스트는 통상적으로 신호 접속 및 전송용 장치 내에서 이용 가능하지 않다. 이러한 구성에서, 이동 무선 장치는 MDDI 호스트 장치이며, 호스트로부터의 통신 링크를 제어하는 "마스터" 역할을 하고, 일반적으로 프리젠테이션, 디스플레이 또는 재생을 위해 클라이언트(순방향 트래픽 또는 링크)에 데이터를 전송한다.

이 인터페이스에서, 호스트는 지정된 지속기간 동안 버스(링크)를 통해 이루어지고 역방향 패킷으로서 호스트에 데이터를 전송할 수 있게 하는 클라이언트에 특별한 명령 또는 패킷 타입을 전송함으로써 클라이언트로부터(역방향 트래픽 또는 링크) 호스트에서의 통신 데이터 수신을 가능하게 한다. 이는 도 3에 도시되어 있으며, (후술하는) 암호화 패킷이라 하는 패킷 종류가 사용되어 전송 링크를 통한 역방향 패킷 전송을 제공하여 역방향 링크를 형성한다. 데이터에 대한 클라이언트를 폴링하기 위해 호스트에 할당된 시간 간격은 호스트에 의해 미리 결정되며, 각 지정된 애플리케이션의 요건에 기반한다. 클라이언트로부터의 정보 또는 데이터 전송에 USB 포트를 이용할 수 없는 이러한 종류의 반이중 양방향 데이터 전송이 특별히 유리하다.

HDTV형 또는 비슷한 고해상도가 가능한 고성능 디스플레이는 풀 모션 비디오를 지원하기 위해 약 1.5 Gbps 속도의 데이터 스트림을 필요로 한다. 타입 2 인터페이스는 병렬로 2비트를 전송함으로써 고속 데이터 전송률을 지원하고, 타입 3 인터페이스는 병렬로 4비트를 전송함으로써, 타입 4 인터페이스는 8비트를 병렬로 전송한다. 타입 2 및 타입 3 인터페이스는 타입 1과 동일한 케이블 및 커넥터를 사용하지만, 2배 및 4배의 데이터 전송률로 동작하여 휴대용 장치에 보다 고성능의 비디오 애플리케이션을 지원할 수 있다. 타입 4 인터페이스는 초고성능 클라이언트 또는 디스플레이에 적합하며, 추가 꼬임 쌍 데이터 신호를 포함하는 약간 더 큰 케이블을 필요로 한다.

MDDI에 의해 사용되는 프로토콜은 사용될 수 있는 가능한 최고 데이터 전송률이 얼마인지를 협의함으로써 각각 타입 1, 2, 3 또는 4 호스트가 일반적으로 타입 1, 2, 3 또는 4 클라이언트와 통신할 수 있게 한다. 최소 가능 장치라 할 수 있는 것의 용량 또는 유효 특징은 링크 성능을 설정하는데 사용된다. 원칙적으로, 호스트 및 클라이언트 모두 타입 2, 타입 3 또는 타입 4 인터페이스를 이용할 수 있는 시스템이라 하더라도, 모두 타입 1 인터페이스로서 동작을 시작한다. 이어서 호스트는 타겟 클라이언트의 용량을 결정하고, 특정 애플리케이션에 적절하게 타입 2, 타입 3 또는 타입 4 모드에 대한 핸드오프 또는 재구성 동작을 협의한다.

일반적으로, 호스트는 (후술하는) 적절한 링크 계층 프로토콜을 사용하고 일반적으로 임의의 시간에 전력을 절약하기 위한 저속 모드로 전력을 낮추거나 다시 동작을 재구성하고, 또는 더 높은 해상도의 디스플레이 콘텐츠에 대한 보다 고속의 전송을 지원하는 고속 모드로 전력을 높일 수 있다. 예를 들어, 호스트는 시스템이 배터리와 같은 전원에서부터 AC 전력으로 전환될 때, 또는 디스플레이 매체의 소스가 낮은 또는 높은 해상도의 포맷으로 전환될 때 인터페이스 타입을 변경할 수도 있고, 또는 이러한 또는 다른 조건이나 이벤트의 조합을 인터페이스 타입 또는 전송 모드 변경의 기초로 간주할 수도 있다.

또한, 시스템은 어떤 방향의 한 모드 및 다른 방향의 다른 모드를 이용하여 데이터를 전달할 수 있다. 예를 들어, 고속으로 디스플레이에 데이터를 전송하는데 타입 4 인터페이스 모드가 사용될 수 있는 한편, 키보드나 포인팅 장치와 같은 주변 장치들로부터 호스트 장치로 데이터 전송시 타입 1 모드가 사용된다. 당업자들은 호스트 및 클라이언트가 다른 속도로 나가는 데이터를 전달할 수 있는 것으로 인식할 것이다.

흔히, MDDI 프로토콜의 사용자들은 "외부" 모드 및 "내부" 모드를 구별할 수 있다. 외부 모드는 한 장치의 호스트를 그 장치의 약 2미터까지 외부의 또는 호스트로부터 그 정도까지의 클라이언트에 접속하기 위한 프로토콜 및 인터페이스의 사용을 기술한다. 이러한 상황에서, 호스트는 외부 클라이언트에 전력을 전송하여 두 장치가 모바일 환경에서 쉽게 동작할 수 있다. 내부 모드는 호스트가 공통 하우징이나 지지 프레임 또는 동일 종류의 구조 내에서와 같이 동일 장치 내에 포함된 클라이언트에 접속되는 경우를 기술한다. 예로서 클라이언트가 디스플레이나 디스플레이 드라이버, 또는 키패드나 터치 패드와 같은 입력 장치, 또는 사운드 시스템이고, 호스트가 중앙 제어기, 그래픽 엔진 또는 CPU 엘리먼트인 무선 전화나 다른 무선 장치, 또는 휴대용 컴퓨터나 게임기 내의 애플리케이션이 있다. 외부 모드 애플리케이션과 달리 클라이언트는 내부 모드 애플리케이션의 호스트에 훨씬 가깝게 배치되기 때문에, 일반적으로 이러한 구성에서 클라이언트에 대한 전력 접속에 관해 논의되는 요건이 없다.

C. 물리적 인터페이스 구조

호스트와 클라이언트 장치 사이의 통신을 확립하기 위한 장치 또는 링크 제어기의 일반적인 배치는 도 4 및 도 5에 도시된다. 도 4 및 도 5에서 MDDI 링크 제어기(402, 502)는 호스트 장치(202)에 설치된 것으로 도시되고, MDDI 링크 제어기(404, 504)는 클라이언트 장치(204)에 설치된 것으로 도시된다. 상기와 같이, 호스트(202)는 일련의 컨덕터를 포함하는

양방향 통신 채널(406)을 이용하여 클라이언트(204)에 접속된다. 후술하는 바와 같이, 호스트 및 클라이언트 링크 제어기 모두 호스트 제어기(드라이버)나 클라이언트 제어기(수신기)로서 응답하도록 설정, 조정 또는 프로그래밍 될 수 있는 단일 회로 설계를 이용한 집적 회로로서 제조될 수 있다. 이는 보다 큰 스케일의 단일 회로 장치 제조로 인해 보다 저렴한 비용을 제공한다.

도 5에서, MDDI 링크 제어기(502)는 호스트 장치(202')에 설치된 것으로 도시되고, MDDI 링크 제어기(504)는 클라이언트 장치(204')에 설치된 것으로 도시된다. 상기와 같이, 호스트(202')는 일련의 컨덕터를 포함하는 양방향 통신 채널(506)을 이용하여 클라이언트(204')에 접속된다. 상술한 바와 같이, 호스트 및 클라이언트 링크 제어기 모두 단일 회로 설계를 이용하여 제조될 수 있다.

MDDI 링크 또는 사용되는 물리적 컨덕터를 통해 호스트와 디스플레이 장치와 같은 클라이언트 사이에 전달되는 신호 또한 도 4 및 도 5에 도시된다. 도 4 및 도 5에 나타난 바와 같이, MDDI를 통한 데이터 전송의 주 경로 또는 메커니즘은 MDDI_Data0+/- 및 MDDI_Stb+/-로 표기한 데이터 신호를 이용한다. 이들 각각은 케이블의 다른 배선 쌍을 통해 전송되는 저전압 데이터 신호이다. 인터페이스를 통해 전송되는 각 비트에 대한 MDDI_Data0 쌍 또는 MDDI_Stb 쌍에 한 번의 전이만이 있다. 이는 전류 기반이 아니라 전압 기반 전송 메커니즘이므로, 정적 전류 소비는 거의 0이다. 호스트는 클라이언트 디스플레이에 대해 MDDI_Stb 신호를 구동한다.

데이터는 MDDI_Data 쌍을 통해 순방향 및 역방향으로 흐르는 동안, 즉 이는 양방향 전송 경로인 동안 호스트는 데이터 링크의 마스터 또는 제어기이다. MDDI_Data0 및 MDDI_Stb 신호 경로는 다른 모드로 동작하여 잡음 면역성을 최대화한다. 이들 라인 상의 신호의 데이터 전송률은 호스트에 의해 전송되는 클록의 속도에 의해 결정되고, 약 1 kbps 내지 400 Mbps 이상의 범위에 걸쳐 변화한다.

타입 2 인터페이스는 타입 1 이상의 하나의 추가 데이터 쌍 또는 컨덕터 또는 경로를 포함하며, 이는 MDDI_Data1+/-라 한다. 타입 3 인터페이스는 타입 2 인터페이스 이상의 2개의 추가 데이터 쌍 또는 신호 경로를 포함하며, 이는 MDDI_Data2+/- 및 MDDI_Data3+/-라 한다. 타입 4 인터페이스는 타입 3 인터페이스 이상의 4개 이상의 데이터 쌍 또는 신호 경로를 포함하며, 이는 각각 MDDI_Data4+/-, MDDI_Data5+/-, MDDI_Data6+/- 및 MDDI_Data7+/-라 한다. 상기 각각의 인터페이스 구성에서, 호스트는 HOST_Pwr 및 HOST_Gnd로 지정된 배선 쌍 또는 신호를 이용하여 클라이언트 또는 디스플레이에 전력을 전송할 수 있다. 후술하는 바와 같이, 전력 전송은 다른 모드에 이용 가능하거나 제공되는 것보다 적은 컨덕터를 사용하는 인터페이스 "타입"이 사용되고 있을 경우, MDDI_Data4+/-, MDDI_Data5+/-, MDDI_Data6+/- 또는 MDDI_Data7+/- 컨덕터에 대한 어떤 구성에서도 원하는 대로 조절될 수도 있다. 이 전력 전송은 일반적으로 외부 모드에 대해 이용되며, 일부 애플리케이션은 다를 수도 있지만 일반적으로 내부 모드에는 불필요하다.

각종 모드에 대한 MDDI 링크를 통해 호스트와 클라이언트(디스플레이) 사이에 전달되는 신호의 요약은 인터페이스 타입에 따라 하기의 표 1에서 설명한다.

[표 1]

타입 1	타입 2	타입 3	타입 4
HOST_Pwr/Gnd MDDI_Stb+/- MDDI_Data0+/-	HOST_Pwr/Gnd MDDI_Stb+/- MDDI_Data0+/- MDDI_Data1+/-	HOST_Pwr/Gnd MDDI_Stb+/- MDDI_Data0+/- MDDI_Data1+/- MDDI_Data2+/- MDDI_Data3+/-	HOST_Pwr/Gnd MDDI_Stb+/- MDDI_Data0+/- MDDI_Data1+/- MDDI_Data2+/- MDDI_Data3+/- MDDI_Data4+/- MDDI_Data5+/- MDDI_Data6+/- MDDI_Data7+/-
선택적 전력	선택적 전력	선택적 전력	선택적 전력
선택적 전력	선택적 전력	선택적 전력	선택적 전력
선택적 전력	선택적 전력	선택적 전력	선택적 전력
선택적 전력	선택적 전력	선택적 전력	선택적 전력

또한, 호스트로부터의 전송을 위한 HOST_Pwr/Gnd 접속은 일반적으로 외부 모드에 제공됨에 유의한다. 내부 애플리케이션 또는 동작 모드는 일반적으로 다른 내부 자원으로부터 직접 전력을 끌어오는 클라이언트를 가지며, 당업자로부터 명백하듯이, 전력 분포를 제어하기 위해 MDDI를 사용하지 않으므로 이러한 분포는 여기서 더 상세히 설명하지 않는다. 그러나 MDDI 인터페이스를 통해 분포되는 전력은 예를 들어 당업자에 의해 이해되는 바와 같이 특정 종류의 전력 제어, 동기화 또는 상호 접속 편의를 고려할 수 있게 하는 것이 확실히 가능하다.

상기 구조 및 동작을 구현하는데 일반적으로 사용되는 케이블은 명목상 길이가 1.5미터 정도, 일반적으로는 2미터 이하이고, 3쌍의 꼬임 쌍 컨덕터를 포함하며, 각각 다중 꼬임 30 AWG 배선이다. 3개의 꼬임 상 위에 추가 드레인 배선으로서 박막 실드 커버팅이 싸이거나 달리 말하면 형성된다. 꼬임 쌍 및 실드 드레인 컨덕터는 클라이언트 커넥터에서 클라이언트에 대한 실드에 접속된 실드로 마무리되며, 공지된 바와 같이 전체 케이블을 커버하는 절연 층이 있다. 배선은 HOST_Gnd 및 HOST_Pwr; MDDI_Stb+ 및 MDDI_Stb-; MDDI_Data0+ 및 MDDI_Data0-; MDDI_Data1+ 및 MDDI_Data1- 등으로 쌍을 이룬다. 그러나, 다양한 도체들 및 케이블은 특정 응용들에 따라 본 발명의 실시예들을 구현하기 위하여 종래에 이해되는 바와같이 사용될 수 있다. 예컨대, 무거운 외측 코팅들 또는 균일한 금속 층들은 일부 응용들에서 케이블을 보호하기 위하여 사용될 수 있는 반면에, 보다 얇고 평평한 도체 리본형 구조들은 다른 응용들에 적합할 수 있다.

D. 데이터 타입 및 전송률

전 범위의 사용자 체험 및 애플리케이션에 대해 유용한 인터페이스를 달성하기 위해, 이동 디지털 데이터 인터페이스(MDDI)는 제어 정보와 함께 이동 디스플레이 장치에 집적 또는 이와 협력하여 동작할 수 있는 다양한 클라이언트 및 디스플레이 정보, 오디오 트랜스듀서, 키보드, 포인팅 장치 및 많은 다른 입력 및 출력 장치, 및 이들의 조합에 대한 지원을 제공한다. MDD 인터페이스는 최소한의 케이블 또는 컨덕터를 사용하여 순방향 또는 역방향 링크로 호스트와 클라이언트 사이에 이동하는 데이터 스트림의 다양한 잠재적 타입을 제공할 수 있도록 설계된다. 등시성 스트림 및 비동기 스트림(업데이트) 모두 지원된다. 집합적 데이터 전송률이 바람직한 최대 MDDI 링크 속도보다 작거나 같은 한 여러 조합의 데이터 타입이 가능하며, 이는 최대 직렬 레이트 및 사용되는 데이터 에어 수에 의해 한정된다. 이들은 하기의 표 2 및 3에 기재된 아이템들을 포함하지만, 이에 한정되는 것은 아니다.

[표 2]

호스트로부터 클라이언트로의 전송		
등시성 비디오 데이터	702×480, 12비트, 30f/s	~ 124.5Mbps
등시성 스테레오 오디오 데이터	44.1kHz, 16비트, 스테레오	~ 1.4Mbps
비동기 그래픽 데이터	800×600, 12비트, 10f/s, 스테레오	~ 115.2Mbps
비동기 제어	최소	≪ 1.0Mbps

[표 3]

클라이언트로부터 호스트로의 전송		
등시성 음성 데이터	8kHz, 8비트	≪ 1.0Mbps
등시성 비디오 데이터	640×480, 12비트, 24f/s	~ 88.5Mbps
비동기 상태, 사용자 입력 등	최소	≪ 1.0Mbps

인터페이스는 고정되는 것이 아니라, 미래의 시스템 유연성을 위해 사용자 정의 데이터를 포함하는 다양한 정보 "타입"의 전송을 지원할 수 있도록 확장 가능하다. 제공되는 데이터의 특정 예는 전체 또는 일부 스크린 비트맵 필드나 압축 비디오 형태의 풀 모션 비디오; 전력을 보존하고 구현 비용을 줄이기 위한 저속의 정적 비트맵; 다양한 해상도 또는 전송률의 PCM 또는 압축 오디오 데이터; 포인팅 장치 위치 및 선택, 및 아직 정의되지 않은 용량에 대한 사용자 정의 가능 데이터이다. 이러한 데이터는 제어 또는 상태 정보와 함께 전송되어 장치 용량 또는 설정된 작동 파라미터를 검출할 수 있다.

본 발명의 실시예들은 이에 한정되는 것은 아니지만 영화(비디오 디스플레이 및 오디오) 시청; 제한된 개인용 뷰잉(viewing)(그래픽, 디스플레이, 때때로 비디오 및 오디오의 조합)을 갖는 개인용 컴퓨터의 이용; PC, 콘솔 또는 개인용 장치상에서의 비디오 게임(동영상 디스플레이 또는 종합 비디오 및 오디오); 비디오폰(양방향 저속 비디오 및 오디오), 스틸 디지털 사진용 카메라 또는 디지털 비디오 이미지 캡처용 캠코더 형태의 장치를 이용한 인터넷 "서핑"; 프리젠테이션을 제공하기 위해 프로젝터로 도킹되는 또는 비디오 모니터, 키보드 및 마우스에 접속된 데스크탑 도킹 스테이션으로 도킹된 전화, 컴퓨터 또는 PDA의 사용; 및 무선 포인팅 장치 및 키보드 데이터를 포함하여, 셀룰러 폰, 스마트 폰 또는 PDA를 사용한 생산성 향상 또는 엔터테인먼트를 포함하는 데이터 전송에 이용되는 기술을 향상시킨다.

후술하는 고속 데이터 인터페이스는 일반적으로 배선 또는 케이블형 링크로서 구성되는 통신 또는 전송 링크를 통해 상당량의 A-V 타입 데이터를 제공하는 것에 관하여 제시된다. 그러나 신호 구조, 프로토콜, 타이밍 또는 전송 메커니즘은 원하는 레벨의 데이터 전송을 유지할 수 있다면 광 또는 무선 매체 형태로 링크를 제공하도록 조정될 수 있음이 명백할 것이다.

MDD 인터페이스 신호들은 기본 신호 프로토콜 또는 구조에 대한 공통 프레임 레이트(CFR)로서 공지된 개념을 이용한다. 공통 프레임 레이트의 사용 뒤의 생각은 동시 등시성 데이터 스트림에 동기화 펄스를 제공하는 것이다. 클라이언트 장치는 이 공통 프레임 레이트를 시간 기준으로 사용할 수 있다. 낮은 CF 레이트는 서브 프레임 헤더를 전송하기 위해 오버헤드를 감소시킴으로써 채널 효율을 향상시킨다. 한편, 높은 CF 레이트는 잠복 기간을 감소시키고, 오디오 샘플에 대해 보다 작은 유연한 데이터 버퍼를 허용한다. 본 발명의 인터페이스의 CF 레이트는 동적으로 프로그래밍 가능하며, 특정 애플리케이션에 사용되는 등시성 스트림에 적절한 많은 값 중 하나로 설정될 수 있다. 즉, CF 값은 원하는 대로 소정 클라이언트 및 호스트 구성에 가장 적합하게 선택된다.

비디오 또는 마이크로 디스플레이와 같은 애플리케이션에 가장 사용하기 쉬운 등시성 데이터 스트림에 대해 조정 가능한 또는 프로그래밍 가능한 서브 프레임당 일반적으로 필요한 바이트 수는 표 4에 나타낸다.

[표 4]

공통 프레임 레이트(CFR) = 300Hz							
	X	Y	비트	프레임 레이트	채널	레이트 (Mbps)	바이트/서브 프레임
컴퓨터 게임	720	480	24	10	1	248.832	103680
컴퓨터 그래픽	800	600	24	10	1	115.200	48000
비디오	640	480	12	29.9710 또는 30	1	221.184	92160
CD 오디오	1	1	16	44100	2	1.4112	588
음성	1	1	8	8000	1	0.064	26-2/3

서브 프레임당 바이트의 단편적인 카운트는 간단한 프로그래밍 가능 M/N 카운터 구조를 이용하여 쉽게 얻어진다. 예를 들어, 26 바이트의 한 서브 프레임에 각각 이어지는 27 바이트의 2개의 프레임을 전송함으로써 CF 당 26-2/3 바이트의 카운트가 구현된다. 더 낮은 CF 전송률이 선택되어 서브 프레임당 정수 개의 바이트를 생성할 수 있다. 그러나 일반적으로, 하드웨어에 간단한 M/N 카운터를 구현하는 것은 더 큰 오디오 샘플 FIFO 버퍼에 필요한 영역보다 본 발명의 일부 또는 모든 실시예를 구현하는데 사용되는 집적 회로 칩 또는 전자 모듈 내에 보다 적은 영역을 요구해야 한다.

다른 데이터 전송률 및 데이터 타입의 영향을 설명하는 전형적인 애플리케이션은 가라오케 시스템이다. 가라오케 시스템은 최종 사용자 또는 사용자들이 뮤직 비디오 프로그램에 따라 노래하는 시스템이다. 노래의 가사가 통상적으로 스크린 밑 어딘가에 표시되어, 사용자는 불러야 할 단어 및 대충 노래의 타이밍을 알게 된다. 이 출원은 드문드문 그래픽이 업데이트 되고, 사용자의 음성 또는 음성들이 스테레오 오디오 스트림과 혼합되는 비디오 디스플레이를 필요로 한다.

300 Hz의 공통 프레임 레이트를 가정하면, 각 서브 프레임은 클라이언트 디스플레이 장치에 대한 순방향 링크에서 (스테레오에서 147개의 16 비트 샘플에 기반한) 92,160 바이트의 비디오 콘텐츠 및 588 바이트의 오디오 콘텐츠로 구성될 것이며, 평균 26.67(26-2/3) 바이트의 음성이 마이크에서 모바일 노래방 기계로 다시 전송된다. 비동기 패킷이 호스트와 어쨌든 헤드 장착된 디스플레이 사이에 전송된다. 이는 기껏해야 768 바이트의 그래픽 데이터(1/4 스크린 높이) 및 잡다한 제어 및 상태 명령들에 대해 200 바이트(여러) 미만의 바이트를 포함한다.

표 5는 가라오케 예의 경우 서브 프레임 내에 데이터가 어떻게 할당되는지를 보여준다. 사용되는 총 레이트는 약 279 Mbps로 선택된다. 280 Mbps의 약간 높은 레이트는 서브 프레임당 약 400 바이트의 데이터가 전송될 수 있게 하고, 이는 임시 제어 및 상태 메시지의 사용을 가능하게 한다.

[표 5]

엘리먼트 레이트	서브 프레임당 오버헤드 바이트	서브 프레임당 미디어 바이트
640×480픽셀 및 30fps의 뮤직비디오	2*28 = 56	92160
10 서브프레임, 1/30초로 업데이트되는 640×120픽셀 및 1fps의 가사 텍스트	28	768
44,100sps, 스테레오, 16비트의 CD 오디오	2*16 = 32	588
8000sps, 모노, 8비트의 음성	28+8+8+(4*16)+(3*27) = 125	26.67
서브 프레임 헤더	22	
총 바이트/CF	263	115815

총 레이트(Mbps)

(263+115815)*8*300 = 278.5872

III.(계속) 고속 디지털 데이터 인터페이스 시스템 구조

E. 링크 계층

MDD 인터페이스 고속 직렬 데이터 신호를 이용하여 전송되는 데이터는 차례로 연결된 시간 멀티플렉싱 패킷 스트림으로 구성된다. 전송 장치가 전송할 데이터가 없을 때에도, MDDI 링크 제어기는 일반적으로 채움 패킷을 자동으로 전송하여 패킷 스트림을 유지한다. 간단한 패킷 구조의 사용은 비디오 및 오디오 신호 또는 데이터 스트림에 대해 신뢰할 수 있는 등시성 타이밍을 확보한다.

패킷 그룹은 서브 프레임이라 하는 신호 엘리먼트 또는 구조 내에 포함되며, 서브 프레임 그룹은 미디어 프레임이라 하는 신호 엘리먼트 또는 구조 내에 포함된다. 서브 프레임은 각각의 크기 및 데이터 전송 용도에 따라 하나 이상의 패킷을 포함하고, 미디어 프레임은 하나 이상의 서브 프레임을 포함한다. 본원의 실시예에 의해 이용되는 프로토콜에 의해 제공되는 가장 큰 서브 프레임은 $2^{32}-1$ 또는 4,294,967,295 바이트 정도이므로, 가장 큰 미디어 프레임 크기는 $2^{16}-1$ 또는 65,535 개 정도의 서브 프레임이 된다.

특별한 서브 프레임 헤더 패킷은 후술하는 바와 같이 각 서브 프레임 시작에 나타나는 고유 식별자를 포함한다. 그 식별자는 호스트와 클라이언트가 시작할 때 클라이언트 장치에서 프레임 타이밍을 획득하는데 사용된다. 링크 타이밍 획득은 뒤에 보다 상세히 설명한다.

통상적으로, 디스플레이 스크린은 풀 모션 비디오가 디스플레이되고 있을 때 미디어 프레임당 한 번 업데이트된다. 디스플레이 프레임 레이트는 미디어 프레임 레이트와 동일하다. 링크 프로토콜은 원하는 애플리케이션에 따라 전체 디스플레이, 또는 단지 정적 이미지에 의해 둘러싸인 작은 영역의 풀 모션 비디오 콘텐츠에 대해 풀 모션 비디오를 지원한다. 웹페이지 또는 이메일 보기 등의 일부 저전력 모바일 애플리케이션에서, 디스플레이 스크린은 이따금 업데이트될 필요만 있다. 이러한 상황에서, 단일 서브 프레임을 전송한 다음 링크를 쉼다운하거나 비활성화하여 전력 소비를 최소화하는 것이 유리하다. 또한, 인터페이스는 스테레오 비전과 같은 효과를 지원하고, 그래픽 원색을 처리한다.

서브 프레임은 시스템이 높은 우선순위의 패킷을 주기적으로 전송할 수 있게 한다. 이는 동시 등시성 스트림이 최소량의 데이터 버퍼링으로 공존할 수 있게 한다. 이는 디스플레이 프로세스에 제공되는 한 가지 이점이며, 다중 데이터 스트림(비디오, 음성, 제어, 상태, 포인팅 장치 데이터 등의 고속 통신)이 본질적으로 공통 채널을 공유할 수 있게 한다. 또한, 수평 동기 펄스 및 CRT 모니터에 대한 블랭크 간격과 같은 디스플레이 기술 특유의 동작이나 다른 클라이언트 기술 특유의 동작이 존재하게 한다.

F. 링크 제어기

도 4 및 도 5에 나타난 MDDI 링크 제어기가 MDDI 데이터 및 스트로브 신호를 수신하는데 사용되는 차동 라인 수신기를 제외하고 완전히 디지털로 구현되도록 제조 또는 조립된다. 그러나 차동 라인 드라이버 및 수신기들도 CMOS형 IC를 제작하는 경우와 같이 링크 제어기를 갖는 동일한 디지털 집적 회로에 구현될 수 있다. 비트 복원에 또는 링크 제어기용 하드웨어 구현에 어떤 아날로그 기능 또는 위상 고정 루프(PLL)도 요구되지 않는다. 호스트 및 클라이언트 링크 제어기는 링크 동기화를 위한 상태 머신을 포함하는 클라이언트 인터페이스를 제외하고, 매우 비슷한 기능들을 포함한다. 따라서, 본 발명의 실시예는 전체적으로 링크 제어를 위해 제조 비용을 감소할 수 있는 호스트 또는 클라이언트로 구성될 수 있는 단일 제어 설계 또는 회로를 생성할 있는 실질적인 장점을 갖게 한다.

IV. 인터페이스 링크 프로토콜

A. 프레임 구조

패킷 전송을 위한 순방향 링크 통신을 구현하는데 사용되는 신호 프로토콜 또는 프레임 구조가 도 6에 도시된다. 도 6에 도시된 바와 같이, 정보 또는 디지털 데이터는 패킷으로 알려진 구성 요소로 그룹화된다. 멀티미디어 패킷은 "서브 프레임"으로 알려진 것을 형성하기 위해 차례로 서로 그룹화되며, 다수의 서브 프레임은 "미디어" 프레임을 구성하기 위해 차례로 서로 그룹화된다. 프레임의 형성 및 서브 프레임의 전송을 제어하기 위해, 각각의 서브 프레임은 서브 프레임 헤더 패킷(SHP)으로 알려진 특정하게 한정된 패킷으로 시작한다.

호스트 장치는 소정의 전송을 위해 사용되는 데이터 레이트를 선택한다. 이러한 레이트는 호스트의 최대 전송 성능, 또는 호스트에 의해 스스로부터 검색된 데이터, 및 클라이언트의 최대 성능 또는 데이터가 전송될 다른 장치에 기초하여 호스트 장치에 의해 동적으로 변경될 수 있다.

MDDI 또는 본 발명의 신호 프로토콜과 동작하도록 설계되거나, 이용가능한 수신 클라이언트 장치는 사용할 수 있는 최대 또는 현재의 최대 데이터 레이트를 결정하기 위해 호스트에 의해 질의될 수 있거나, 디폴트 슬로워 최소 레이트가 사용 가능한 데이터 타입 및 지지된 특성과 함께 사용될 수도 있다. 이러한 정보는 후술하는 바와 같이, 클라이언트 성능 패킷(DCP)을 사용하여 전송될 수 있다. 클라이언트 디스플레이 장치는 사전 선택된 최소 데이터 레이트 또는 최소 데이터 레이트 영역 내에서 데이터를 전송하거나 인터페이스를 사용하는 다른 장치와 통신할 수 있으며, 호스트는 클라이언트 장치의 전체 성능을 결정하기 위해 이러한 영역 내에서 데이터 레이트를 사용하는 질의를 할 수 있다.

비트맵의 특성 및 클라이언트의 비디오 프레임 레이트 성능을 한정하는 다른 상태 정보는 호스트가 실제로 유효하거나 선택적인, 또는 소정의 시스템 제한 내에서 요구되는 인터페이스를 구성할 수 있도록 상태 패킷으로 호스트로 전송될 수 있다.

호스트는, 현재 서브 프레임에서 전송될 데이터 패킷이 없는(더 많은) 경우, 또는 호스트가 순방향 링크에 대해 선택된 데이터 송신 레이트와 보조를 맞추기에 충분할 레이트로 전송할 수 없는 경우, 필러 패킷을 전송한다. 각각의 서브 프레임이 서브 프레임 헤더 패킷으로 시작하기 때문에, 앞선 서브 프레임의 마지막은 앞선 서브 프레임을 완전히 메우는 패킷(필러 패킷과 거의 유사)을 포함한다. 그 자체로 패킷을 갖는 데이터에 대한 공간이 부족한 경우, 필러 패킷은 서브 프레임의 또는 다음의 앞선 서브 프레임의 말단 및 서브 프레임 헤더 패킷 이전의 최종 패킷과 거의 유사할 것이다. 상기 서브 프레임 내에서 송신될 각각의 패킷에 대한 서브 프레임에 잔여하는 충분한 공간이 있다는 것을 보장하는 것은 호스트 장치에서의 제어 연산의 작업이다. 동시에, 일단 호스트 장치가 데이터 패킷의 전송을 개시하면, 호스트는 데이터 언더-런 조건을 유발하지 않고 프레임 내의 크기를 갖는 패킷을 연속하여 완료할 수 있다.

실시예의 일 특징에서, 서브 프레임 송신은 두 모드를 갖는다. 하나의 모드는 라이브 비디오 및 오디오 스트림을 송신하는 데 사용되는 주기적인 서브 프레임 모드이거나, 주기적인 타이밍 예폭이다. 이러한 모드에서, 서브 프레임 길이는 0이 아닌 것으로 한정된다. 두 번째 모드는 비동기 또는 비주기 모드로서, 프레임은 새로운 정보가 이용가능할 때, 비트맵 데이터를 클라이언트로 제공하는데 사용된다. 이러한 모드는 서브 프레임 헤더 패킷에서 서브 프레임 길이를 0으로 설정함으로써 한정된다. 주기적인 모드를 사용할 때, 서브 프레임 패킷 수신은 클라이언트가 순방향 링크 프레임 구조와 동기화된 때 개시할 수도 있다. 이는 도49 또는 도63과 관련하여 후술되는 상태 다이어그램에 따라 한정된 "in sync" 상태에 대응한다. 비동기 비주기 서브 프레임 모드에서, 수신은 제1 서브 프레임 헤더 패킷이 수신된 후 개시한다.

B. 전체 패킷 구조

실시예에 의해 구현되는 통신 또는 신호 프로토콜, 또는 데이터를 송신하기 위한 방법 및 수단을 구성하는데 사용되는 패킷의 형식 또는 구조가 이하에 설명되는데, 인터페이스는 확장가능하고 추가의 패킷 구조는 원하는 대로 부가될 수 있음을 이해해야 한다. 패킷은 인터페이스에서 이들의 기능에 따라 다양한 "패킷 타입", 즉 명령, 정보, 값, 또는 송신하거나 관련된 데이터로 분류되거나 나뉜다. 따라서, 각각의 패킷 타입은 전송될 패킷 및 데이터를 조작하는데 사용되는 소정의 패킷에 대한 예정된 패킷 구조를 나타낸다. 명백하듯이, 패킷은 예정된 길이를 갖거나, 이들 각각의 기능에 따라 가변 또는 동적 변화가능한 길이를 갖는다. 또한, 비록 동일한 기능이 실현되지만 표준으로의 수신 동안 프로토콜이 변화할 경우 발생할 수 있는 바와 같이, 패킷은 상이한 이름을 가질 수 있다. 다양한 패킷에 사용된 바이트들 또는 바이트 값은 부호 없는 다중 비트 정수(8 또는 16비트)로 구성된다. 타입 오더에 리스트된 "타입" 설계와 함께 이용되는 패킷의 요약이 표 VI-1 내지 VI-4에 도시된다.

각각의 표는 예시 또는 이해의 용이함을 위해 전체 패킷 구조 내에서 패킷의 일반 "타입"을 나타낸다. 이러한 그룹에 의해 본원 발명을 설명하기 위해 부과된 제한은 없으며, 패킷은 원하는 다양한 형태로 구성될 수 있다. 패킷의 전송이 유효한 것으로 고려되는 지침이 또한 표시된다.

표 VI-1: 링크 제어 패킷들

패킷 이름	패킷 타입	순방향에서 유효성	역방향에서 유효성
서브 프레임 헤더 패킷	15359	×	
필러 패킷	0	×	×

역방향 링크 캡슐화 패킷	65	×	
링크 섷다운 패킷	69	×	
인터페이스 타입 핸드오프 요청 패킷	75	×	
인터페이스 타입 응답 패킷	76		×
구현 타입 핸드오프 패킷	77	×	
순방향 오디오 채널 인에이블 패킷	78	×	
왕복 지연 측정 패킷	82	×	
순방향 스큐 계산 패킷	83	×	

표 VI-2: 기본 미디어 스트림 패킷

패킷 이름	패킷 타입	순방향에서 유효성	역방향에서 유효성
비디오 스트림 패킷	16	×	×
오디오 스트림 패킷	2	×	×
반전된 스트림 패킷	1-15, 18-31, 33-55	×	×
사용자 한정 스트림 패킷	6-63	×	×
제어 맵 패킷	4	×	×
반전된 오디오 샘플 레이트 패킷	9	×	
투명 컬러 인에이블 패킷	1	×	

표 VI-3: 클라이언트 상태 및 제어 패킷들

패킷 이름	패킷 타입	순방향에서 유효성	역방향에서 유효성
클라이언트 기능 패킷	66		×
키보드 데이터 패킷	67	×	×
포인팅 장치 데이터 패킷	68	×	×
클라이언트 요청 및 상태 패킷	70		×
디지털 콘텐츠 보호 오버헤드 패킷	80	×	×
요청 VCP 특성 패킷	128	×	
VCP 특성 응답 패킷	129		×
세트 VCP 특성 패킷	130	×	
요청 유효 파라미터 패킷	131	×	
유효 파라미터 응답 패킷	132		×
요청 특성 상태 패킷	138	×	
유효 상태 응답 리스크 패킷	139		×
패킷 프로세싱 지연 파라미터 패킷	140		×
개인 디스플레이 기능 패킷	141		×
클라이언트 에러 보고 패킷	142		×
스케일링된 비디오 스트림 기능 패킷	143		×
클라이언트 식별 패킷	144		×
택일적 디스플레이 기능 패킷	145		×
레지스터 액세스 패킷	146	×	×

표 VI-4: 진보한 그래픽 및 디스플레이 패킷

패킷 이름	패킷 타입	순방향에서 유효성	역방향에서 유효성
비트 블록 전송 패킷	71	×	
비트맵 영역 총족 패킷	72	×	
비트맵 패턴 총족 패킷	73	×	
판독 프레임 버퍼 패킷	74	×	
알파 커서 이미지 기능 패킷	133		×
알파 커서 투명 맵 패킷	134	×	
알파 커서 이미지 오프셋 패킷	135	×	
알파 커서 비디오 스트림 패킷	17	×	
스케일링된 비디오 스트림 기능 패킷	143		×
스케일링된 비디오 스트림 셋업 패킷	136	×	

스케일링된 비디오 스트림 응답 패킷	137		x
스케일링된 비디오 스트림 패킷	18	x	

본 명세서 내의 다른 논의와 명확하게 구별되는 것은 반전 캡슐화 패킷, 클라이언트 기능 패킷, 및 클라이언트 요청 및 상태 패킷은 각각 외부 모드 연산의 경우 통신 인터페이스의 많은 실시예에 중요하거나 심지어 요구되는 반면, 내부 모드 연산의 경우 이들은 선택적인 것이거나 거의 그와 유사하다. 이는 감소된 세트의 통신 패킷, 및 대응하는 제어 및 타이밍의 간략화와 함께 매우 높은 속도로 데이터의 통신을 가능하게 하는 MDD 인터페이스 프로토콜의 또다른 타입을 생성한다.

패킷은 패킷 길이 필드, 패킷 타입 필드, 데이터 바이트 필드, 및 도7에 도시된 CRC를 포함하는 공통의 기본 구조 또는 최소 필드의 전체 세트를 갖는다. 도7에 도시된 바와 같이, 패킷 길이 필드는 패킷에서 전체 비트의 수 또는 패킷 길이 필드와 CRC 필드 사이의 길이를 특징하는 다중 비트 또는 바이트 값의 형태로 정보를 포함한다. 일 실시예에서, 패킷 길이 필드는 패킷 길이를 한정하는 16 비트 또는 2 바이트 폭의 부호 없는 정수를 포함한다. 패킷 타입 필드는 패킷 내에 포함된 정보의 타입을 나타내는 다른 멀티 비트 필드이다. 실시예에서, 이는 16비트 부호 없는 정수의 형태로 16비트 또는 2 바이트 폭이며, 이러한 데이터 타입을 디스플레이 기능, 핸드오프, 비디오 또는 오디오 스트림, 상태 등으로서 특징한다.

제3 필드는 데이터 바이트 필드로서, 패킷의 일부로서 호스트와 클라이언트 장치 사이에서 송신 또는 전송되는 비트 또는 데이터를 포함한다. 데이터의 포맷은 송신될 데이터의 특정 타입에 따른 각각의 패킷 타입에 대해 구체적으로 한정되며 부가의 필드의 시리즈로 분리될 수도 있는데, 각각은 자신 고유의 요구 조건을 갖는다. 즉, 각각의 패킷 타입은 이러한 부분 또는 필드에 대해 한정된 형태를 가질 것이다. 최종 필드는 데이터 바이트, 패킷 타입, 및 패킷 길이 필드에 대해 계산되는 16비트 주기 리턴던시 체크의 결과를 포함하는 CRC 필드인데, 패킷에서 정보의 무결성을 형성하는데 사용된다. 다시 말해, CRC 필드 자체를 제외하고 전체 패킷에 대해 계산된다. 클라이언트는 일반적으로 검출된 CRC 에러의 전체 양을 유지하고 이를 클라이언트 요청 및 상태 패킷에서 호스트로 되돌려준다(이하 참조).

일반적으로, 이러한 필드 폭 및 구성은 짝수 바이트 경계에 정렬된 2바이트 필드, 및 4바이트 경계에 정렬된 4바이트 필드를 유지하도록 설계된다. 이는 패킷 구조가 대부분 또는 통상 사용되는 프로세서 또는 제어 회로에 대해 만나는 데이터 타입 정렬 규칙을 깨트리지 않고 호스트와 클라이언트의 또는 이와 관련한 주메모리 공간에 용이하게 구축되게 한다.

패킷의 전송 동안, 필드는 우선 최하위 비트(LSB)에서 시작하여 최종으로 전송된 최상위 비트(MSB)로 종료하게 송신된다. 길이가 1바이트 이상인 파라미터는 우선 최하위 비트를 이용하여 송신되는데, 이는 LSB가 우선 송신되는 더 짧은 파라미터에 사용됨에 따라 8비트 길이보다 큰 파라미터에 대해 사용되는 동일한 비트 송신 패턴을 초래한다. 각각의 패킷의 데이터 필드는 일반적으로 이하의 연속한 섹션에서 한정된 순서로 송신되는데, 리스트된 제1 필드가 우선 송신되고, 설명된 최종 필드가 마지막으로 송신된다. MDDI_Data0 신호 경로에 대한 데이터는 타입1, 타입2, 타입3, 또는 타입4와 같은 소정의 모드에서 인터페이스에 대해 송신된 바이트의 '0'비트와 정렬된다.

디스플레이를 위해 데이터를 조작할 때, 픽셀의 어레이를 위한 데이터는 전자 분야에서 통상적으로 행해지는 바와 같이, 우선 행으로 송신되고, 이어 열로 송신된다. 다시 말해, 비트 맵의 동일한 행에 나타나는 모든 픽셀은 가장 좌측 픽셀이 우선 송신되고 가장 우측 픽셀이 최종으로 송신되는 순서로 송신된다. 행의 최우측 픽셀이 송신된 후, 시퀀스의 다음 픽셀은 다음 행의 가장 좌측 픽셀이다. 다른 구성이 필요에 따라 수용될 수 있지만, 픽셀의 행들이 일반적으로 디스플레이의 최상부로부터 하부의 순으로 송신된다. 더욱이, 비트맵의 조절시, 본 명세서에 따르는 통상의 접근은 위치 또는 오프셋 "0,0"으로서 비트맵의 좌상부 코너를 라벨링함으로써 기준 포인트를 한정하는 것이다. 비트맵에서 위치를 한정 또는 결정하기 위해 사용되는 X 및 Y 좌표는 각각 비트맵의 우측 및 하부에 도달함에 따라 증가한다. 제1 행 및 제1 열(이미지의 좌상부 코너)은 0의 지수 값으로 시작한다. 디스플레이의 사용자에게 의해 보여지는 바와 같이, X 좌표의 크기는 이미지의 우측을 향해 증가하며, Y 좌표의 크기는 이미지의 하부를 향해 증가한다.

디스플레이 윈도우는 비트맵의 가시 부분이며, 비트맵의 픽셀 부분은 물리적인 디스플레이 매체 상에서 사용자에게 의해 보여질 수 있다. 디스플레이 유닛 및 비트맵이 동일한 크기인 경우가 통상적이다. 디스플레이 윈도우의 좌측 상부 코너는 언제나 비트맵 픽셀 위치 '0,0'를 디스플레이한다. 디스플레이 윈도우의 폭은 비트맵의 X축에 대응하며, 이러한 실시예의 디스플레이 윈도우 폭은 대응하는 비트맵의 폭과 동일하거나 작다. 윈도우의 높이는 비트맵의 Y축에 대응하며, 본 실시예에 대한 디스플레이 윈도우 높이는 대응하는 비트맵의 높이와 동일하거나 작다. 디스플레이 윈도우 그 자체는 비트맵의 가시적 부분으로만 한정되기 때문에 프로토콜에 어드레싱 가능하지 않다.

비트맵과 디스플레이 윈도우 사이의 관계는 컴퓨터, 전자 분자, 인터넷 통신, 및 다른 전자 관련 기술분야에 공지되어 있다. 따라서, 이러한 원리의 추가 논의 또는 설명은 제공하지 않는다.

C. 패킷 정의

1. 서브 프레임 헤더 패킷

서브 프레임 헤더 패킷은 모든 서브 프레임의 제1 패킷이며, 도8에 도시된 기본 구조를 갖는다. 서브 프레임 헤더 패킷은 호스트 클라이언트 동기화를 위해 사용되며, 모든 호스트는 이러한 패킷을 생성할 수 있는 반면, 모든 클라이언트는 이러한 패킷을 수신 및 번역할 수 있어야 한다. 도8의 일 실시예에 도시될 수 있는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, 고유 워드, 예비된 1, 서브 프레임 길이, 프로토콜 버전, 서브 프레임 카운트 및 미디어 프레임 카운트 필드를 일반적으로 이 순서로 갖도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 타입 15359(0x3bff 16진수)로서 일반적으로 한정되며 20 바이트의 예정된 고정 길이를 이용하지만, 패킷 길이 필드는 포함하지 않는다.

패킷 타입 필드 및 고유 워드 필드는 각각 2바이트 값(16비트 부호 없는 정수)을 사용한다. 이러한 두 필드의 4바이트 조합은 32비트 고유 워드를 양호한 자동 상관과 함께 형성한다. 일 실시예에서, 실제 고유 워드는 0x005a3bff이며, 하위 16비트는 패킷 타입으로 우선 송신되고, 최상위 16비트는 그 후에 송신된다.

예비된 1 필드는 미래의 사용을 위해 예비된 공간인 2바이트를 포함하며, 일반적으로 이 시점에서 모든 비트가 0으로 구성된다. 이러한 필드의 일 목적은 연속한 2바이트 필드가 16비트 워드 어드레스로 정렬되게 하고 4바이트 필드가 32비트 워드 어드레스로 정렬되게 한다. 최하위 바이트는 호스트가 다중 클라이언트 장치를 어드레싱할 수 있는 지를 나타내기 위해 예비된다. 이러한 바이트를 위한 0의 값은 호스트가 단일 클라이언트 장치와 연산가능하다는 것을 나타내기 위해 예비된다.

서브 프레임 길이 필드는 서브 프레임당 바이트의 수를 한정하는 4바이트의 정보, 또는 값을 포함한다. 일 실시예에서, 필드의 길이는 단지 하나의 서브 프레임이 링크가 휴지 상태로 닫히기 전에 호스트에 의해 송신될 것을 나타내기 위해 0으로 설정될 수도 있다. 이러한 필드에서의 값은 하나의 서브 프레임으로부터 다음 서브 프레임으로 송신시 동적으로 변화하는 "온-더-플라이"일 수 있다. 이러한 기능은 등시성 데이터 스트림을 수용하기 위해 동기 펄스에서 다소의 시간 조절을 하기 위해 유용하다. 서브 프레임 헤더 패킷의 CRC가 유효하지 않으면, 링크 제어기는 현재 서브 프레임의 길이를 추정하기 위해 공지된 양호한 서브 프레임 헤더 패킷의 서브 프레임 길이를 이용해야 한다.

프로토콜 버전 필드는 호스트에 의해 사용된 프로토콜 버전을 설명하는 2바이트를 포함한다. 프로토콜 버전 필드는 사용될 프로토콜의 제1 또는 현재 버전을 설명하기 위해 '0'으로 설정될 수도 있다. 이러한 값은 새로운 버전이 생성됨에 따라 시간에 대해 변화할 것이며, 일부 버전 필드에 대해 '1'의 값으로 이미 업그레이드된다. 버전 값은 알려진 바와 같이 MDDI와 같은 인터페이스를 커버하는 승인된 표준 문서에 대한 현재 버전 수를 따른다.

서브 프레임 카운트 필드는 미디어 프레임의 개시 이후, 송신된 서브 프레임의 수를 나타내는 시퀀스 수를 설명하는 2바이트를 포함한다. 미디어 프레임의 제1 서브 프레임은 0의 서브 프레임 카운트를 갖는다. 미디어 프레임의 최종 서브 프레임은 $n-1$ 의 값을 갖는데, 여기서, n 은 미디어 프레임당 서브 프레임의 수이다. 서브 프레임 카운트 필드의 값은 0의 카운트를 가질 미디어 프레임의 서브 프레임을 제외하고, 앞선 서브 프레임 패킷의 서브 프레임 카운트에 1을 더한 것과 같다. 만일 서브 프레임 길이가 (비주기적 서브 프레임을 나타내는) 0과 같으면, 서브 프레임 카운트는 또한 0과 같게 세팅된다.

미디어 프레임 카운트 필드는 전송되는 현재 미디어 아이템 또는 데이터의 개시 이후 송신된 미디어 프레임의 수를 나타내는 시퀀스 수를 설명하는 4 바이트(32비트 부호 없는 정수)를 포함한다. 미디어 아이템의 제1 미디어 프레임은 0의 미디어 프레임 카운트를 갖는다. 미디어 프레임 카운트는 각각의 미디어 프레임의 서브 프레임에 앞서 증가하며, 최대 미디어 프레임 카운트(예를 들어, 미디어 프레임의 수 $2^{32}-1=4,294,967,295$)가 사용된 후, 0으로 돌아온다. 미디어 프레임 카운트 값은 엔드 애플리케이션의 요구에 적합하게 하기 위해 호스트에 의해 소정의 시간에 일반적으로 리셋될 수도 있다.

2. 필터 패킷

필터 패킷은 어떠한 다른 정보도 순방향 또는 역방향 링크상으로 전송되는 것이 유용하지 않을 경우 클라이언트 장치와 송수신하는 패킷이다. 필터 패킷이 필요한 경우 필터 패킷이 다른 패킷을 송신하는 최대 유연성을 가능하게 하기 위해 최소 길이를 갖는 것이 권장된다. 서브 프레임 또는 역방향 링크 캡슐화 패킷(이하 참조)의 바로 단부에서, 링크 제어기는 패킷

밀도를 유지하기 위해 잔여 공간을 필딩하기 위해 필러 패킷의 크기를 설정한다. 필러 패킷은 호스트 또는 클라이언트가 송신 또는 교환할 어떠한 정보도 없는 경우, 링크 상의 시간을 유지하는데 유용하다. 모든 호스트 및 클라이언트는 인터페이스의 유효한 사용을 위해 이러한 패킷을 송신 및 수신하는 것을 필요로 한다.

필러 패킷의 포맷 및 콘텐츠의 실시예는 도9에 도시된다. 도9에 도시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, 필러 바이트, 및 CRC 필드를 갖도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 통상적으로 타입 0으로 한정되는데, 이는 2바이트 타입 필드로 표시된다. 필러 바이트 필드의 비트 또는 바이트는 필러 패킷이 원하는 길이로 되도록 모든 0비트 값의 가변 수를 포함한다. 더 작은 필러 패킷은 이러한 필드에서 어떠한 바이트도 포함하지 않는다. 즉, 패킷은 단지 패킷 길이, 패킷 타입, 및 CRC로 구성되며, 일 실시예에서 6바이트의 예정된 고정 길이 또는 4의 패킷 길이 값을 이용한다. CRC 값은 패킷 길이를 포함하는 패킷에서 모든 바이트에 대해 결정되는데, 이는 다른 패킷 타입에서 제외될 수도 있다.

3. 비디오 스트림 패킷

비디오 스트림 패킷은 디스플레이 장치의 통상적인 직사각형 영역을 업데이트 하기 위해 비디오 데이터를 운반한다. 이러한 영역의 크기는 단일 픽셀만큼 작거나, 전체 디스플레이만큼 클 수도 있다. 스트림을 디스플레이하기 위해 필요한 모든 콘텐츠가 비디오 스트림 패킷 내에 포함되므로, 스트림 리소스에 의해 한정된, 동시에 디스플레이되는 거의 제한없는 수의 스트림이 존재한다. 비디오 스트림 패킷(비디오 데이터 포맷 기술자)의 일 실시예의 포맷은 도10에 도시된다. 도10에 도시된 바와 같이, 일 실시예에서, 이러한 타입의 패킷은 패킷 길이(2바이트), 패킷 타입, b클라이언트 ID, 비디오 데이터 기술자, 픽셀 디스플레이 속성, X 좌측 가장자리, Y 상부 가장자리, X 우측 가장자리, Y 하부 가장자리, X 및 Y 시작, 픽셀 카운트, 파라미터 CRC, 픽셀 데이터, 및 픽셀 데이터 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 16으로 식별되는데, 이는 2바이트 타입 필드에서 표시된다. 일 실시예에서, 클라이언트는 RGB를 사용하는 비디오 스트림 패킷, 모노크롬, 및 클라이언트 기능 패킷의 Y Cr Cb 기능 필드를 수신하는 성능을 나타낸다.

일 실시예에서, b클라이언트 ID 필드는 클라이언트 ID에 대해 유지된 2바이트의 정보를 포함한다. 이는 새롭게 개발된 통신 프로토콜이므로, 실제 클라이언트 ID는 아직 알려지지 않거나 충분히 상업적이지 않다. 따라서, 이러한 필드의 비트는 통상적으로 상기 ID 값이 알려지기 전까지 0으로 설정되는데, 이때 ID 값은 기술 분야의 당업자에게 명확하듯이, 삽입되거나 사용될 수 있다. 동일한 프로세스가 통상적으로 후술하는 클라이언트 ID 필드에 대해 적절할 것이다.

전술한 공통의 프레임 개념은 오디오 버퍼 크기를 최소화하고 레이턴스를 감소시키는 효율적인 방식이다. 그러나 비디오 데이터의 경우, 미디어 프레임 내의 다수의 비디오 스트림 패킷에 걸쳐 하나의 비디오 프레임의 픽셀을 확산시킬 필요가 있을 수도 있다. 단일 비디오 스트림 패킷에서의 픽셀이 디스플레이에 대해 완벽한 직사각형 윈도우에 대응하지 않을 가능성이 매우 높다. 초당 30 프레임의 비디오 프레임 레이트의 예일 경우, 초당 300 서브 프레임이 존재하며, 이는 미디어 프레임당 10 서브 프레임을 초래한다. 만일 각각의 프레임에서 480 행의 픽셀이 존재하면, 각각의 서브 프레임에서 각각의 비디오 스트림 패킷은 48 행의 픽셀을 포함할 것이다. 다른 상황에서, 비디오 스트림 패킷은 정수의 픽셀 행을 포함할 것이다. 다른 이는 미디어 프레임당 서브 프레임의 수가 비디오 프레임당행의 수(비디오 라인으로도 알려짐)로 균등하게 분할되지 않는 다른 비디오 프레임 크기에 대해 적절하다. 효율적인 동작을 위해, 각각의 비디오 스트림 패킷은 심지어 정수 행의 픽셀을 포함하지 않는다고 해도, 통상적으로 정수의 픽셀을 포함해야 한다. 이는 픽셀이 각각 1바이트 이상이거나, 이들이 도12에 도시된 패킷 형태에 있는 경우 중요하다.

전술한 바와 같이, 예로든 비디오 데이터 기술자 필드의 연산을 실현하기 위해 사용된 포맷 및 콘텐츠는 도11A-11E에 도시된다. 도11A-11E에서, 비디오 데이터 포맷 기술자 필드는 현재 패킷에서 현재의 스트림의 픽셀 데이터에서 각각의 픽셀의 포맷을 설명하는 16비트 부호 없는 정수의 형태로 2바이트를 포함한다. 상이한 비디오 스트림 패킷이 상이한 픽셀 데이터 포맷을 사용, 즉 비디오 데이터 포맷 기술자에서 상이한 값을 사용할 수도 있는 것이 가능하며, 유사하게, 스트림(디스플레이의 영역)은 자신의 데이터 포맷 온-더-플라이를 변경할 수도 있다. 픽셀 데이터 포맷은 클라이언트 기능 패킷에서 한정된 바와 같이 클라이언트에 대한 유효한 포맷 중 적어도 하나에 따를 수 있다. 비디오 데이터 포맷 기술자는 일정한 포맷이 특정 비디오 스트림의 수명에 대해 계속하여 사용될 것을 의미하지 않는 단지 현재 패킷에 대한 픽셀 포맷을 한정한다.

도11A 내지 11D는 비디오 데이터 포맷 기술자가 코딩되는 방법을 설명한다. 이러한 도면에서 사용된 바와 같이, 이러한 실시예에서, 비트[15:13]는 도11A에 도시된 바와 같이 '000'과 동일하며, 비디오 데이터는 모노크롬 픽셀의 어레이로 구성되는데, 픽셀당 비트의 수는 비디오 데이터 포맷 기술자 워드의 비트 3 내지 0으로 한정된다. 비트 11 내지 4는 통상적으로 미래의 사용 또는 애플리케이션을 위해 예비되며 이러한 상황에서는 0으로 설정된다. 비트[15:13]이 값'001'로 인서트되면, 도 11B에 도시된 바와 같이, 비디오 데이터는 컬러 맵(팔레트) 전체의 컬러를 각각 설명하는 컬러 픽셀의 어레이

로 구성된다. 이러한 상황에서, 비디오 데이터 포맷 기술자 워드의 비트 5 내지 0은 픽셀당 비트의 수를 한정하며, 비트 11 내지 6은 통상적으로 미래의 사용 또는 애플리케이션을 위해 예비되며 0으로 설정된다. 비트[15:13]이 '010'으로 인서팅 되면, 도11C에 도시된 바와 같이, 비디오 데이터는 컬러 픽셀의 어레이로 구성되는데, 빨강의 픽셀당 비트의 수는 비트 11 내지 8로 한정되며, 녹색의 픽셀당 비트 수는 비트 7 내지 4로 한정되며, 파랑의 픽셀당 비트의 수는 비트 3 내지 0으로 한정된다. 이러한 상황에서, 각각의 픽셀에서 전체 비트의 수는 빨강, 녹색, 및 파랑에 사용된 비트의 수의 합이다.

그러나 비트[15:13]이 도11D에 도시된 바와 같이, 대신에 값 또는 스트림 '011'과 동일할 경우, 비디오 데이터는 휘도 및 색상 정보를 갖는 4:2:2 YCbCr 포맷으로 비디오 데이터의 어레이로 구성되는데, 휘도(Y)의 픽셀당 비트의 수는 비트 11 내지 8로 한정되며, Cb 성분의 비트의 수는 비트 7 내지 4로 한정되며, Cr 성분의 비트의 수는 비트 3 내지 0으로 한정된다. 각각의 픽셀에서 비트의 전체 수는 빨강, 녹색 및 청색에 사용된 비트의 수의 합이다. Cb 및 Cr 성분은 Y의 절반 레이트로 전송된다. 게다가, 이러한 패킷의 픽셀 데이터 부분의 비디오 샘플은 다음과 같이 구성된다: Cbn, Yn, Crn, Yn+1, Cbn+2, Yn+2, Yn+3,...이며, 여기서 Cbn 및 Crn은 Yn 및 Yn+1과 관련되며, Cbn+2 및 Crn+2는 Yn+2 및 Yn+3 등과 관련된다.

Yn, Yn+1, Yn+2 및 Yn+3은 좌측에서 우측으로 단일 행의 4개의 연속한 픽셀의 휘도 값이다. 만일 비디오 스트림 패킷에 의해 참조된 윈도우에서 행(X 우측 가장자리-X 좌측 가장자리+1)의 픽셀의 홀수가 존재하면, 각각의 행의 최종 픽셀에 대응하는 Y 값은 다음 행의 첫 번째 픽셀의 Cb 값에 앞서며, Cr 값은 행의 최종 픽셀을 위해 전송되지 않는다. Y Cb Cr 포맷을 사용하는 윈도우는 픽셀의 짝수인 폭을 갖는 것이 권장된다. 패킷에서의 픽셀 데이터는 짝수의 픽셀을 포함해야 한다. 이는 픽셀 데이터의 최종 픽셀이 비디오 스트림 패킷 헤더에 설명된 윈도우에서 행의 최종 픽셀에 대응하는 경우, 즉 픽셀 데이터에서 최종 픽셀의 X 위치는 X 우측 가장자리와 동일한 때, 픽셀의 홀수 및 짝수를 포함할 수도 있다.

대신에 비트[15:13]이 값'100'과 동일한 경우, 비디오 데이터는 Bayer 픽셀의 어레이로 구성되는데, 픽셀당 비트의 수는 비디오 데이터 포맷 기술자 워드의 비트 3 내지 0으로 한정된다. 픽셀 그룹 패턴은 도11E에 도시된 바와 같이 비트 5 및 4로 한정된다. 픽셀 데이터의 순서는 수평 또는 수직일 수 있으며, 행 또는 열에서의 픽셀은 순방향 또는 역방향 순서로 전송될 수도 있으며, 비트 8 내지 6으로 한정된다. 비트 11 내지 9는 0으로 설정된다. Bayer 포맷에서 픽셀 그룹의 4개의 픽셀 그룹은 소정의 디스플레이 기술에서 단일 픽셀로 종종 언급되는 것과 유사하다. 그러나 Bayer 포맷에서 하나의 픽셀은 픽셀 그룹 모자이크 패턴의 4개의 컬러링 된 픽셀 중 하나이다.

도면에 도시된 모든 5개 포맷의 경우, "P"로 표시된 비트 12는 픽셀 데이터 샘플이 패킹되거나, 바이트 정렬된 픽셀 데이터인지를 설명한다. 이러한 필드에서의 '0'의 값은 픽셀 데이터 필드에서 각각의 픽셀이 MDD 인터페이스 바이트 경계와 바이트 정렬되는 것을 나타낸다. '1'의 값은 각각의 픽셀 및 각각의 픽셀 내의 각각의 컬러가 앞선 픽셀 또는 사용되지 않은 비트를 남기지 않은 픽셀 내의 컬러에 대해 패킹-업됨을 의미한다. 바이트 정렬과 패킹된 픽셀 데이터 포맷 사이의 차이는 도12에 더욱 상세하게 도시되며, 바이트 정렬된 데이터가 사용된 패킹된 픽셀 포맷과 대조적으로 데이터 서브 프레임의 사용되지 않은 부분으로 남길 수도 있음을 보여준다.

특정 디스플레이 윈도우에 대한 미디어 프레임의 제1 비디오 스트림 패킷에서 제1 픽셀은 X 좌측 가장자리 및 Y 상부 가장자리에 의해 한정된 스트림 윈도우의 상부 좌측 코너로 진행할 것이며, 수신된 다음의 픽셀은 동일한 행 등의 다음 픽셀 위치에 배치된다. 미디어 프레임의 이러한 제1 패킷에서, X 시작 값은 X 좌측 가장 자리와 통상적으로 동일하며, Y 시작 값은 통상적으로 Y 상부 가장자리와 동일할 것이다. 동일한 스크린 윈도우에 대응하는 연이은 패킷에서, X 및 Y 시작 값은 통상적으로 앞선 서브 프레임에서 송신된 비디오 스트림 패킷에서 최종 픽셀이 송신된 후 따르는 스크린 윈도우에서의 픽셀 위치로 설정될 것이다.

4. 오디오 스트림 패킷

오디오 스트림 패킷들은 클라이언트의 오디오 시스템을 통해, 또는 독립 오디오 프리젠테이션 장치를 통해 플레이되는 오디오 데이터를 전달한다. 상이한 오디오 데이터 스트림들이 사운드 시스템의 개별 오디오 채널들에 할당될 수 있다: 예를 들면, 사용되는 오디오 시스템 타입에 따라 좌측-전방, 우측-전방, 중앙, 좌측-후방, 및 우측-후방. 오디오 채널들의 완전한 보완은 강화된 공간-음향 신호 처리를 포함하는 헤드셋을 통해 제공된다. 클라이언트는 클라이언트 성능 패킷의 오디오 채널 성능 및 오디오 샘플 레이트 필드들을 사용하여 오디오 스트림 패킷을 수신한다. 오디오 스트림 패킷의 포맷은 도 13에 제시된다.

도 13에 제시된 바와 같이, 이러한 타입의 패킷은 일 실시예에서 패킷 길이, 패킷 타입, bClient ID, 오디오 채널 ID, 예약 1, 오디오 샘플 카운트, 샘플 및 패킹당 비트, 오디오 샘플 레이트, 파라미터 CRC, 디지털 오디오 데이터, 및 오디오 데이터 CRC 필드들을 가지도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 일반적으로 타입 32 패킷으로 식별된다.

bClient ID 필드는 이전에 사용된 것과 같이 클라이언트 ID를 위해 예약되는 2 바이트 정보를 포함한다. 예약 1 필드는 차후 사용을 위해 예약되는 2 바이트를 포함하고, 일반적으로 이 지점에서 모든 비트들은 0으로 설정되도록 구성된다.

샘플 및 패킹당 비트들 필드는 오디오 데이터의 패킹 포맷을 규정하는 부호를 갖지 않는 8비트의 정수 형태의 1 바이트를 포함한다. 이러한 포맷은 일반적으로 PCM 오디오 샘플당 비트들의 수를 정의하기 위한 비트 4 내지 0을 위해 사용된다. Y 비트 5는 디지털 오디오 데이터 샘플들이 패킹되는지 여부를 규정한다. 패킹 및 바이트-정렬 오디오 샘플들 사이의 차이는(여기서, 10비트 샘플들을 사용함) 도 14에 제시된다. 값 "0"은 디지털 오디오 데이터 필드 내의 각각의 PCM 오디오 샘플이 MDDI 인터페이스 바이트 경계와 바이트-정렬된다는 것을 표시하고, 값 "1"은 각각의 연속적인 PCM 오디오 샘플이 이전 오디오 샘플과 패킹된다는 것을 표시한다. 이러한 비트는 일반적으로 비트 4-0에서 정의되는 값(PCM 오디오 샘플당 비트들의 수)이 8의 배수가 아닌 경우에만 효과적이다. 비트 7-6은 차후 사용을 위해 예약되고, 일반적으로 0 값으로 설정된다.

5. 반전된 스트림 패킷

일 실시예에서, 패킷 타입 1 내지 15, 18 내지 31, 및 33 내지 55는 다양한 애플리케이션에 요구되는 바와 같이, 미래의 버전 또는 패킷 프로토콜의 변화에 사용하기 위해 한정된 스트림 패킷에 대해 반전된다. 또한, 이는 다른 기술과 비교하여 기술 및 시스템의 변화에 직면하여 MDD 인터페이스가 더욱 플렉서블하고 유용하게 하는 부분이다.

6. 사용자 한정된 스트림 패킷

타입 56 내지 63으로 알려진 8개의 데이터 스트림 타입은 MDDI 링크와 사용하기 위해 설비 제조자에 의해 한정될 수도 있는 적절한 애플리케이션에 사용하기 위해 예비된다. 이들은 사용자 한정 스트림 패킷으로 알려져 있다. 이러한 패킷은 소정의 목적을 위해 사용될 수도 있지만, 호스트 및 클라이언트는 이러한 사용의 결과가 잘 이해되고 알려지는 상황에서만 상기 패킷을 사용한다. 이러한 패킷 타입에 대한 스트림 파라미터 및 데이터의 특정한 한정은 이러한 패킷 타입을 구현하고 이들의 사용을 탐색하는 특정한 설비 제조자 또는 인터페이스 설계자에 남겨진다. 사용자 한정 스트림 패킷의 몇몇 사용에는 테스트 파라미터 및 테스트 결과, 공장 공차 데이터, 및 주변 특정 사용 데이터를 운반하는 것이다. 일 실시예에서 사용된 바와 같은 사용자 한정 스트림 패킷의 포맷은 도 15에 도시된다. 도 15에 도시된 바와 같이, 이러한 타입의 패킷은 패킷 길이(2바이트), 패킷 타입, b클라이언트 ID 번호, 스트림 파라미터, 파라미터 CRC, 스트림 데이터, 및 스트림 데이터 CRC 필드를 갖도록 구성된다.

7. 컬러 맵 패킷

컬러 맵 패킷은 클라이언트에 대해 컬러를 제공하기 위해 사용된 컬러 맵 록업 테이블의 콘텐츠를 설명한다. 일부 애플리케이션은 단일 패킷에서 송신가능한 데이터의 양보다 더 큰 컬러 맵을 필요로 할 수도 있다. 이러한 경우, 후술되는 각각의 오프셋 및 길이 필드를 사용함으로써 컬러 맵의 상이한 서브셋을 갖는 다중 컬러 맵 패킷은 전송될 수도 있다. 일 실시예에서 컬러 맵 패킷의 포맷은 도 16에 도시된다. 도 16에 도시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, h클라이언트 ID, 컬러 맵 아이템 카운트, 컬러 맵 오프셋, 파라미터 CRC, 컬러 맵 데이터, 및 데이터 CRC 필드를 갖도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 일반적으로 패킷 타입 필드(2바이트)에 설명된 바와 같이 타입 64 패킷(비디오 데이터 포맷 및 컬러 맵 패킷)으로서 식별된다. 클라이언트는 클라이언트 기능 패킷의 컬러 맵 폭 필드와 컬러 맵 크기를 사용하여 컬러 맵 패킷을 수신하는 성능을 나타낸다.

8. 역방향 링크 캡슐화 패킷

일 실시예에서, 데이터는 역방향 링크 캡슐화 패킷을 이용하여 역방향으로 전송된다. 순방향 패킷이 전송되며 MDDI 링크 연산(전송 방향)은 패킷이 역방향으로 전송될 수 있도록 이러한 패킷의 중앙 부근에서 변화 또는 전환된다. 일 실시예에서 역방향 링크 캡슐화 패킷의 포맷은 도 17에 도시된다. 도 17에 도시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, h클라이언트 ID, 역방향 링크 플래그, 역방향 레이트 디바이저, 턴 어라운드 1, 역방향 데이터 패킷, 턴 어라운드 2, 및 모든 제로 2 필드를 갖도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 일반적으로 타입 65 패킷으로 식별된다. 외부 모드에 대해, 모든 호스트는 이러한 패킷을 생성하고 데이터를 수신해야 하며, 모든 클라이언트는 요구된 프로토콜 및 결과 속도의 사용을 효율적으로 이용하기 위해 호스트로 데이터를 전송할 수 있어야 한다. 이러한 패킷의 구현은 인터넷 모드에 대해 선택적이지만, 역방향 링크 캡슐화 패킷은 클라이언트로부터 데이터를 수신하기 위해 호스트에 대해 사용된다.

MDDI 링크 제어기는 역방향 링크 캡슐화 패킷을 전송하면서 특정한 방식으로 동작한다. MDD 인터페이스는 링크의 제어기로서 호스트에 의해 일반적으로 항상 구동된다. 호스트는 역방향 링크 캡슐화 패킷의 턴 어라운드 및 역방향 데이터 패킷 부분의 각각의 비트에 대해 0을 송신하는 것처럼 동작한다. 호스트는 두 개의 턴 어라운드 시간 동안 그리고 역방향 데이터 패킷에 대해 한정된 시간 동안 각각의 비트 경계에서 MDDI_스트로브 신호를 토글한다. (이는 모든 제로 데이터를 송신하는 것과 동일한 특성이다.)

호스트는 턴 어라운드1에 의해 설명된 시간 주기 동안 MDDI 데이터 신호 라인 드라이버를 디스에이블하고, 클라이언트는 턴 어라운드 2 필드에 의해 설명된 시간 주기를 따르는 드라이버 리인에이블 필드 동안 자신의 라인 드라이버를 리인에이블한다. 클라이언트는 턴 어라운드 길이 파라미터를 판독하고 턴 어라운드 1 필드에서 마지막 비트 후에 즉시 호스트를 향해 데이터 신호를 구동시킨다. 즉, 클라이언트는 이하의 패킷 콘텐츠에서 설명되고 곳곳에서 설명된 바와 같이, 새로운 데이터를 MDDI 스트로브의 소정의 상승 에지에 대한 링크로 클록킹한다. 클라이언트는 패킷을 호스트로 전송할 수 있는 시간의 길이를 알기 위해 패킷 길이 및 턴 어라운드 길이 파라미터를 이용한다. 클라이언트는 호스트로 전송한 데이터가 없는 경우 필러 패킷을 전송하거나 데이터 라인을 0으로 구동시킬 수도 있다. 만일 데이터 라인이 0으로 구동되면, 호스트는 이를 0의 길이(유효한 길이가 아님)를 가진 패킷으로 해석하고, 호스트는 현재의 역방향 링크 캡슐화 패킷의 지속 기간 동안 클라이언트로부터 소정 이상의 패킷을 수신하지 않는다.

호스트는 MDDI_데이터 신호를 모든 제로 1 필드 동안 로직 제로 레벨로 구동시키고, 클라이언트는 턴 어라운드 2 필드의 시작 전에, 즉 모든 제로 2 필드 기간 동안 적어도 하나의 역방향 링크 클록 기간 동안 MDDI 데이터 라인을 로직 제로 레벨로 구동시킨다. 이는 데이터 라인을 턴 어라운드 1 및 턴 어라운드 2 필드 시간 기간 동안 결정적인 상태로 유지시킨다. 만일 클라이언트가 전송할 더 이상의 패킷이 없는 경우, 동면 바이어스 레지스터(다른 곳에서 설명함)가 역방향 데이터 패킷 필드의 나머지에 대해 또는 약 16개의 순방향 링크 또는 그 이상의 기간에 대해 논리 제로 레벨로 데이터 라인을 유지하기 때문에, 클라이언트는 패킷들을 구동시킨 후 데이터 라인을 로직 제로 레벨로 디스에이블할 수도 있다.

일 실시예에서, 클라이언트 요청 및 상태 패킷의 역방향 링크 요청 필드는 클라이언트가 데이터를 호스트로 역전송하기 위해 역방향 링크 캡슐화 패킷에서 필요로 하는 바이트의 수를 호스트에게 통보하는데 사용될 수도 있다. 호스트는 역방향 링크 캡슐화 패킷에서 바이트의 최소한의 수를 할당함으로써 요청을 승인하도록 시도한다. 호스트는 서브 프레임에서 하나 이상의 역방향 링크 캡슐화 패킷을 전송할 수도 있다. 클라이언트는 클라이언트 요청 및 상태 패킷을 거의 임의의 시간에 전송할 수도 있으며, 호스트는 하나의 서브 프레임에서 요청된 바이트의 전체 수로서 역방향 링크 요청 파라미터를 해석할 것이다.

9. 클라이언트 기능 패킷

호스트는 일반적으로 선택적이거나 원하는 방식으로 호스트 대 클라이언트 링크를 구성하도록 통신하고 있는 클라이언트(디스플레이)의 기능을 알 필요가 있다. 디스플레이가 순방향 링크 동기화가 요구된 후 호스트로 클라이언트 기능 패킷을 전송하는 것이 권장된다. 이러한 패킷의 전송은 역방향 링크 캡슐화 패킷에서 역방향 링크 플래그를 이용하는 호스트에 의해 요청될 때 요구된다. 클라이언트 기능 패킷은 호스트에게 클라이언트의 기능을 통보하기 위해 사용된다. 외부 모드인 경우, 모든 호스트는 이러한 패킷을 수신할 수 있어야 하며, 모든 클라이언트는 이러한 인터페이스 및 프로토콜을 전적으로 이용하기 위해 이러한 패킷을 전송할 수 있어야 한다. 이러한 패킷의 구현은 인터넷 모드에 대해 선택적인데, 이는 이러한 상황에서 디스플레이, 키보드 또는 다른 입/출력 장치와 같은 클라이언트의 기능이 미리 잘 한정되고 소정 타입의 단일 구성 또는 유닛으로 제조 또는 어셈블리시 호스트에 알려져야 하기 때문이다.

일 실시예에서 클라이언트 기능 패킷의 포맷은 도18에 도시된다. 도18에 도시된 바와 같이, 이러한 실시예의 경우, 이러한 패킷의 타입은 패킷 길이, 패킷 타입, 반전된 cCLIENT ID, 프로토콜 버전, 최소 프로토콜 버전, 데이터 레이트 기능, 인터페이스 타입 기능, Alt 디스플레이의 수, 예비된 1, 비트맵 폭, 비트맵 높이, 디스플레이 윈도우 폭, 디스플레이 윈도우 높이, 컬러 맵 크기, 컬러 맵 RGB 폭, RGB 기능, 색상 기능, 예비된 2, Y, Cr Cb 기능, Bayer 기능, 알파-커서 이미지 플레인, 클라이언트 특성 기능, 최대 비디오 프레임 레이트, 최소 비디오 프레임 레이트, 최소 서브 프레임 레이트, 오디오 버퍼 깊이, 오디오 채널 기능, 오디오 샘플 레이트 기능, 오디오 샘플 해상도, 최소 오디오 샘플 해상도, 최소 샘플 레이트 기능, 키보드 데이터 포맷, 포인팅 장치 데이터 포맷, 콘텐츠 보호 타입, Mfr. 이름, 제조 코드, 예비된 3, 시리얼 번호, Mfr.의 구간, Mfr.의 해, 및 CRC 필드를 갖도록 구성된다. 실시예에서, 이러한 타입의 패킷은 일반적으로 타입 66 패킷으로 식별된다.

10. 키보드 데이터 패킷

키보드 데이터 패킷은 클라이언트로부터의 키보드 데이터를 호스트로 전송하는데 사용된다. 무선(또는 유선) 키보드는 헤드 장착된 비디오 디스플레이/오디오 제공 장치를 포함하는 다양한 디스플레이 또는 오디오 장치와 관련하여 사용되지만 이에 한정되지 않는다. 키보드 데이터 패킷은 몇몇 공지된 키보드형 장치로부터 수신된 키보드 데이터를 호스트로 중계한다. 이러한 패킷은 데이터를 키보드로 전송하기 위해 순방향 링크에 사용될 수도 있다. 클라이언트는 클라이언트 기능 패킷에서 키보드 데이터 필드를 이용하여 키보드 데이터 패킷을 전송 및 수신하는 기능을 나타낸다.

키보드 데이터 패킷의 포맷은 도19에 도시되며, 키보드로부터 또는 키보드에 대한 정보의 바이트의 가변 수를 포함한다. 도19에 도시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, b클라이언트 ID, 키보드 데이터 포맷, 키보드 데이터, 및 CRC 필드를 갖도록 구성된다. 여기서, 이러한 타입의 패킷은 일반적으로 타입 67 패킷으로 식별된다.

b클라이언트 ID는 전송한 바와 같이 예비된 필드이며, CRC는 패킷의 전체 바이트에 대해 실행된다. 키보드 데이터 포맷 필드는 키보드 데이터 포맷을 설명하는 2바이트 값을 포함한다. 비트 6 내지 0은 클라이언트 기능 패킷에서 키보드 데이터 포맷 필드와 동일해야 한다. 이러한 값은 127과 동일하지 않다. 비트 15 내지 7은 미래의 사용을 위해 예비되고, 결국 현재 0으로 설정된다.

11. 포인팅 장치 데이터 패킷

포인팅 장치 데이터 패킷은 무선 마우스 또는 다른 포인팅 장치로부터의 위치 정보를 클라이언트로부터 호스트로 전송하기 위한 방법, 구조 또는 수단으로 사용된다. 데이터는 이러한 패킷을 사용하여 순방향 링크 상의 포인팅 장치로 송신될 수 있다. 포인팅 장치 데이터 패킷의 포맷의 예는 도20에 도시되며, 포인팅 장치로부터 또는 그에 대한 정보의 바이트의 가변 수를 포함한다. 도20에 도시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 캐럿 타입, b클라이언트 ID, 포인팅 장치 포맷, 포인팅 장치 데이터, 및 CRC 필드를 갖도록 구성된다. 실시예에서, 이러한 타입의 패킷은 일반적으로 1-바이트 타입 필드에서 타입 68 패킷으로 식별된다.

12. 링크 쉼다운 패킷

링크 쉼다운 패킷은 MDDI 데이터 및 스트로브가 쉼다운되고 저전력 소비 "동면" 상태로 진입할 것을 나타내는 방법 또는 수단으로서 호스트로부터 클라이언트로 전송된다. 이러한 패킷은 링크를 쉼다운하는데 유용하며, 정적 비트맵이 이동통신 장치로부터 디스플레이로 전송된 후 또는 당분간 호스트로부터 클라이언트로 전송할 추가의 정보가 없는 경우, 전력을 보존한다. 정상 동작은 호스트가 패킷들을 다시 전송하는 경우에 재개된다. 절전 모드 후의 제 1 패킷은 서브 프레임 헤더 패킷이다. 일 실시예에 대한 클라이언트 상태 패킷의 포맷은 도 21에 제시된다. 도 21에 제시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, CRC 필드 및 모두 제로 필드들을 갖도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 일반적으로 1-바이트 타입 필드에서 타입 69 패킷으로 식별된다.

패킷 길이 필드는 패킷 길이 필드를 포함하지 않는 패킷에서 전체 바이트의 수를 설명하기 위해 2바이트를 사용한다. 일 실시예에서, 이러한 패킷의 패킷 길이는 링크 쉼다운 패킷이 전송된 시간에서 효율적으로 인터넷 타입 또는 링크 모드에 의존한다. 따라서, 통상의 패킷 길이는 타입 1 모드에 대해 20바이트(패킷에서 총 22바이트), 타입 2 모드에 대해 36바이트(패킷에서 총 38바이트), 타입 3 모드 링크에 대해 68바이트(패킷에서 총 70바이트), 및 타입 4 모드에 대해 132 바이트(패킷에서 총 134 바이트)가 된다.

모든 제로 필드는 MDDI_Data 신호가 충분한 시간 동안 로직 제로 레벨에 있어서 클라이언트가 호스트의 라인 드라이버를 디스에이블하기 전에 단지 MDDI_Stb를 이용하여 클록을 복구하기 시작하는 것을 보장하기 위해 바이트의 가변 수를 사용한다. 모든 제로 필드의 길이는 링크 쉼다운 패킷이 전송된 시간에 효율적으로 인터페이스 타입 또는 링크 연산 모드에 의존한다. 모든 제로 필드의 길이는 소정의 인터페이스 타입 세팅에 대해 MDDI_Stb에 대한 64 펄스를 생성하도록 의도된다. 따라서, 각각의 인터페이스 타입에 대한 모든 제로 길이는 타입1에 대해 16바이트, 타입2에 대해 32바이트, 타입3에 대해 64바이트, 및 타입4에 대해 128바이트가 된다.

CRC 필드는 패킷 길이로부터 패킷 타입까지의 바이트의 16비트 CRC를 포함하는 2바이트를 사용한다.

저전력 동면 상태에서, MDDI_Data() 드라이버는 모든 제로 필드의 최종 비트 이후 48번째 MDDI_Stb 사이클에 대해 16 번째 이후 시작하는 높은 임피던스 상태로 디스에이블된다. 타입-2, 타입-3, 또는 타입-4 링크에 대해, MDDI_Data 1 내

지 MDDL_DataPwr 7 신호들은 높은 임피던스 상태로 배치되는데, 동시에 MDDL_Data() 드라이버는 디스에이블된다. 호스트 또는 클라이언트는 MDDL 링크가 동면 상태에서부터 깨어나도록 할 수도 있는데, 이는 본 발명의 장점을 위한 핵심 정보 사항이다.

모든 제로 필드의 한정에서 설명한 바와 같이, MDDL_Stb는 클라이언트 제어기에서 정연하게 셋다운을 용이하게 하기 위해 링크 셋다운 패킷의 CRC 필드의 MSB를 따르는 64 사이클에 대해 토글한다. 일 사이클은 하이-투-로우 전이에 앞서는 로우-투-하이 전이이거나, 로우-투-하이 전이에 앞서는 하이-투-로우 전이이다. 모든 제로 필드가 전송된 후, 호스트에서 MDDL_Stb 드라이버는 디스에이블된다.

13. 클라이언트 요청 및 상태 패킷들

호스트는 일반적으로 최적의 방식으로 호스트-클라이언트 링크를 구성하기 위해서 클라이언트로부터 적은 양의 정보를 필요로 한다. 클라이언트는 하나의 클라이언트 요청 및 상태 패킷을 호스트로 각각의 서브 프레임에서 전송하는 것이 바람직하다. 클라이언트는 이러한 패킷이 신뢰성 있게 호스트로 전달되는 것을 보장하기 위해서 역방향 링크 캡슐화 패킷의 제 1 패킷으로서 이러한 패킷을 전송하여야 한다. 이러한 패킷의 전송은 또한 역방향 링크 캡슐화 패킷의 역방향 링크 플래그들을 사용하여 호스트에 의해 요청되는 경우에 또한 수행된다. 클라이언트 요청 및 상태 패킷은 호스트로 에러 및 상태를 보고하는데 사용된다. 외부 모드 동작에 대해, 모든 호스트는 이러한 패킷을 수신할 수 있어야 하며, 모든 클라이언트는 적절하게 또는 최적으로 MDD 인터페이스 프로토콜을 사용하기 위해서 이러한 패킷을 전송할 수 있어야 한다. 내부 연산을 위해 내부 호스트 및 내부 클라이언트가 권장되지만, 이러한 패킷에 대해 지원될 수 있으며, 이는 요구되지 않는다.

클라이언트 요청 및 상태 패킷의 포맷은 도 22에 제시된다. 도 22에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, cClient ID, 역방향 링크 요청, 성능 변경, 클라이언트 비지, CRC 에러 카운트, 및 CRC 필드들을 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 1 바이트 타입 필드에서 타입 70 패킷으로 식별되고, 일반적으로 미리 선택된 고정된 12 바이트 길이를 사용한다.

역방향 링크 요청 필드는 데이터를 다시 호스트로 전송하기 위해서 역방향 링크 캡슐화 패킷에서 클라이언트가 필요한 바이트들의 수를 호스트에게 통보하는데 사용된다. 호스트는 역방향 링크 캡슐화 패킷에서 적어도 그 숫자만큼의 바이트를 할당함으로써 그 요청의 허용을 시도하여야 한다. 호스트는 데이터를 수용하기 위해서 서브 프레임에서 하나 이상의 역방향 링크 캡슐화 패킷을 전송할 수도 있다. 클라이언트는 클라이언트 요청 및 상태 패킷을 임의의 시점에서 전송할 수 있고, 호스트는 하나의 서브 프레임에서 요청된 바이트들의 총 수로서 역방향 링크 요청 파라미터를 해석할 것이다. 역방향 링크 데이터가 다시 호스트로 어떻게 전송되는 지에 관한 구체적인 내용 및 예는 후술될 것이다.

14. 비트 블록 전송 패킷

비트 블록 전송 패킷은 임의의 방향에서 디스플레이 영역들을 스크롤하는 수단, 구성 또는 방법을 제공한다. 이러한 성능을 갖는 디스플레이들은 클라이언트 성능 패킷의 디스플레이 특징 성능 표시자의 비트 0에서 이러한 성능을 보고할 것이다. 비트 블록 전송 패킷의 일 실시예에 대한 포맷은 도 23에 제시된다. 도 23에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, 상부 좌측 X 값, 상부 좌측 Y 값, 윈도우 폭, 윈도우 높이, 윈도우 X 이동, 윈도우 Y 이동, 및 CRC 필드들을 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 71 패킷으로 식별되고, 일 실시예에서 미리 선택된 고정된 15 바이트 길이를 사용한다.

이러한 필드들은 이동될 윈도우의 상부 좌측 코너 좌표의 X 및 Y 값, 이동될 윈도우의 폭 및 높이, 및 픽셀들의 수를 규정하는데 사용되어 윈도우가 수평 및 수직으로 각각 이동되도록 한다. 상기 후자의 2개의 필드의 양의 값은 윈도우가 우측 및 아래로 이동하도록 하고, 음의 값은 좌측 및 상부로 이동하도록 한다.

15. 비트맵 영역 채움 패킷

비트맵 영역 채움(fill) 패킷은 디스플레이 영역을 하나의 컬러로 쉽게 초기화하는 수단, 구성 또는 방법을 제공한다. 이러한 성능을 갖는 디스플레이들은 클라이언트 성능 패킷의 클라이언트 특징 성능 표시자 필드의 비트 1에서 이러한 성능을 표시할 것이다. 비트맵 영역 채움 패킷의 포맷에 대한 일 실시예는 도 24에 제시된다. 도 24에 제시되는 바와 같이, 이러한 경우, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, 상부 좌측 X 값, 상부 좌측 Y 값, 윈도우 폭, 윈도우 높이, 데이터 포맷 기술자, 픽셀 영역 채움값, 및 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 1 바이트 타입 필드에서 타입 72 패킷으로 식별되고, 미리 선택된 고정된 17 바이트 길이를 사용한다.

16. 비트맵 패턴 채움 패킷

비트맵 패턴 채움(fill) 패킷은 디스플레이 영역을 미리 선택된 패턴으로 쉽게 초기화하는 수단 또는 구조를 제공한다. 이러한 능력을 갖는 디스플레이는 클라이언트 성능 패킷의 클라이언트 특징 성능 필드의 비트 2에서 이러한 성능을 보고할 것이다. 이러한 채움 패턴의 상부 좌측 코너는, 수평 또는 수직 패턴 오프셋이 비 제로가 아니라면, 채워질 윈도우의 상부 좌측 코너와 정렬된다. 채워질 윈도우가 채움 패턴보다 넓거나 길면, 패턴은 윈도우를 채우기 위해서 수평 또는 수직으로 여러 번 반복될 것이다. 최종 반복된 패턴의 우측 또는 바닥은 필요에 따라 잘려진다(truncated). 윈도우가 채움 패턴보다 작으면, 채움 패턴의 우측 또는 바닥은 윈도우와 맞추기 위해서 잘려진다.

만일 수평 패턴 오프셋이 비제로이면, 윈도우의 좌측 사이드와 수평 패턴 오프셋을 더한 우측 사이드 사이의 픽셀들은 패턴의 가장 우측 픽셀로 채워진다. 수평 패턴 오프셋은 패턴 폭보다 작을 것이다. 유사하게, 수직 패턴 오프셋이 비제로이면, 윈도우의 상부측과 수직 패턴 오프셋을 더한 측면의 상부 사이의 픽셀은 가장 낮은 패턴의 픽셀로 채워진다. 수직 패턴 오프셋은 패턴 높이보다 적을 것이다.

비트맵 패턴 채움 패킷의 포맷에 대한 일 실시예는 도 25에 제시된다. 도 25에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, 상부 좌측 X 값, 상부 좌측 Y 값, 윈도우 폭, 윈도우 높이, 패턴 폭, 패턴 높이, 수평 패턴 오프셋, 수직 패턴 오프셋, 데이터 포맷 기술자, 파라미터 CRC, 패턴 픽셀 데이터, 및 픽셀 데이터 CRC 필드를 갖도록 구성된다. 소정의 실시예에서, 이러한 타입의 패킷은 일반적으로 1 바이트 타입 필드에서 타입 73 패킷으로 식별된다.

17. 통신 링크 데이터 채널 패킷

통신 링크 데이터 채널 패킷은 PDA와 같은 고-레벨 계산 성능을 갖는 클라이언트가 셀룰러 전화 또는 무선 데이터 포트 장치와 같은 무선 트랜시버와 통신하는 구성, 수단 또는 방법을 제공한다. 이러한 상황에서, MDDI 링크는 통신 장치와 이동 디스플레이를 갖는 계산 장치 사이의 편리한 고속 인터페이스로서 동작하고, 여기서 이러한 패킷은 그 장치에 대한 운영 시스템의 데이터 링크 계층에서 데이터를 전달한다. 예를 들어, 웹 브라우저, 이메일 클라이언트, 또는 PDA가 이동 디스플레이 내에 구축되는 경우, 이러한 패킷이 사용될 수 있다. 이러한 성능을 갖는 디스플레이들은 클라이언트 성능 패킷의 클라이언트 특징 성능 필드의 비트 3에서 이러한 성능을 보고할 것이다.

통신 링크 데이터 채널 패킷에 대한 일 실시예의 포맷은 도 26에 제시된다. 도 26에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, 파라미터 CRC, 통신 링크 데이터, 및 통신 데이터 CRC 필드를 갖도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 일반적으로 타입 필드에서 타입 74 패킷으로 식별된다.

18. 인터페이스 타입 핸드오프 요청 패킷

인터페이스 타입 핸드오프 요청 패킷은 호스트로 하여금, 클라이언트 또는 디스플레이가 기존 또는 현재 모드에서 타입 1(직렬), 타입 2(비트 병렬), 타입 3(4 비트 병렬), 또는 타입 4(8 비트 병렬) 모드로 이동하도록 하는 요청을 할 수 있도록 하는 수단, 방법 또는 구조를 제공한다. 호스트가 특정 모드를 요청하기 전에, 호스트는 클라이언트 성능 패킷의 디스플레이 특징 성능 표시자의 비트 6 및 7을 검사하여 클라이언트가 요구되는 모드에서 동작할 수 있다는 것을 확인하여야 한다. 인터페이스 타입 핸드오프 요청 패킷의 포맷에 대한 일 실시예는 도 27에 제시된다. 도 27에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, 인터페이스 타입, 예약 1, 및 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 75 패킷으로서 식별되고, 미리 선택된 고정된 4 바이트 길이를 사용한다.

19. 인터페이스 타입 확인 패킷

인터페이스 타입 확인(acknowledge) 패킷은 클라이언트에 의해 전송되고, 클라이언트가 인터페이스 타입 핸드오프 패킷의 수신을 확인하게 하는 수단, 방법, 또는 구조를 제공한다. 요청된 모드, 타입 1(직렬), 타입 2(2비트 병렬), 타입 3(4 비트 병렬), 또는 타입 4(8비트 병렬) 모드는 이러한 패킷에서 파라미터로서 호스트로 다시 에코(echo)된다. 인터페이스 타입 확인 패킷에 대한 일 실시예의 포맷은 도 28에 제시된다. 도 28에 제시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, cClient ID, 인터페이스 타입, 예약 1, 및 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 76으로 식별되고, 미리 선택된 고정된 4 바이트 길이를 사용한다.

20. 실행 타입 핸드오프 패킷

실행(perform) 타입 핸드오프 패킷은 호스트가, 이동국으로 하여금 이러한 패킷에 규정된 모드로 핸드오프 하도록 명령하는 수단이다. 이는 인터페이스 타입 핸드오프 요청 패킷 및 인터페이스 타입 확인 패킷에 의해 이전에 요청되고 확인된 것과 동일한 모드이다. 호스트 및 클라이언트는 이러한 패킷이 전송된 후에 동의된 모드로 스위칭하여야 한다. 클라이언트는 이러한 모드 변경 동안 링크 동기를 상실하고 재-획득할 수도 있다. 실행 타입 핸드오프 패킷의 포맷은 도 29에 제시된다. 도 29에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, 예약 1, 및 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 1 바이트 타입 필드에서 타입 77로 식별되고, 미리 선택된 고정된 4 바이트 길이를 사용한다.

21. 순방향 오디오 채널 인에이블 패킷

이 패킷은 호스트가 클라이언트의 오디오 채널들을 인에이블 또는 디세이블 할 수 있도록 하는 구조, 방법, 또는 수단을 제공한다. 이러한 성능은 호스트에 의해 출력될 오디오가 존재하지 않는 경우, 전력을 보존하기 위해서 클라이언트(예를 들어, 디스플레이)가 오디오 증폭기를 턴-오프할 수 있기 때문에 유용하다. 이는 표시자로서 오디오 스트림의 존재 또는 부재를 함축적으로 간단히 사용하여 구현하기에는 매우 어렵다. 클라이언트 시스템의 전력이 상승되는 경우, 디폴트 상태에서 모든 오디오 채널들이 인에이블된다. 순방향 오디오 채널 인에이블 패킷에 대한 일 실시예의 포맷은 도 30에 제시된다. 도 30에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, 오디오 채널 인에이블 마스크, 및 CRC 필드들을 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 1 바이트 타입 필드에서 타입 78 패킷으로 식별되고, 미리 선택된 고정된 4 바이트 길이를 사용한다.

22. 역방향 오디오 샘플 레이트 패킷

이 패킷은 호스트가 역방향 링크 오디오 채널을 인에이블 또는 디세이블하고, 이러한 스트림의 오디오 데이터 샘플 레이트를 설정할 수 있도록 하여준다. 호스트는 클라이언트 성능 패킷에서 유효하다고(valid) 정의되는 샘플 레이트를 선택한다. 호스트가 유효하지 않은 샘플 레이트를 선택하면, 클라이언트는 오디오 스트림을 호스트로 전송하지 않고, 적절한 에러, 에러 값, 또는 에러 신호는 클라이언트 에러 보고 패킷에서 호스트로 전송될 것이다. 호스트는 샘플 레이트를 255 값으로 설정함으로써 역방향 링크 오디오 스트림을 디세이블할 수 있다. 클라이언트가 초기에 전력-증가(powered-up) 또는 연결시에 가정되는 디폴트 상태는 디세이블되는 역방향 링크 오디오 스트림을 사용한다. 역방향 오디오 샘플 레이트 패킷에 대한 일 실시예의 포맷은 도 31에 제시된다. 도 31에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, 오디오 샘플 레이트, 예약 1, 및 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 79 패킷으로 식별되고, 미리 선택된 고정된 4 바이트 길이를 사용한다.

23. 디지털 콘텐츠 보호 오버헤드 패킷

이 패킷은 사용되는 디지털 콘텐츠 보호 방법에 관련된 메시지들을 호스트 및 클라이언트가 교환하도록 하게 하는 구조, 방법 또는 수단을 제공한다. 현재 차후의 대안적인 보호 방식 지정을 위해 예약된 룸(room)을 갖는 2가지 타입의 콘텐츠 보호(디지털 전송 콘텐츠 보호(DTCP) 또는 고-대역폭 디지털 콘텐츠 보호 시스템(HDCP))가 고려된다. 사용되는 방법은 이러한 패킷에서 콘텐츠 보호 타입 파라미터에 의해 규정된다. 디지털 콘텐츠 보호 오버헤드 패킷의 일 실시예의 포맷은 도 32에 제시된다. 도 32에 제시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, bClient ID, 콘텐츠 보호 타입, 콘텐츠 보호 오버헤드 메시지, 및 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 80 패킷으로 식별된다.

24. 투명한 컬러 인에이블 패킷

투명한(transparent) 컬러 인에이블 패킷은 디스플레이에서 어떤 컬러가 투명한지를 규정하고, 이미지를 디스플레이하는데 투명한 컬러의 사용을 인에이블 또는 디세이블하기 위해서 사용되는 구조, 방법 또는 수단이다. 이러한 성능을 갖는 디스플레이들은 클라이언트 성능 패킷의 클라이언트 특징 성능 필드의 비트 4에서 이러한 성능을 보고할 것이다. 투명한 컬러에 대한 값을 갖는 픽셀이 비트맵에 기록되면, 그 컬러는 이전 값으로부터 변경되지 않는다. 투명한 컬러 인에이블 패킷의 포맷은 도 33에 제시된다. 도 33에 제시되는 바와 같이, 일 실시예에서, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, 투명한 컬러 인에이블, 예약 1, 알파 커서(Alpha-Cursor) 식별자, 데이터 포맷 기술자, 투명한 픽셀 값, 및 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 1 바이트 타입 필드에서 타입 81로 식별되고, 미리 선택된 고정된 10 바이트 길이를 사용한다.

25. 왕복 지연 측정 패킷

왕복 지연 측정 패킷은 호스트로부터 클라이언트(디스플레이)로의 전파 지연에, 클라이언트(디스플레이)로부터 호스트로의 지연을 더한 전파 지연을 측정하는데 사용되는 구조, 방법, 또는 수단을 제공한다. 이러한 측정치는 본질적으로 라인 드라이버 및 수신기, 그리고 상호연결 서브-시스템에 존재하는 지연들을 포함한다. 이러한 측정치는 전환 지연 및 상술한 역방향 링크 캡슐화 패킷의 역방향 링크 레이트 제수(divisor) 파라미터들을 설정하는데 사용된다. 이러한 패킷은 MDDI 링크가 특정 애플리케이션에 의도된 최대 속도에서 동작되는 경우에 특히 유용하다. 패킷은 왕복 지연 측정의 영역을 증가시키기 위해 타입 1 모드로, 그리고 낮은 데이터 레이트로 전송될 수도 있다. MDDI_Stb 신호는 모두 제로 데이터가 뒤이은 필드들 동안 전송되는 것처럼 행동한다: 가드 타임, 모두 제로, 및 측정 주기. 이는 MDDI_Stb가 그 데이터 전송물의 절반에서 토글링하도록 하여, 측정 주기 동안 클라이언트에서 주기적인 클럭으로 사용될 수 있게 된다.

일 실시예에서, 클라이언트는 일반적으로 클라이언트 성능 패킷의 클라이언트 특정 성능 필드의 18 비트를 사용하여 왕복 지연 측정 패킷을 지원하는 능력을 표시한다. 모든 클라이언트들이 왕복 지연 측정을 지원하는 것이 바람직하지만, 호스트가 최대 케이블 지연 및 최대 드라이버 및 수신기 지연들에 기초하여 최악의 왕복 지연을 파악하는 것 역시 가능하다. 호스트는 또한 내부 모드에서 사용되는 링크에 대해 미리 왕복 지연을 파악할 수 있는데, 왜냐하면 이는 인터페이스가 사용되는 장치의 알려진 설계 엘리먼트(컨덕터 길이, 회로 타임, 및 특징 등)에 기반하기 때문이다.

왕복 지연 측정 패킷의 포맷은 도 34에 제시된다. 도 34에 제시되는 바와 같이, 일 실시예에서, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, 파라미터 CRC, 가드 타임 1, 측정 주기, 모두 제로, 및 가드 타임 2를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 82 패킷으로 식별되며, 미리 선택된 고정된 533 비트 길이를 사용한다.

왕복 지연 측정 패킷 동안 발생하는 이벤트들의 타이밍은 도 35에 제시된다. 도 35에서, 호스트는 왕복 지연 측정 패킷을 전송하고, 이는 파라미터 CRC 및 스트로브 정렬 필드들의 존재에 의해 제시되며, 모두 제로 및 가드 타임 1 필드가 뒤따른다. 지연 3502는 패킷이 클라이언트 장치 또는 처리 회로에 도달하기 전에 발생한다. 클라이언트가 그 패킷을 수신하면, 클라이언트는 클라이언트에 의해 결정되는 측정 주기의 시작부에서 0xff, 0xff, 및 30 바이트 패턴을 실제로 정확하게 전송한다. 클라이언트가 이러한 시퀀스 전송을 시작하는 실제 시간은 호스트의 관점에서는 측정 주기의 시작부로부터 지연된다. 이러한 지연량은 패킷이 라인 드라이버 및 수신기, 그리고 상호연결 서브시스템(케이블, 컨덕터)을 통해 전파하는데 걸리는 시간이다. 클라이언트로부터 호스트로 상기 패턴이 전파되는데 유사한 지연량 3504가 발생한다.

클라이언트로, 그리고 클라이언트로부터 전달되는 신호들에 대한 왕복 지연 시간을 정확하게 결정하기 위해서, 호스트는 측정 주기의 시작 후에 0xff, 0xff, 및 30 바이트 0x0 시퀀스의 시작이 도달 탐지될 때까지 발생하는 순방향 링크 비트 시간 주기들의 수를 카운트한다. 이러한 정보는 호스트로부터 클라이언트로, 그리고 다시 클라이언트로부터 호스트로 전달되는 왕복 신호의 시간량을 결정하는데 사용된다. 그리고 나서, 이러한 양의 절반이 클라이언트로의 일 방향 전달에서 발생하는 지연으로 간주된다.

호스트 및 클라이언트 모두는 가드 기간들 동안 MDDI_DATA 라인을 정의된 상태로 유지하기 위해서 라인을 논리-제로 레벨로 구동한다. 보호 시간들 동안 호스트 및 클라이언트의 인에이블 및 디스에이블 시간은 MDDI_Data 신호가 소정의 유효한 왕복 지연 시간에 대해 유효한 낮은 레벨이 되는 것이다.

26. 순방향 링크 왜곡 교정 패킷

순방향 링크 왜곡 교정(calibration) 패킷은 MDDI_Stb 신호에 대한 MDDI_Data 신호들의 전파 지연에서의 차이에 대해 클라이언트 또는 디스플레이가 그 자신을 교정하도록 하여준다. 지연 왜곡 보상이 없다면, 최대 데이터 전송률은 일반적으로 이러한 지연들에서의 잠재적인 최악의 변동을 고려하기 위해서 제한된다. 일반적으로, 이러한 패킷은 순방향 링크 데이터 전송률이 대략 50Mbps 또는 그 이하의 레이트로 구현되는 경우에만 전송된다. 디스플레이를 교정하기 위해서 이러한 패킷을 전송한 후에, 데이터 전송률은 50Mbps 이상으로 증가될 수 있다. 데이터 전송률이 이러한 왜곡 교정 처리기간 동안 너무 높게 설정되면, 디스플레이는 1 비트 타임 이상으로 지연 왜곡 보상이 오프(off)될 수 있는 비트 주기의 앨리어스(alias)와 동기될 수 있고, 이는 에러 데이터 클럭킹을 초래한다. 인터페이스의 가장 높은 데이터 전송률 타입 또는 가장 큰 가능한 인터페이스 타입은 모든 기존 데이터 비트들이 교정되도록 순방향 링크 왜곡 교정 패킷을 전송하기에 앞서 선택된다.

순방향 링크 왜곡 교정 패킷의 포맷의 일 실시예는 도 56에 제시된다. 도 56에 제시된 바와 같이, 이러한 타입의 패킷은 패킷 길이(2 바이트), 패킷 타입, hClient ID, 파라미터 CRC, 모두 제로, 교정 데이터 시퀀스, 및 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 필드에서 타입 83으로 식별되고, 일 실시예에서 미리 선택된 고정된 5115 길이를 갖는다.

가상 제어 패널

가상 제어 패널(VCP)은 호스트가 클라이언트에서 임의의 사용자 제어를 설정할 수 있도록 하여준다. 이러한 파라미터들이 호스트에 의해 조정될 수 있도록 함으로써, 클라이언트의 사용자 인터페이스는 간략화될 수 있는데, 왜냐하면 사용자가 오디오 볼륨 또는 디스플레이 밝기와 같은 파라미터들을 조정하도록 하는 스크린들이 클라이언트의 하나 이상의 마이크로프로세서에 의해서가 아니라 호스트 소프트웨어에 의해 발생할 수 있기 때문이다. 호스트는 클라이언트에서 파라미터 세팅들을 판독하고, 각 제어에 대한 유효값의 범위를 결정하는 능력을 갖는다. 클라이언트는 통상적으로 어떤 제어 파라미터들이 조정될 수 있는지를 호스트로 보고하는 성능을 갖는다.

제어 코드들(VCP 코드들) 및 일반적으로 규정되는 관련 데이터 값들은 클라이언트에서 제어 및 세팅들을 규정하는데 사용된다. MDDI 규격에서의 VCO 코드들은 16비트로 확장되어 패킷 정의에서 적절한 데이터 필드 정렬을 보존하고, 차후에 이러한 인터페이스 또는 차후 개선된 인터페이스에 고유한 보충(supplementary) 값들을 지원한다.

27. 요청 VCP 피쳐 패킷

요청 VCP 피쳐 패킷은 특정 제어 파라미터 또는 모든 유효 제어 파라미터들의 현재 세팅을 호스트가 요청하는 수단, 메커니즘, 또는 방법을 제공한다. 일반적으로, 클라이언트는 VCP 피쳐 응답 패킷에서 적절한 정보를 가지고 VCP 패킷에 응답한다. 일 실시예에서, 클라이언트는 클라이언트 성능 패킷의 클라이언트 특징 성능 표시자의 비트 20을 사용하여 요청 VCP 피쳐 패킷을 지원하는 능력을 표시한다.

일 실시예에서, 요청 VCP 피쳐 패킷의 포맷은 도 69에 제시된다. 도 69에 제시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, MCCS VCP 코드, 및 CRC 필드를 갖도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 일반적으로 타입 128로 식별되고, 이는 2 바이트 타입 필드에서 표시된다. 패킷 길이 필드를 포함하지 않는 패킷에서의 비트들의 총 수를 규정하는 패킷 길이는 일반적으로 8 바이트 길이에서 이러한 타입의 패킷에 대해 고정된다.

hClient ID 필드는 미래의 구현에서 Client ID로 사용하기 위해 예비되며, 통상적으로 0으로 설정된다. MCCS VCP 코드 필드는 MCCS VCP 제어 코드 파라미터를 규정하는 2 바이트 정보를 포함한다. 0-255 범위의 값은 규정된 MCCS 코드에 대응하는 VCP 피쳐 응답 리스트에서 하나의 아이템을 가지고 VCP 피쳐 응답 패킷이 리턴되도록 한다. 65535(0xffff) MCCS VCP 코드는 클라이언트에 의해 지원되는 각각의 제어에 대한 피쳐 응답 리스트 아이템을 포함하는 VCP 피쳐 응답을 가지고 VCP 피쳐 응답 패킷을 요청한다. 이러한 필드에 대한 256-65534 값들은 차후 사용을 위해 예약되고 현재는 사용되지 않는다.

28. VCP 피쳐 응답 패킷

VCP 피쳐 응답 패킷은 클라이언트가 특정 제어 파라미터 또는 모든 유효 제어 파라미터들의 현재 세팅으로 호스트 요청에 응답하는 수단, 메커니즘, 또는 방법을 제공한다. 일반적으로, 클라이언트는 요청 VCP 피쳐 패킷에 응답하여 VCP 피쳐 응답 패킷을 전송한다. 이러한 패킷은 특정 파라미터의 현재 세팅을 결정하고, 특정 제어에 대한 유효 범위를 결정하고, 특정 제어가 클라이언트에 의해 지원되는지 여부를 결정하고, 또는 클라이언트에 의해 지원되는 제어들 세트를 결정하는데 유용하다. 클라이언트에서 실행되지 않는 특정 제어를 참조하는 요청 VCP 특징이 전송되면, VCP 피쳐 응답 패킷은 적절한 에러 코드를 포함하는 실행되지 않는 제어에 대응하는 하나의 VCP 피쳐 응답 리스트 아이템을 가지고 리턴된다. 일 실시예에서, 클라이언트는 클라이언트 성능 패킷의 디스플레이 특징 성능 표시자 필드의 비트 13을 사용하여 VCP 피쳐 응답 패킷을 지원하는 능력을 표시한다.

일 실시예에서, VCP 피쳐 응답 패킷의 포맷은 도 70에 제시된다. 도 70에 제시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, cClient ID, MCCS 버전, 응답 시퀀스 번호, VCP 피쳐 응답 리스트 및 CRC 필드를 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 129로 식별되고, 이는 2 바이트 타입 필드에서 표시된다.

cClient ID 필드는 클라이언트 ID에 대해 예약된 정보를 포함한다. 이러한 필드는 차후 사용을 위해 예약되고, 일반적으로 0으로 설정된다. MCCS 버전 필드는 클라이언트에 의해 실행되는 VESA MCCS 규격의 버전의 규정하는 2 바이트 정보를 포함한다.

2 바이트 응답 시퀀스 번호 필드는 클라이언트에 의해 리턴되는 VCP 피쳐 응답 패킷들의 시퀀스 번호를 규정하는 정보 또는 데이터를 포함한다. 클라이언트는 MCCS 제어 코드 값 65535를 가지고 요청 VCP 피쳐 패킷에 응답하여 하나 또는 그

이상의 VCP 피쳐 응답 패킷들을 리턴한다. 클라이언트는 다수의 VCP 피쳐 응답 패킷들에 걸쳐 피쳐 응답 리스트를 확산시킬 수도 있다. 이 경우, 클라이언트는 시퀀스 번호를 각각의 연속적인 패킷에 할당하고, 하나의 요청 VCP 피쳐 패킷에 응답하여 전송되는 VCP 피쳐 응답 패킷들의 시퀀스 번호들은 0에서 시작하여 1 씩 증가한다. 최종 VCP 피쳐 응답 패킷의 최종 VCP 특징 리스트 아이템은 그 패킷이 최종 패킷이고, 리턴되는 패킷 그룹 중 가장 높은 시퀀스 번호를 갖는다는 것을 표시하기 위해서 0xffff와 동일한 MCCS VCP 제어 코드 값을 포함하여야 한다. 만약 1개의 VCP 피쳐 응답 패킷이 요청 VCP 피쳐 패킷에 응답하여 전송되면, 그 하나의 패킷 내의 응답 시퀀스 번호는 0이고, VCP 피쳐 응답 리스트는 0xffff와 동일한 MCCS VCP 제어 코드를 갖는 기록을 포함한다.

리스트 필드의 특징들의 수는 이 패킷에서 VCP 피쳐 응답 리스트에 존재하는 VCP 특징 리스트 아이템들의 수를 규정하는 2 바이트를 포함하고, VCP 피쳐 응답 리스트 필드는 하나 이상의 VCP 피쳐 응답 리스트 아이템들을 포함하는 바이트들 그룹이다. 일 실시예에서, 하나의 VCP 피쳐 응답 리스트 아이템의 포맷이 도 71에 제시된다.

도 71에 제시되는 바와 같이, 각각의 VCP 피쳐 응답 리스트 아이템은 그 길이가 12 바이트이고, MCCS VCP 코드, 결과 코드, 최대 값, 및 현재 값 필드들을 포함한다. 2 바이트 MCCS VCP 코드 필드는 이 리스트 아이템과 관련된 MCCS VCP 제어 코드 파라미터를 규정하는 데이터 또는 정보를 포함한다. VESA MCCS 규격 버전 2 및 후속 버전에서 정의되는 제어 코드값들만이 본 실시예의 경우 유효한 것으로 간주된다. 2 바이트 결과 코드 필드는 규정된 MCCS VCP 제어에 대한 정보 요청과 관련된 에러 코드를 규정하는 정보를 포함한다. 이 필드에서 값 '0'은 에러가 없음을 표시하고, 값 '1'은 규정된 제어가 클라이언트에서 실행되지 않는다는 것을 의미한다. 2 내지 65535의 이러한 필드에 대한 추가적인 값들이 차후 사용 및 다른 애플리케이션에서의 실행을 위해 현재 예약되지만, 지금을 위해서는 사용되지 않는다.

4 바이트 최대값 필드는 규정된 MCCS 제어가 설정될 수 있는 가장 큰 가능한 값을 규정하는, 32 비트의 부호를 갖지 않는 정수를 포함한다. 요청된 제어가 클라이언트에서 실행되지 않으면, 이 값은 0으로 설정된다. 리턴되는 값이 그 길이에 있어서 32 비트(4 바이트)보다 작으면, 이 값은 최상위 바이트(미사용)들을 0으로 설정함으로써 32 비트 정수로 만들어진다. 4 바이트 현재값 필드는 규정된 MCCS VCP 연속(C) 또는 불-연속(NC) 제어에 대한 현재값을 규정하는 정보를 포함한다. 요청된 제어가 클라이언트에서 실행되지 않거나, 제어가 실행되지만 테이블(T) 데이터 타입인 경우, 이 값은 0으로 설정된다. 리턴되는 값이 VESA MCCS 규격(specification) 당 그 길이에 있어서 32 비트(4 바이트)보다 작으면, 최상위 바이트(미사용)들을 0으로 설정함으로써 32 비트 정수로 만들어진다.

29. 설정 VCP 피쳐 패킷

설정 VCP 피쳐 패킷(Set VCP Feature Packet)은 클라이언트에서 연속 및 비-연속 제어들 모두에 대한 VCP 제어 값들을 호스트가 설정(set)하는 수단, 메커니즘, 또는 방법을 제공한다. 일 실시예에서, 클라이언트는 클라이언트 성능 패킷의 디스플레이 특징 성능 표시자 필드의 비트 20을 사용하여 설정 VCP 피쳐 패킷을 지원하는 능력을 표시한다.

일 실시예에서 설정 VCP 피쳐 패킷의 포맷은 도 72에 제시된다. 도 72에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, MCCS VCP 코드, 리스트의 값들의 수, 제어 값 리스트 및 CRC 필드들을 갖도록 구성된다. 이러한 타입의 패킷은 일반적으로 타입 130으로 식별되고, 2 바이트 타입 필드에서 표시되며, 패킷 길이 필드에 독점적인 20 바이트 길이이다.

hClient ID 필드는 다시 클라이언트 ID를 규정하고, 클라이언트 ID로서 동작하는 1-바이트 값을 사용한다. 이러한 필드는 차후 사용을 위해 예약되고 현재 0으로 설정된다. MCCS VCP 코드 필드는 조정될 MCCS VCP 제어 코드 파라미터를 규정하기 위해 2 바이트 정보 또는 값들을 사용한다. 리스트 필드의 값들의 2 바이트 수는 제어 값 리스트에 존재하는 16 바이트 값들의 수를 규정하는 정보 또는 값들을 포함한다. MCCS 제어 코드가 클라이언트의 테이블과 관련되지 않으면, 제어 값 리스트는 일반적으로 하나의 아이템을 포함한다. 비-테이블 관련 제어의 경우에 있어서, 제어 값 리스트는 MCCS VCP 코드 필드에 의해 규정되는 제어 파라미터에 기록되는 새로운 값을 규정하는 값을 포함할 것이다. 테이블-관련 제어에 있어서, 제어 값 리스트의 데이터 포맷은 규정된 MCCS VCP 코드의 파라미터 설명에 의해 규정된다. 리스트가 1 바이트 이상의 값들을 포함하면, 최하위 바이트가 먼저 전송되고, 이는 다른 곳에서 정의된 방법과 일치한다. 마지막으로, 2-바이트 CRC 필드는 패킷 길이를 포함하는 패킷 내의 모든 바이트들 중 16비트 CRC를 포함한다.

30. 요청 유효 파라미터 패킷

요청 유효 파라미터 패킷은 규정된 비-연속(NC) 또는 테이블 제어에 의해 지원되는 파라미터 리스트를 포함하는 유효 파라미터 응답 패킷을 클라이언트가 리턴하도록 요청하기 위해 유효한 수단 또는 구조로서 사용된다. 이러한 패킷은 단지 클라이언트의 테이블에 관련되는 비-연속 제어 또는 제어만을 규정하여야 하고, 모든 제어들을 규정하기 위해서 65535

(0xffff) MCCS VCP 코드 값을 규정하지는 않는다. 비-지원 또는 무효(Invalid) MCCS VCP 코드가 규정되면, 적절한 예러값이 유효 파라미터 응답 패킷에서 리턴된다. 일 실시예에서, 클라이언트는 디스플레이 성능 패킷의 디스플레이 특징 성능 표시자 필드의 비트 13을 이용하여 요청 유효 파라미터 패킷을 지원하는 능력을 표시한다.

일 실시예에서, 요청 유효 파라미터 패킷의 포맷은 도 73에 제시된다. 도 73에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, MCCS VCP 코드, 및 CRC 필드를 갖도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 일반적으로 타입 131로 식별되고, 2 바이트 타입 필드에서 표시된다.

2 바이트 패킷 길이 필드에서 표시되는 패킷 길이는 일반적으로 8의 패킷 길이 필드를 포함하지 않고, 패킷에서 총 바이트 수를 갖도록 설정된다. hClient ID는 다시 클라이언트 ID를 규정하지만, 차후 사용을 위해 현재 예약되고, 당업자에게 명백한 바와 같이, 0으로 설정된다. 2 바이트 MCCS VCP 코드 필드는 질의 될 비-연속 MCCS VCP 제어 코드 파라미터를 규정하는 값을 포함한다. 이러한 필드의 값은 클라이언트에서 실행되는 비-연속 제어에 대응된다. 256 내지 65535 (0xffff) 값들은 일반적으로 예약되고, 무효(Invalid)인 것으로 간주되며, 예러 응답에서 비실행된 제어로서 간주된다.

31. 유효 파라미터 응답 패킷

유효 파라미터 응답 패킷은 요청 유효 파라미터 패킷에 응답하여 전송된다. 이는 테이블 콘텐츠를 리턴하는 비-연속 MCCS VCP 제어 또는 제어에 대한 유효 세트들을 식별하는 수단, 방법, 또는 메커니즘으로서 사용된다. 제어가 클라이언트의 테이블과 관련되면, VCP 파라미터 응답 리스트는 단순히 요청된 시퀀스 테이블 값들의 특정 리스트를 포함한다. 테이블의 콘텐츠가 하나의 유효 파라미터 응답 패킷에 적합하지 않으면, 시퀀스 응답 시퀀스 번호들을 갖는 다수의 패킷이 클라이언트에 의해 전송될 수 있다. 일 실시예에서, 클라이언트는 디스플레이 성능 패킷의 디스플레이 특징 성능 표시자 필드의 비트 13을 사용하여 유효 파라미터 응답 패킷을 지원하는 능력을 표시한다.

호스트는 다음 방식으로 테이블 콘텐츠를 요청할 수 있다: 호스트는 기록/판독 파라미터, LUT 오프셋, 및 RGB 섹션과 같은 필요한 또는 요구되는 파라미터들을 포함하는 설정 VCP 피쳐 패킷을 전송한다: 그리고 나서, 요구되는 제어를 규정하는 요청 유효 파라미터 패킷이 호스트에 의해 전송된다: 그리고 나서, 클라이언트는 테이블 데이터를 포함하는 하나 이상의 유효 파라미터 응답 패킷들을 리턴한다. 이러한 동작 시퀀스는 MCCS 동작 모델에서 설명되는 테이블 판독 기능들과 유사한 기능을 수행한다.

특정 클라이언트 파라미터가 클라이언트에 의해 지원되지 않으면, 일 실시예에서 이러한 패킷의 대응하는 필드는 255 값을 포함할 것이다. 클라이언트에서 사용되는 파라미터들에 있어서, 대응하는 필드는 클라이언트의 파라미터 값을 포함하여야 한다.

일 실시예에서, 유효 파라미터 응답 패킷의 포맷은 도 74에 제시된다. 도 74에 제시되는 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, cClient ID, VCCS VCP 코드, 응답 코드, 시퀀스 번호, 리스트의 번호 값들, VCP 파라미터 응답 리스트, 및 CRC 필드들을 갖도록 구성된다. 일 실시예에서, 이러한 타입의 패킷은 일반적으로 타입 132로서 식별되고, 2 바이트 타입 필드에서 표시된다.

cClient ID 필드는 상술한 바와 같이 차후 클라이언트 ID를 위해 예약되고, 3 바이트 MCCS VCP 코드 패킷은 이러한 패킷에 의해 기술되는 비-연속 MCCS VCP 제어 코드 파라미터를 규정하는 값을 포함한다. 무효 MCCS VCP 제어 코드가 요청 유효 파라미터 패킷에 의해 규정되면, 동일한 무효 파라미터 값이 요청 코드 필드의 적절한 값을 가지고 이러한 필드에서 규정될 것이다. MCCS 제어 코드가 무효이면, VCP 파라미터 응답 리스트는 제로 길이를 가질 것이다.

응답 코드 필드는 규정된 MCCS VCP 제어에 대한 정보 요청과 관련된 응답의 본성(nature)을 규정하는 2 바이트 정보 또는 값들을 포함한다. 이러한 필드의 값이 0이면, 어떠한 예러도 이러한 데이터 타입에 대해 존재하지 않는 것으로 간주되고, 이 시퀀스의 최종 유효 파라미터 응답 패킷이 전송되며, 이는 가장 높은 응답 시퀀스 번호를 갖는다. 이러한 필드의 값이 1이면, 어떠한 예러도 존재하지 않는 것으로 간주되지만, 보다 높은 시퀀스 번호들을 갖는 다른 유효 응답 패킷들이 전송될 것이다. 이러한 필드의 값이 2이면, 규정된 제어는 클라이언트에서 실행되는 것으로 간주되지 않는다. 이러한 필드의 값이 3이면, 규정된 제어는 비-연속 제어가 아니다(0에서 그 최대 값까지 모든 값들의 유효 세트를 항상 갖는 연속 제어임). 4 내지 65535인 이러한 필드에 대한 값들은 차후 사용을 위해 예약되고 일반적으로 사용되지 않는다.

2 바이트 응답 시퀀스 번호 필드는 클라이언트에 의해 리턴되는 유효 파라미터 응답 패킷들의 시퀀스 번호를 규정한다. 클라이언트는 요청 유효 파라미터 패킷에 응답하여 하나 이상의 유효 파라미터 응답 패킷들을 리턴한다. 클라이언트는 다수

의 유효 파라미터 응답 패킷들에 대해 VCP 파라미터 응답 리스트를 확산시킬 수 있다. 이러한 후자의 경우에, 클라이언트는 시퀀스 번호를 각각의 연속적인 패킷에 할당하고, 시퀀스의 최종 패킷을 제외한 모든 패킷에서 응답 코드를 1로 설정할 것이다. 시퀀스의 최종 유효 파라미터 응답 패킷은 가장 높은 시퀀스 번호를 가질 것이며, 응답 코드는 0의 값을 포함한다.

리스트 필드의 값들의 2 바이트 수는 VCP 파라미터 응답 리스트에 존재하는 16비트 값들의 수를 규정한다. 응답 코드가 0이 아니면, 리스트 파라미터의 값들의 수는 0이다. VCP 파라미터 응답 리스트 필드는 MCCS 제어 코드 필드에 의해 규정되는 비-연속 제어에 대한 유효 값들의 세트를 표시하는 0 내지 32760까지의 2 바이트 값들 리스트를 포함한다. 비-연속 제어 코드들에 대한 정의는 VESA MCCS 규격에서 규정된다. 마지막으로, 이러한 실시예에서, CRC 필드는 패킷 길이를 포함하는 패킷에서 모든 바이트들 중 16 비트 CRC를 포함한다.

알파 커서 이미지

통신 링크 상에서 데이터를 통신하기 위한 MDD 인터페이스 및 관련된 독창적인 프로토콜 및 메커니즘은 서로를 오버래핑하고 가변적인 투명도를 가질 수 있는 다수의 이미지 평면들에 대한 지원을 제공한다. 하드웨어 커서는 가변 X-Y 오프셋을 갖는 오버래핑 이미지를 사용하여 구현될 수 있다. 알파 커서 기능성 및 관련된 프로토콜 지원에 대한 개관이 아래에서 제시된다. 알파 커서 이미지 패킷을 지원하는 능력은 요청 특정 상태 패킷에 응답하여 전송되는 알파 커서 이미지 성능 패킷에서 정의된다.

32. 알파 커서 이미지 성능 패킷

알파 커서 이미지 성능 패킷은 알파 커서 이미지 및 클라이언트에서의 관련 투명성 맵들에 대한 특성들을 정의하는데 사용된다. 일 실시예에서, 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에서 133 파라미터 값을 사용하여 알파 커서 이미지 성능 패킷을 지원하는 능력을 표시한다. 패킷 길이 필드에서 규정되는 패킷 길이는 일 실시예에 있어서 20의 고정 값으로 설정되고, 패킷 길이 필드를 포함하지 않는다.

일 실시예에서, 알파 커서 이미지 성능 패킷의 포맷은 도 75에 제시된다. 도 75에 제시된 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, cClient ID, 알파 커서 식별기, 알파 커서 비트맵 폭, 알파 커서 비트맵 높이, RGB 성능, 단색 성능, 예약 1, Y Cr Cb 성능, 투명성 맵 Res, 성능 비트, 및 CRC 필드를 갖도록 구성된다. cClient ID 필드는 일반적으로 차후 클라이언트 ID 사용을 위해 예약되고, 현재 0으로 설정된다.

알파 커서 식별자 필드(2 바이트)는 특정 알파 커서 평면을 식별하는 값을 포함한다. 클라이언트가 n개의 알파 커서 이미지 평면들을 지원하면, 알파 커서 식별자는 0 내지 n-1의 유효 범위를 갖는다. 일 실시예에서, 값 n은 디스플레이 성능 패킷의 알파 커서 이미지 평면 필드에 의해 규정된다. 클라이언트는 각각의 알파 커서 이미지 평면에 대한 고유한 알파 커서 이미지 성능 패킷을 리턴한다.

2-바이트 알파 커서 비트맵 폭 필드 값은 픽셀들 수로서 표현되는 알파 커서 비트맵 이미지의 폭을 규정하고, 2-바이트 알파 커서 비트맵 높이는 픽셀들의 수로 표현되는 알파 커서 비트맵 이미지의 높이를 규정한다.

RGB 성능 필드는 RGB 포맷으로 디스플레이될 수 있는 해상도의 비트들 수를 규정하기 위해서 2 바이트를 사용한다. 클라이언트가 RGB 포맷을 사용할 수 없으면, 이 값은 0이다. RGB 성능 워드는 3개의 개별 값들로 구성되고, 일 실시예에서, 이러한 값들은 다음과 같이 구현된다: 비트 3 내지 0은 각 픽셀에서 청색 비트들의 최대 수(청색 강도)를 정의한다: 비트 7 내지 4는 각 픽셀에서 녹색 비트들의 최대 수(녹색 강도)를 정의한다: 비트 11 내지 8은 각 픽셀에서 적색 비트들의 최대 수(적색 강도)를 정의한다: 그리고 비트 15 내지 12는 RGB 성능 정보를 표현하는데 있어서 차후 사용을 위해 예약되어, 이들은 일반적으로 현재 0으로 설정된다.

1 바이트 단색 용량 필드는 단색 포맷으로 디스플레이될 수 있는 해상도의 비트들의 수를 특정하기 위해 사용될 수 있다. 클라이언트가 단색 포맷을 사용할 수 없다면, 이러한 값은 0으로 설정된다. 7부터 4까지의 비트들은 나중에 사용하기 위해 예약되며, 그러므로 일반적으로 0으로 설정된다. 3부터 0까지의 비트들은 각각의 픽셀에 존재할 수 있는 그레이스케일의 비트들의 최대 개수를 정의한다. 이러한 네 개의 비트들은 각각의 픽셀이 1부터 15까지의 비트들로 구성되는 것을 특정할 수 있다. 상기 값이 0이면, 단색 포맷은 클라이언트에 의해 지원되지 않는다.

1 바이트 예약 1 필드는 일반적으로 나중에 사용하기 위해 예약된 값을 포함하며, 상기 필드에 있는 모든 비트들은 0으로 설정된다. 이것은 다음 2 바이트 필드들이 16 비트 워드 주소로 할당되도록 하고 4 바이트 필드들이 32 비트 워드 주소로 할당되도록 할 것이다.

2 바이트 Y Cb Cr 기능 필드는 Y Cb Cr 포맷으로 디스플레이될 수 있는 해상도의 비트들의 수를 특정하는 값들 또는 정보를 포함한다. 클라이언트가 Y Cb Cr 포맷을 사용할 수 없다면 상기 값은 0으로 설정된다. 일반적으로, 일 실시예에서, Y Cb Cr 기능 워드는 세 개의 개별적인 값들로 구성된다: 0부터 3까지의 비트들은 Cr 샘플을 특정하는 비트들의 최대 수를 정의하고; 4부터 7까지의 비트들은 Cb 샘플을 특정하는 비트들의 최대 수를 정의하고; 8부터 11까지의 비트들은 Y 샘플을 특정하는 비트들의 최대 수를 정의하고; 12부터 15까지의 비트들은 나중에 Y Cb Cr 기능 정보 또는 값들을 표현하기 위해 예약되며, 현재는 0으로 설정되어 있다.

1 바이트 투명도 맵 해상도 필드는 알파 커서 이미지 투명도 맵의 각각의 픽셀 위치에 있는 비트들의 수(깊이)를 특정하는 값들 또는 정보를 포함한다. 상기 값은 1부터 8까지의 범위에 존재할 수 있다. 상기 값이 0이면, 투명도 맵은 알파 커서 이미지 버퍼(버퍼는 알파 커서 식별 필드에 의해 특정됨)를 위해 지원되지 않는다.

1 바이트 기능 비트 필드는 알파 커서 이미지 버퍼와 관련된 기능들을 특정하는 플래그들의 세트를 포함하는 값들 또는 정보를 제공한다. 일 실시예에서, 플래그들은 아래와 같이 정의된다: 비트 0은 패킹된 포맷으로 존재하는 알파 커서 비디오 스트림 패킷의 픽셀 데이터를 선택하기 위해 사용된다. 비트 1은 알파 커서 투명도 패킷의 투명도 맵 데이터가 패킷 포맷으로 존재하는 것을 나타내기 위해 사용된다. 바이트-할당되고 패킹된 투명도 맵 데이터의 예는 도 76에 도시되어 있다. 비트 2는 알파 커서 이미지 평면이 알파 커서 이미지 오프셋 패킷을 사용하여 이미지 오프셋 기능을 지원하는 것을 나타내기 위해 사용된다. 비트 3은 알파 커서 이미지 평면이 컬러 맵 데이터 포맷을 지원할 수 있다는 것을 나타내기 위해 사용된다. 동일한 컬러 맵 테이블이 알파 커서 이미지 평면들을 위해 사용되며, 메인 이미지 버퍼와 스케일 비디오 스트림들을 위해 사용된다. 컬러 맵은 다른 곳에서 설명되는 컬러 맵 패킷을 사용하여 구성된다.

4부터 7까지의 비트들은 나중에 사용하기 위해 예약되며, 일반적으로 0값 또는 로직 레벨로 설정되어 있다.

33. 알파 커서 투명도 맵 패킷

알파 커서 투명도 맵 패킷은 특정한 알파 커서 이미지 평면에 대한 이미지 투명도 맵의 콘텐츠를 정의한다. 몇몇 애플리케이션들은 하나의 패킷으로 전송될 수 있는 데이터의 양보다 큰 투명도 맵을 요구할 수 있다. 이러한 경우들에서, 다중 알파 커서 투명도 맵 패킷들이 아래에서 설명되는 투명도 맵 X 및 Y 시작 필드들을 사용함으로써 전송될 수 있으며, 각각의 패킷은 투명도 맵의 상이한 서브세트를 가질 수 있다. 이러한 필드들은 비디오 스트림 패킷의 X 시작 및 Y 시작 필드들과 유사한 방식으로 동작한다. 클라이언트는 알파 커서 이미지 기능 패킷의 알파 커서 식별 필드에 의해 특정된 각각의 특정 알파 커서 평면에 대한 알파 커서 이미지 기능 패킷의 투명도 맵 해상도 필드를 사용하는 일 실시예에서 알파 커서 투명도 맵 패킷을 지원하는 능력을 표시한다. 패킷 길이와 클라이언트 ID 필드들은 이전에 논의된 다른 패킷들에 대하여 이전과 같이 동작한다. 일 실시예에서, 패킷 타임 필드에 있는 134의 값은 알파 커서 투명도 맵 패킷으로서 패킷을 식별하기 위해 사용된다.

일 실시예에 대한 알파 커서 투명도 맵의 포맷은 도 76에 도시되어 있다. 도 76에 도시되 바와 같이, 이러한 타입의 패킷은 패킷 길이, 패킷 타입, hClient ID, 알파 커서 식별자, 투명도 맵 X 시작, 투명도 맵 Y 시작, 투명도 맵 해상도, 예약 1, 파라미터 CRC, 투명도 맵 미디어 및 투명도 맵 데이터 CRC 필드들을 포함하도록 구성된다.

2 바이트 알파 커서 식별 필드는 특정 알파 커서 평면을 식별하는 값을 가지고 있다. 클라이언트가 n개의 알파 커서 이미지 평면들을 지원한다면, 알파 커서 식별자는 0부터 n-1까지의 유효한 범위를 가진다.

2 바이트 투명도 맵 X 및 Y 시작 필드들은 각각 절대적인 X 및 Y 좌표들을 특정하며, 여기서 포인트(투명도 맵 X 시작, 투명도 맵 Y 시작)는 아래의 투명도 맵 데이터 필드에 있는 첫 번째 픽셀이다.

투명도 맵 해상도 필드(1 바이트)는 투명도 맵의 해상도와 데이터가 패킹되었는지 여부를 특정하는 값을 포함한다. 상기 필드에 대한 일 실시예에서, 0부터 3까지의 비트들은 모든 투명도 맵 테이블 아이템들에 존재하는 해상도의 비트들의 수를 정의한다. 유효한 값들은 1에서 8까지의 비트들이 되도록 넓이를 특정한다. 0과 9부터 15까지의 값들은 유효하지 않은 것으로 간주된다. 이러한 값은 알파 커서 이미지 기능 패킷의 투명도 맵 해상도 필드에 있는 클라이언트에 의해 리턴된 값과 매칭되어야 한다. 4부터 6까지의 비트들은 나중의 사용을 위해 예약되며, 그러므로 일반적으로 현재 시점에서 로직-0으로 설정되어야 한다. 상기 바이트의 비트 7은 투명도 맵 데이터가 패킹된 형태 또는 바이트-할당된 형태로 있는지 여부를 특정한다. 비트 7이 '1'이면 투명도 맵 데이터는 패킹된 형태이며, '0'이면 데이터는 바이트-할당된 형태이다. 패킹되고 바이트-할당된 투명도 맵 데이터의 예는 다른 곳에서 도시되어 있다. 이러한 비트 값은 알파 커서 이미지 기능 패킷의 기능 비트 필드의 비트 1의 값과 매칭되어야 한다.

1 바이트 예약 1 필드는 나중의 사용을 위해 예약되며, 그러므로 상기 필드의 모든 비트들은 일반적으로 로직-0 레벨과 동일하게 설정된다. 상기 필드의 하나의 목적은 다음의 2 바이트 필드들이 16 비트 워드 주소로 할당되고 4 바이트 필드들이 32 비트 워드 주소로 할당되도록 하는 것이다. 파라미터 CRC 필드는 패킷 길이부터 예약 1 필드까지 모든 바이트들의 16 비트 CRC를 포함한다. 상기 CRC 검사가 실패하면, 전체 패킷이 버려지게 된다.

투명도 맵 데이터 필드에 있어서, 각각의 투명도 맵 위치는 넓이가 1에서 8비트 범위에 있다. 하나의 투명도 맵이 하나의 알파 및 커서 투명도 맵 패킷에 적합하지 않으면, 전체 투명도 맵은 상이한 투명도 맵 데이터와 각각의 픽셀에 있는 투명도 맵 X 및 Y 시작 값들과 함께 다수의 패킷을 전송함으로써 특정될 수 있다.

2 바이트 투명도 맵 데이터 CRC 필드는 오직 투명도 맵 데이터의 16 비트 CRC를 포함한다. 상기 CRC 검사가 실패하면, 투명도 맵 데이터는 여전히 사용될 수는 있으나, CRC 에러 카운트가 증가하게 될 것이다.

34. 알파 커서 이미지 오프셋 패킷

알파 커서 이미지 오프셋 패킷은 메인 디스플레이 이미지의 상부 좌측 코너로부터 커서의 X 및 Y 오프셋을 특정한다. 알파 커서 이미지 오프셋 패킷의 포맷은 도 77에 도시되어 있다. 도 77에 도시된 바와 같이, 일 실시예에서, 알파 커서 이미지 오프셋 패킷은 패킷 길이, 패킷 타입, hClient ID, 알파 커서 X 오프셋, 알파 커서 Y 오프셋 및 CRC 필드들과 함께 구성된다. 일 실시예에서, 클라이언트는 알파 커서 이미지 기능 패킷의 알파 커서 식별자 필드에 의해 특정된 각각의 특정 알파 커서 평면에 대한 알파 커서 이미지 기능 패킷의 기능 비트 필드의 비트 2를 사용하여 알파 커서 이미지 오프셋 패킷을 지원하기 위한 능력을 표시한다. 일 실시예에서, 패킷 길이는 2 바이트 패킷 길이 필드에서 도시된 바와 같이 10으로 고정된다. 일 실시예에서, 135의 패킷 타입은 알파 커서 이미지 오프셋 패킷으로서 패킷을 식별한다.

2 바이트 알파 커서 X 및 Y 오프셋 필드들은 각각 메인 이미지의 좌측 및 상부로부터 커서 이미지의 픽셀들의 가장 좌측 열 및 상부 행의 수평 및 수직 오프셋을 특정하는 값들을 포함한다. hClient ID - 2 바이트는 클라이언트 ID에 대하여 예약된 16 비트의 양의 정수를 포함한다. 상기 필드는 나중의 사용을 위해 예약되며 0으로 설정되어야 한다.

35. 알파 커서 비디오 스트림 패킷

알파 커서 비디오 스트림 패킷은 알파 커서 이미지 평면의 사각형 구역을 업데이트하기 위한 비디오 데이터를 전달한다. 이러한 구역의 크기는 하나의 픽셀만큼 작을 수도 있고 또는 전체 디스플레이만큼 클 수도 있다. 알파 커서 비디오 스트림 패킷의 포맷은 도 78에 도시되어 있다. 도 78에 도시된 바와 같이, 일 실시예에서, 알파 커서 비디오 스트림 패킷은 패킷 길이, 패킷 타입, bClient ID, 비디오 데이터 포맷 속성들, X 좌측 예지, Y 상부 예지, X 우측 예지, Y 하부 예지, X 시작, Y 시작, 픽셀 카운트, 파라미터 CRC 픽셀 데이터 및 픽셀 데이터 CRC 필드들과 함께 구성된다. 일 실시예에서, 클라이언트는 알파 커서 이미지 기능 패킷의 알파 커서 식별자 필드에 의해 특정된 각각의 특정 알파 커서 평면에 대한 알파 커서 이미지 기능 패킷을 사용함으로써 알파 커서 비디오 스트림 패킷 및 자신과 관련된 파라미터들을 지원하기 위해 능력을 표시하며, 패킷 타입 필드에 있는 17의 값은 알파 커서 비디오 스트림 패킷으로서 패킷을 표시하거나 또는 식별한다. hClient ID 필드(2 바이트)는 클라이언트 ID로서 나중에 사용하기 위해 예약되고, 일반적으로 0으로 설정되며, 이는 기술적으로 용이하게 이해될 것이다.

2 바이트 비디오 데이터 포맷 서술자(descriptor) 필드는 현재 패킷의 현재 스트림의 픽셀 데이터에 있는 각각의 픽셀의 포맷을 특정하는 정보 또는 값을 포함한다. 픽셀 데이터 포맷은 알파 커서 이미지 기능 패킷에서 정의된 바와 같이 알파 커서 이미지 평면에 대한 유효한 포맷들 중 적어도 하나와 일치해야 한다. 비디오 데이터 포맷 서술자 필드는 오직 현재의 패킷에 대한 픽셀 포맷을 정의하는 값을 포함하며 일정한 포맷이 특정 비디오 스트림의 존재 기간 동안 계속해서 사용되어야 한다는 것을 의미하지 않는다. 위의 도 11은 비디오 데이터 포맷 서술자가 어떻게 코딩되는지를 나타낸다. 포맷은 아래와 같다:

일 실시예에서, 비트들 [15:13]이 '000'이면, 비디오 데이터는 단색 픽셀들의 어레이로 구성되며, 픽셀 당 비트들의 수는 비디오 데이터 포맷 서술자 워드의 비트들 3 내지 0에 의해 정의된다. 그 후에 비트들 11 내지 4는 0으로 설정된다. 비트들 [15:13]이 '001'이면, 비디오 데이터는 컬러 픽셀들의 어레이로 구성되며, 각각의 픽셀은 컬러 맵(팔레트)을 통해 컬러를 특정한다. 비디오 데이터 포맷 서술자 워드의 비트들 5 내지 0은 픽셀 당 비트들의 수를 정의하며, 비트들 11 내지 6은 0으로 설정된다. 비트들 [15:13]이 '010'이면, 비디오 데이터는 원시(raw) RGB 포맷의 컬러 픽셀들의 어레이로 구성되며, 적

색에 대한 픽셀 당 비트들의 수는 비트들 11 내지 8을 통해 정의되고, 녹색에 대한 픽셀 당 비트들의 수는 비트들 7 내지 4를 통해 정의되고, 청색에 대한 픽셀 당 비트들의 수는 비트들 3 내지 0을 통해 정의된다. 각각의 픽셀의 전체 비트들의 수는 적색, 녹색 및 청색을 위해 사용된 비트들의 수의 합이다.

비트들 [15:13]이 '011'이면, 비디오 데이터는 휘도(luminance) 및 색차 정보를 가지는 4:2:2 Y Cb Cr 포맷의 비디오 데이터의 어레이로 구성된다. 휘도(Y)에 대한 픽셀 당 비트들의 수는 비트들 11 내지 8을 통해 정의되고, Cb 성분에 대한 픽셀 당 비트들의 수는 비트들 7 내지 4를 통해 정의되고, Cr 성분에 대한 픽셀 당 비트들의 수는 비트들 3 내지 0을 통해 정의된다. Cb 및 Cr 성분들은 Y의 레이트의 절반인 레이트로 전송된다. 상기 패킷의 픽셀 데이터 부분에 있는 비디오 샘플들은 아래와 같이 구성될 것이다: Cbn, Yn, Crn, Yn+1, Cbn+2, Yn+2, Crn+2, Yn+3, . . . 여기서, Cbn 및 Crn은 Yn 및 Yn+1과 관련되고, Cbn+2 및 Crn+2는 Yn+2 및 Yn+3과 관련되며, 다음 샘플들도 같은 방식으로 관련된다. Yn, Yn+1, Yn+2 및 Yn+3은 좌측에서 우측으로 하나의 행에 있는 네 개의 연속적인 픽셀들의 휘도 값들이다. 컬러 성분들의 순서는 마이크로소프트의 UYVY FOURCC 포맷과 동일하다. 비디오 스트림 패킷에 의해 참조된 윈도우에 있는 행(X 우측 에지 - X 좌측 에지 + 1)에 홀수개의 픽셀들이 존재하면, 각 행의 마지막 픽셀에 대응하는 Cb 값 다음에는 다음 행의 첫 번째 픽셀의 Y 값이 이어질 것이다.

Y Cb Cr 포맷을 사용하는 윈도우들은 짝수개의 픽셀들로 이루어진 넓이를 가지는 것이 바람직하다. 패킷의 픽셀 데이터는 짝수개의 픽셀들을 포함할 것이다. 픽셀 데이터의 마지막 픽셀이 비디오 스트림 패킷 헤더에서 특정된 윈도우에 있는 행의 마지막 픽셀에 대응하는 경우에, 즉, 픽셀 데이터의 마지막 픽셀의 X 위치가 X 우측 에지에 동일할 때, 픽셀 데이터는 홀수 또는 짝수개의 픽셀들을 포함할 수 있다.

모든 5개의 포맷에 있어서, (도면들에서 "P"로 지정된) 비트 12는 픽셀 데이터 샘플들이 패킹되었는지 여부를 특정한다. 비트 12의 값이 '0'이면, 각각의 픽셀과 픽셀 데이터 필드에 있는 각각의 픽셀의 컬러는 MDDI 인터페이스 바이트 경계를 통해 바이트-할당된다. 비트 12의 값이 '1'이면, 각각의 픽셀과 픽셀 데이터에 있는 각각의 픽셀의 컬러는 이전의 픽셀 또는 사용되지 않은 비트들이 없는 픽셀의 컬러에 대하여 패킹된다.

일 실시예에서, 픽셀 데이터 속성 필드(2 바이트)는 일련의 비트 값들을 가지고 있으며, 이들은 다음과 같이 해석된다. 비트들 1 및 0은 디스플레이 픽셀 데이터가 어떻게 라우팅되는지를 선택한다. '11'의 비트 값에 대하여, 데이터는 양쪽 눈에 대하여 디스플레이되고, '10'의 비트 값에 대하여, 데이터는 오직 왼쪽 눈에 대하여만 라우팅되며, '01'의 비트 값에 대하여, 데이터는 오직 우측 눈에 대하여만 라우팅된다.

픽셀 데이터 속성 필드의 비트 2는 픽셀 데이터가 비월 주사 포맷으로 표현되는지 여부를 표시하며, 상기 비트가 '0' 값을 가지면 픽셀 데이터는 표준 진행 포맷으로 구성되고 행 번호(픽셀 Y 좌표)는 하나의 행에서 다음 행으로 이동할 때 1만큼 증가한다. 상기 비트가 '1' 값을 가지면, 픽셀 데이터는 비월 주사 포맷으로 구성되며, 행의 수는 하나의 행에서 다음 행으로 이동할 때 2만큼 증가된다. 비트 3은 픽셀 데이터가 대안적인 픽셀 포맷으로 구성된다는 것을 표시한다. 이것은 비트 2에 의해 인에이블되는 표준 비월 주사 모드와 유사하나, 비월 주사가 수평적이 아니라 수직적이다. 비트 3이 '0'이면, 픽셀 데이터는 표준 진행 포맷으로 구성되며, 열 번호(픽셀 X 좌표)는 각각의 연속적인 픽셀이 수신되면 1만큼 증가된다. 비트 3이 '1'이면, 픽셀 데이터는 대안적인 픽셀 포맷으로 구성되며, 열 번호는 각각의 픽셀이 수신되면 2만큼 증가된다.

픽셀 데이터 속성 필드의 비트 4는 픽셀 데이터가 디스플레이 또는 카메라와 관련되는지 여부를 표시하며, 여기서 데이터는 무선 전화기 또는 유사한 장치 또는 휴대용 컴퓨터 또는 위에서 논의된 다른 장치들에 대한 내부 디스플레이로 전달되거나 또는 내부 디스플레이로부터 전달되거나, 또는 데이터는 상기 장치에 설치되어 있거나 상기 장치에 직접 연결된 카메라로 전달되거나 또는 카메라로부터 전달된다. 비트 4가 '0'이면, 픽셀 데이터는 디스플레이 프레임 버퍼로 전달되거나 또는 디스플레이 프레임 버퍼로부터 전달된다. 비트 4가 '1'이면, 픽셀 데이터는 몇몇 타입의 카메라 또는 비디오 장치로 전달되거나 또는 이들 장치로부터 전달되며, 이러한 장치들은 기술적으로 잘 알려진 장치들이다.

픽셀 데이터 속성 필드의 비트 5는 나중의 사용을 위해 또는 MDD 인터페이스의 애플리케이션들을 위해 예약되며, 그러므로 일반적으로 0값 또는 '0'으로 설정된다.

픽셀 데이터 속성 필드의 비트 7 및 6은 픽셀 데이터가 기록될 프레임 버퍼를 특정하는 디스플레이 업데이트 비트들이다. 보다 상세한 효과들은 다른 곳에서 논의된다. '01'의 비트 값들에 대하여, 픽셀 데이터는 오프라인 이미지 버퍼에 기록된다. '00'의 비트 값들에 대하여, 픽셀 데이터는 디스플레이를 재생하기 위해 사용되는 이미지 버퍼에 기록된다. '11'의 비트 값들에 대하여, 픽셀 데이터는 모든 이미지 버퍼들에 기록된다. '10'의 비트 값들 또는 조합은 유효하지 않은 값 또는 지정

으로서 취급되며, 픽셀 데이터는 무시되고 어떠한 이미지 버퍼들에도 기록되지 않는다. 이러한 값은 인터페이스에 대한 나중의 애플리케이션들을 위해 사용될 수 있다. 비트 8 내지 15는 나중의 사용의 위해 예약되며, 그러므로 일반적으로 0으로 설정된다.

일 실시예에서, 2 바이트 X 시작 및 Y 시작 필드들은 픽셀 데이터 필드의 첫 번째 픽셀에 대한 포인트(X 시작, Y 시작)의 절대 X 및 Y 좌표를 특정한다. 2 바이트 X 좌측 에지 및 Y 상부 에지 필드들은 픽셀 데이터 필드에 의해 채워진 알파 커서 이미지 윈도우의 좌측 에지에 대한 X 좌표와 상부 에지에 대한 Y 좌표를 특정하며, X 우측 에지 및 Y 하부 에지 필드들은 업데이트되는 알파 커서 이미지 윈도우의 우측 에지에 대한 X 좌표와 하부 에지에 대한 Y 좌표를 특정한다.

픽셀 카운트 필드(2 바이트)는 픽셀 데이터 필드에 있는 픽셀들의 수를 특정한다. 2 바이트 파라미터 CRC 필드는 패킷 길이부터 픽셀 카운트까지 모든 바이트들의 CRC를 포함한다. 이러한 CRC 검사가 실패하면, 전체 패킷은 버려지게 된다.

픽셀 데이터 필드는 디스플레이되며, 비디오 데이터 포맷 서술자 필드에 의해 설명된 방식으로 포맷되는 원시 비디오 정보를 포함한다. 상기 데이터는 다른 곳에서 논의된 바와 같이 한 번에 하나의 "행"이 전송된다. 픽셀 데이터 CRC 필드(2 바이트)는 오직 픽셀 데이터의 16 비트 CRC를 포함한다. 이러한 값의 CRC 검증이 실패하면, 픽셀 데이터는 여전히 사용되지만, CRC 에러 카운트는 증가하게 된다.

스케일 비디오 스트림 이미지들

MDD 인터페이스 또는 프로토콜 메커니즘 또는 방법은 호스트가 원시 이미지보다 더 크게 또는 더 작게 스케일링된 이미지를 클라이언트로 전송하도록 허용하는 스케일 비디오 스트림 이미지들을 지원하며, 스케일링된 이미지는 메인 이미지 버퍼로 복사된다. 스케일 비디오 스트림 기능 및 관련된 프로토콜 지원에 대한 개관은 다른 곳에서 설명된다. 스케일 비디오 스트림들을 지원하기 위한 기능은, 요청 특정 상태 패킷(Request Specific Status Packet)에 응답하여 전송되는, 스케일 비디오 스트림 기능 패킷에 의해 또는 스케일 비디오 스트림 기능 패킷 내에 정의된다.

36. 스케일 비디오 스트림 기능 패킷

스케일 비디오 스트림 기능 패킷은 클라이언트에 있거나 클라이언트에 의해 사용되는 스케일 비디오 스트림 소스 이미지의 특성들을 정의한다. 스케일 비디오 스트림 기능 패킷의 포맷은 일반적으로 도 79에 도시되어 있다. 도 79에 도시된 바와 같이, 일 실시예에서, 스케일 비디오 스트림 기능 패킷은 패킷 길이, 패킷 타입, cClient ID, 스트림들의 최대 개수, 소스 최대 X 크기, 소스 최대 Y 크기, RGB 기능, 단색 용량, 예약 1, Y Cr Cb 기능, 예약 2 및 CRC 필드들을 포함하도록 구성된다. 일 실시예에서, 패킷 길이는 길이 필드에서 도시된 바와 같이 고정된 20 바이트가 되도록 선택되며, 길이 필드는, 클라이언트 ID를 위한 사용을 위해 예약되거나 그렇지 않으면 0으로 설정되는, 2 바이트 cClient ID 필드와 CRC 필드를 포함한다. 일 실시예에서, 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에 있는 143의 파라미터 값을 이용하여 스케일 비디오 스트림 기능 패킷을 지원하기 위한 능력을 표시한다.

2 바이트 스트림들의 최대 개수 필드는 한 번에 할당될 수 있는 동시에 스케일 비디오 스트림들의 최대 개수를 식별하기 위한 값을 포함한다. 일 실시예에서, 스케일 비디오 스트림들의 최대 개수가 이미 할당된 경우에, 클라이언트는 스케일 비디오 스트림을 할당하기 위한 요청을 거부해야 한다. 최대 개수보다는 적은 개수의 스케일 비디오 스트림이 할당된 경우에도, 클라이언트는 자신의 다른 리소스 제한들에 기반하여 할당 요청을 거부할 수 있다.

소스 최대 X 크기 및 소스 최대 Y 크기 필드들(2 바이트)은 각각 다수의 픽셀로 표현된 스케일 비디오 스트림 소스 이미지의 최대 폭 및 높이에 대한 값들을 특정한다.

RGB 기능 필드는 RGB 포맷으로 디스플레이될 수 있는 해상도의 비트들의 수를 특정하기 위한 값들을 사용한다. 스케일 비디오 스트림이 RGB 포맷을 사용할 수 없다면, 상기 값은 0으로 설정된다. RGB 기능 워드는 세 개의 개별적인 양의 값들로 구성된다: 비트들 3 내지 0은 각 픽셀의 청색에 대한 비트들의 최대 개수(청색 밀도)를 정의하고, 비트들 7 내지 4는 각 픽셀의 녹색에 대한 비트들의 최대 개수(녹색 밀도)를 정의하고, 비트들 11 내지 8은 각 픽셀의 적색에 대한 비트들의 최대 개수(적색 밀도)를 정의하며, 비트들 15 내지 12는 기능 정의를 위한 나중의 사용을 위해 예약되며 일반적으로 0으로 설정된다.

1 바이트 단색 용량 필드는 단색 포맷에서 디스플레이될 수 있는 해상도의 비트들의 수를 특정하는 값을 포함한다. 스케일 비디오 스트림이 단색 포맷을 사용할 수 없다면, 상기 값은 0으로 설정된다. 비트들 7 내지 4는 나중의 사용을 위해 예약되며, 따라서 시간의 경과하면 변경될 수 있음에도 불구하고, 현재의 애플리케이션들을 위해 '0'으로 설정되어야 하며, 이는

당업자들에 의해 이해될 것이다. 비트들 3 내지 0은 각 픽셀에 존재할 수 있는 그레이스케일의 비트들의 최대 수를 정의한다. 이러한 네 개의 비트들은 각 픽셀이 1에서 15까지의 비트들로 구성되는 것을 특정하기 위해 사용될 수 있다. 상기 값이 0이면, 단색 포맷은 스케일 비디오 스트림에 의해 지원되지 않는다.

예약 1 필드(여기서, 1 바이트)는 스케일 비디오 스트림 패킷 정보 또는 데이터와 관련된 값들을 제공하기 위한 나중의 사용을 위해 예약된다. 그러므로 현재, 상기 필드의 모든 비트들은 로직 '0'으로 설정된다. 상기 필드의 하나의 목적은 모든 다음 2 바이트 필드들이 16 비트 워드 주소로 할당되도록 하고 4 바이트 필드들이 32 비트 워드 주소로 할당되도록 하는 것이다.

2 바이트 Y Cb Cr 기능 필드는 Y Cb Cr 포맷으로 디스플레이될 수 있는 해상도의 비트들의 수를 특정하는 값들을 포함한다. 스케일 비디오 스트림이 Y Cb Cr 포맷을 사용할 수 없으면, 상기 값은 0으로 설정된다. Y Cb Cr 기능 워드는 세 개의 개별적인 양의 값들로 구성된다: 비트들 3 내지 0은 Cr 샘플을 특정하는 비트들의 최대 개수를 정의하고, 비트들 7 내지 4는 Cb 샘플을 특정하는 비트들의 최대 개수를 정의하고, 비트들 11 내지 8은 Y 샘플을 특정하는 비트들의 최대 개수를 정의하며, 비트들 15 내지 12는 나중의 사용을 위해 예약되고 일반적으로 0으로 설정된다.

1 바이트 기능 비트 필드는 스케일 비디오 스트림과 관련된 기능들을 특정하는 플래그들의 세트를 포함한다. 플래그들은 다음과 같이 정의된다: 비트 0은 스케일 비디오 스트림 패킷에 있는 픽셀 데이터가 패킹된 포맷으로 구성될 수 있도록 커버한다. 패킹되고 바이트-할당된 픽셀 데이터의 예는 도 12에서 도시되어 있다. 비트 1은 나중의 사용을 위해 예약되며 0으로 설정될 것이다; 비트 2는 나중의 사용을 위해 예약되며 0으로 설정될 것이다; 비트 3은 컬러 맵 데이터 포맷에서 특정될 수 있는 스케일 비디오 스트림들을 커버한다. 동일한 컬러 맵 테이블은 메인 이미지 버퍼와 알파 커서 이미지 평면들뿐만 아니라 스케일 비디오 스트림들에 대하여도 사용된다. 컬러 맵은 다른 곳에서 설명된 컬러 맵 패킷을 사용하여 구성되어 있다; 그리고 비트들 7 내지 4는 나중의 사용을 위해 예약되며 일반적으로 0으로 설정된다.

예약 2 필드(여기서, 1 바이트)는 스케일 비디오 스트림 패킷 정보 또는 데이터와 관련된 값들을 제공하기 위한 나중의 사용을 위해 예약된다. 그러므로 현재 상기 필드의 모든 비트들은 로직 '0'으로 설정되어 있다. 상기 필드의 하나의 목적은 모든 다음 2 바이트 필드들이 16 비트 워드 주소로 할당되도록 하고 4 바이트 필드들이 32 비트 워드 주소로 할당되도록 하는 것이다.

37. 스케일 비디오 스트림 셋업 패킷

스케일 비디오 스트림 셋업 패킷은 스케일 비디오 스트림의 파라미터들을 정의하기 위해 사용되며 클라이언트는 이미지의 버퍼링과 스케일링을 위한 내부 저장 매체에 할당하기 위해 상기 정보를 사용한다. 스트림은 0값을 가지는 X 이미지 크기 및 Y 이미지 크기 필드들과 함께 상기 패킷을 전송함으로써 할당 해제(de-allocate)될 수 있다. 할당 해제되었던 스케일 비디오 스트림들은 동일한 또는 다른 스트림 파라미터들을 통해 추후에 재할당될 수 있다. 일 실시예에서, 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에 있는 143의 파라미터 값을 이용함으로써, 그리고 스케일 비디오 스트림 기능 패킷의 스트림들의 최대 개수 필드에 있는 0이 아닌 값을 이용함으로써 스케일 비디오 스트림 셋업 패킷을 지원하기 위한 기능을 표시한다.

스케일 비디오 스트림 셋업 패킷의 포맷은 일반적으로 도 80에 도시되어 있다. 도 80에 도시된 바와 같이, 일 실시예에서, 스케일 비디오 스트림 셋업 패킷은 패킷 길이, 패킷 타입, hClient, 스트림 ID, 시각 데이터 포맷 서술자, 픽셀 데이터 속성들, X 좌측 예지, Y 상부 예지, X 우측 예지, Y 하부 예지, X 이미지 크기, Y 이미지 크기 및 CRC 필드들을 포함하도록 구성된다.

2 바이트 패킷 길이 필드는 패킷 길이 필드를 포함하지 않는 패킷의 전체 바이트 개수를 특정한다. 일 실시예에서, 상기 패킷 길이는 24로 고정되어 있다. 2 바이트 패킷 타입 필드는 스케일 비디오 스트림 셋업 패킷으로서 패킷을 식별하기 위해 136의 값을 사용한다. 2 바이트 hClient ID 필드는 클라이언트 ID로서 나중에 사용하기 위해 예약되며, 일반적으로 나중의 사용 시점 또는 프로토콜 사용자가 어떤 ID 값들이 사용될 것인지를 결정할 때까지 모두 0값으로 설정된다.

스트림 ID 필드는 스트림 ID에 대한 유일한 식별자를 특정하기 위해 2 바이트를 사용한다. 상기 값은 호스트에 의해 할당되며 0부터 디스플레이 기능 패킷에서 특정된 최대 스트림 ID 값까지의 값을 가질 수 있다. 호스트는 각각의 액티브 스트림이 유일한 값을 할당받고 더 이상 액티브 상태에 있지 않은 스트림들이 할당 해제되거나 재할당되는 것을 보장하기 위해 주의 깊게 스트림 ID 값들의 사용을 관리해야 한다.

일 실시예에서, 비디오 데이터 포맷 서술자 필드는 현재 패킷의 현재 스트림의 픽셀 데이터에 있는 각각의 픽셀의 포맷을 특정하기 위해 2 바이트를 사용한다. 픽셀 데이터 포맷은 알파 커서 이미지 기능 패킷에서 정의된 것처럼 알파 커서 이미지 평면에 대한 유효한 포맷들 중 적어도 하나에 적합해야 한다. 비디오 데이터 포맷 서술자는 오직 현재 패킷에 대한 픽셀 포맷을 정의하며, 일정한 포맷이 특정 비디오 스트림의 존재 기간 동안 계속해서 사용되어야 한다는 것을 의미하지 않는다. 도 11은 비디오 데이터 포맷 서술자가 어떻게 코딩되는지에 대한 일 실시예를 나타내며, 위에서 다른 패킷들에 대하여 논의된 바와 같다.

2 바이트 픽셀 데이터 속성 필드는 다음과 같이 해석되는 값들을 가지고 있다:

비트들 1 및 0은 픽셀 데이터가 라우팅되어야 하는 디스플레이를 선택한다.

비트들 [1:0] = 11 or 00 - 데이터는 양쪽 눈에 디스플레이된다.

비트들 [1:0] = 10 - 데이터는 왼쪽 눈에만 디스플레이된다.

비트들 [1:0] = 01 - 데이터는 우측 눈에만 디스플레이된다.

비트 2는 픽셀 데이터가 비월 주사 포맷으로 구성되어 있는지 여부를 표시한다. 비트 2가 0이면, 픽셀 데이터는 표준 진행 포맷으로 구성된다. 행 번호(픽셀 Y 좌표)는 하나의 행에서 다음 행으로 이동할 때 1만큼 증가될 것이다. 비트 2가 1이면, 픽셀 데이터는 비월 주사 포맷으로 구성된다. 행 번호(픽셀 Y 좌표)는 하나의 행에서 다음 행으로 이동할 때 2만큼 증가될 것이다.

비트 3은 픽셀 데이터가 대안적인 픽셀 포맷으로 구성되어 있는지 여부를 표시한다. 이것은 비트 2에 의해 인에이블되는 표준 비월 주사 모드와 유사하나, 비월 주사가 수평적인 대신에 수직적이다. 비트 3이 0이면, 픽셀 데이터는 표준 진행 포맷으로 구성된다. 열 번호(픽셀 X 좌표)는 연속적인 픽셀이 수신될 때마다 1만큼 증가될 것이다. 비트 3이 1이면, 픽셀 데이터는 대안적인 픽셀 포맷으로 구성된다. 열 번호(픽셀 X 좌표)는 각 픽셀이 수신되면 2만큼 증가될 것이다.

비트 4는 픽셀 데이터가 디스플레이 또는 카메라와 관련되어 있는지 여부를 표시한다. 비트 4가 0이면, 픽셀 데이터는 디스플레이 프레임 버퍼로부터 전달된 것이거나 디스플레이 프레임 버퍼로 전달된 것이다. 비트 4가 1이면, 픽셀 데이터는 카메라로부터 전달된 것이거나 카메라로 전달된 것이다. 비트 5는 나중의 사용을 위해 예약되며, 그러므로 일반적으로 0으로 설정된다.

비트들 7 및 6은 픽셀 데이터가 기록되는 프레임 버퍼를 특정하는 디스플레이 업데이트 비트들이다. 프레임 업데이트 비트들의 효과는 다른 곳에서 보다 상세하게 설명되어 있다. 비트들 [7:6]이 '01'이면, 픽셀 데이터는 오프라인 이미지 버퍼에 기록된다. 비트들 [7:6]이 '00'이면, 픽셀 데이터는 디스플레이를 재생하기 위해 사용되는 이미지 버퍼에 기록된다. 비트들 [7:6]이 '11'이면, 픽셀 데이터는 모든 이미지 버퍼들에 기록된다. 비트들 [7:6]이 '10'이면, 이것은 유효하지 않은 값으로 처리된다. 이러한 비트들은 현재 나중의 사용을 위해 예약되어 있다. 이러한 상황에서, 픽셀 데이터는 무시될 것이며, 어떤 이미지 버퍼들에서 기록되지 않을 것이다.

비트들 8 내지 15는 나중의 사용을 위해 예약되며 일반적으로 논리 0 레벨 또는 값으로 설정될 것이다.

38. 스케일 비디오 스트림 확인 응답 패킷

스케일 비디오 스트림 확인 응답 패킷은 클라이언트가 스케일 비디오 스트림 셋업 패킷을 수신한 것을 승인하도록 허용한다. 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에 있는 143의 파라미터 값을 통해 그리고 스케일 비디오 스트림 기능 패킷의 스트림들의 최대 개수 필드에 있는 0이 아닌 값을 통해 스케일 비디오 스트림 확인 응답 패킷을 지원하기 위한 기능을 표시한다.

스케일 비디오 스트림 확인 응답 패킷의 포맷은 도 81에 도시되어 있다. 도 81에 도시된 바와 같이, 일 실시예에서, 스케일 비디오 스트림 확인 응답 패킷은 패킷 길이, 패킷 타입, cClient, 스트림 ID, ACK 코드 및 CRC 필드들을 포함하도록 구성된다. 2 바이트 패킷 길이 필드는 상기 패킷 타입에 대한 10의 값과 함께, 패킷 길이 필드를 제외하고, 전체 바이트들의 개수를 특정하기 위해 사용되며, 137의 패킷 타입은 스케일 비디오 스트림 확인 응답 패킷으로서 패킷을 식별한다.

2 바이트 cClient ID 필드는 클라이언트 ID를 위한 나중의 사용을 위해 예약되며, 일반적으로 0으로 설정된다. 2 바이트 스트림 ID 필드는 스트림 ID에 대한 유일한 식별자를 특정한다. 이것은 스케일 비디오 스트림 셋업 패킷에 의해 할당된 값과 동일하다.

2 바이트 ACK 코드 필드는 특정 스케일 비디오 스트림을 업데이트하기 위한 시도의 결과를 설명하는 코드를 포함하는 값들을 제공한다. 일 실시예에서, 코드들은 다음과 같이 정의된다:

- 0 - 스트림 할당 시도는 성공적이었다.
- 1 - 스트림 할당 해제 시도는 성공적이었다.
- 2 - 이미 할당된 스트림 ID를 할당하기 위한 유효하지 않은 시도이다.
- 3 - 이미 할당 해제된 스트림 ID를 할당 해제하기 위한 유효하지 않은 시도이다.
- 4 - 클라이언트는 스케일 비디오 스트림들을 지원하지 않는다.
- 5 - 스트림 파라미터들은 클라이언트의 기능과 일치하지 않는다.
- 6 - 스트림 ID 값이 클라이언트에 의해 허용되는 최대값보다 크다.
- 7 - 특정 스트림을 할당하기 위해 클라이언트에서 이용가능한 리소스들이 불충분하다.

2 바이트 CRC 필드는 패킷 길이를 포함하는 패킷의 모든 바이트들의 CRC를 포함한다.

39. 스케일 비디오 스트림 패킷

스케일 비디오 스트림 패킷은 특정 스케일 비디오 스트림과 관련된 픽셀 데이터를 전송하기 위해 사용된다. 상기 패킷에 의해 영역 기준의 크기는 스케일 비디오 스트림 셋업 패킷에 의해 정의된다. 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에 있는 143의 파라미터 값을 통해 그리고 스케일 비디오 스트림 확인 응답 패킷의 ACK 코드 필드에 있는 성공적인 스케일 비디오 스트림 할당 응답을 이용하여 스케일 비디오 스트림 패킷을 지원하기 위한 자신의 기능을 표시한다.

스케일 비디오 스트림 패킷의 일 실시예에 대한 포맷은 일반적으로 도 82에 도시되어 있다. 도 82에 도시된 바와 같이, 스케일 비디오 스트림 패킷은 패킷 길이, 패킷 타입, hClient, 스트림 ID, 파라미터 CRC, 픽셀 카운트, 픽셀 데이터 및 픽셀 데이터 CRC 필드들을 포함하여 구성된다. 2 바이트 패킷 타입 필드는 스케일 비디오 스트림 패킷으로서 패킷을 식별하기 위해 18의 값을 사용한다. hClient ID 필드는 클라이언트 ID를 위해 예약되며, 일반적으로 0으로 설정된다. 이전과 같이, 2 바이트 스트림 ID 필드는 스트림 ID에 대한 유일한 식별자를 특정한다. 상기 값은 스케일 비디오 스트림 셋업 패킷의 호스트에 의해 특정되며 스케일 비디오 스트림 확인 응답 패킷에서 확인된다.

2 바이트 픽셀 카운트 필드는 아래의 픽셀 데이터 필드에 있는 픽셀들의 수를 특정한다. 2 바이트 파라미터 CRC 필드는 패킷 길이부터 픽셀 카운트까지의 모든 바이트들의 CRC를 가진다. CRC 검사가 실패하면, 전체 패킷은 버려지게 될 것이다. 2 바이트 픽셀 데이터 필드는 스케일링되고 디스플레이될 원시 비디오 정보를 포함한다. 데이터는 비디오 데이터 포맷 서술자 필드에 의해 제시된 방식으로 포맷된다. 상기 데이터는 이전에 정의된 바와 같이 한 번에 하나의 행이 전송된다.

2 바이트 픽셀 데이터 CRC 필드는 오직 픽셀 데이터의 CRC만을 포함한다. CRC 검사가 실패하더라도, 픽셀 데이터는 여전히 사용될 것이나 CRC 에러 카운트는 증가하게 될 것이다.

40. 요청 특정 상태 패킷

요청 특정 상태 패킷은 클라이언트가 상기 패킷에서 특정된 바와 같이 호스트로 기능 또는 상태 패킷을 다시 전송하도록 호스트가 요청하기 위한 수단, 메커니즘 또는 방법을 제공한다. 클라이언트는 다음 역방향 링크 캡슐화 패킷에 있는 특정 타입의 패킷을 리턴시킨다. 클라이언트가 요청 특정 상태 패킷에 응답하기 위한 기능을 갖고 있으면, 클라이언트는 클라이

엔트 기능 패킷의 클라이언트 특성 기능 필드에 있는 비트 17을 설정할 것이다. 클라이언트가 응답할 수 있는 모든 종류의 상태 패킷을 결정하기 위해 호스트가 사용하는 편리한 방법은 다른 곳에 기재된 유효 상태 응답 리스트 패킷을 사용하는 것이다. 클라이언트는 클라이언트 기능 패킷의 클라이언트 특성 기능 필드의 비트 21을 이용하여 요청 특정 상태 패킷을 지원하기 위한 자신의 기능을 표시할 수 있다.

요청 특정 상태 패킷의 일 실시예에 대한 포맷은 일반적으로 도 83에 도시되어 있다. 도 83에 도시된 바와 같이, 요청 특정 상태 패킷은 패킷 길이, 패킷 타입, hClient ID, 상태 패킷 ID 및 CRC 필드들을 포함하여 구성된다. 패킷 길이 필드는 패킷 길이 필드를 포함하지 않고 패킷에 있는 전체 바이트들의 수를 특정하며, 일반적으로 상기 패킷 타입에 대하여 10의 값으로 고정되어 있다. 138의 패킷 타입은 요청 특정 상태 패킷으로서 패킷을 식별한다. hClient ID 필드(2 바이트)는 클라이언트 ID를 위한 나중의 사용을 위해 예약되며, 현재 0으로 설정되는 한편, 2 바이트 상태 패킷 ID 필드는 클라이언트가 호스트로 전송할 기능 또는 상태 패킷의 타입을 특정한다. 통상적인 패킷 타입은 다음과 같다:

66 - 클라이언트 기능 패킷은 클라이언트에 의해 전송될 것이다.

133 - 알파 커서 이미지 기능 패킷은 클라이언트에 의해 전송될 것이다.

139 - 클라이언트가 전송할 수 있는 기능 및 상태 패킷들의 정확한 타입들을 식별하는 유효 상태 응답 리스트 패킷이 전송될 것이다.

140 - 패킷 처리 지연 파라미터 패킷이 클라이언트에 의해 전송될 것이다.

141 - 개인용 디스플레이 기능 패킷이 클라이언트에 의해 전송될 것이다.

142 - 디스플레이 에러 보고 패킷이 클라이언트에 의해 전송될 것이다.

143 - 스케일 비디오 스트림 기능 패킷이 클라이언트에 의해 전송될 것이다.

144 - 디스플레이 식별 패킷이 클라이언트에 의해 전송될 것이다.

패킷 타입 56 내지 63은 제조자 별로 특정된 기능 및 상태 식별자들을 위해 사용될 수 있다.

CRC 필드들은 다시 패킷 길이를 포함하는 패킷에 있는 모든 바이트들의 CRC를 포함한다.

41. 유효 상태 응답 리스트 패킷

유효 상태 응답 리스트 패킷은 클라이언트가 응답하기 위한 기능을 가지는 상태 및 기능 패킷들의 리스트를 갖는 구조, 수단 또는 방법을 호스트에 제공한다. 클라이언트는 클라이언트 기능 패킷의 클라이언트 특성 기능 필드의 비트 21을 이용하여 유효 상태 응답 리스트 패킷을 지원하기 위한 기능을 표시할 수 있다.

유효 상태 응답 리스트 패킷의 일 실시예에 대한 포맷은 일반적으로 도 84에 도시되어 있다. 도 84에 도시된 바와 같이, 유효 상태 응답 리스트 패킷은 패킷 길이, 패킷 타입, cClient ID, 리스트에 있는 값들 개수, 유효 파라미터 응답 리스트 및 CRC 필드들을 포함하여 구성된다. 이러한 타입의 패킷에 대한 패킷 길이는 일반적으로 10의 값으로 고정되어 있으며, 139의 타입 값은 유효 상태 응답 패킷으로서 패킷을 식별한다. cClient ID 필드는 클라이언트 ID로서 나중에 사용하기 위해 예약되며 일반적으로 0으로 설정된다. 2 바이트의 리스트에 있는 값들 개수 필드는 다음 유효 파라미터 응답 리스트에 있는 아이템들의 수를 특정한다.

유효 파라미터 응답 리스트는 클라이언트가 호스트로 전송할 수 있는 기능 또는 상태 패킷들의 타입들을 특정하는 2 바이트의 파라미터 리스트를 포함한다. 클라이언트가 자신이 (클라이언트 기능 패킷에 있는 클라이언트 특성 기능 필드의 비트 21을 이용하여) 요청 특정 상태 패킷에 응답할 수 있다는 것을 표시하면, 클라이언트는 적어도 클라이언트 기능 패킷(패킷 타입 = 66) 및 유효 상태 응답 리스트 패킷(패킷 타입 = 139)을 전송할 수 있다. 일 실시예의 목적들을 위한 각각의 할당들에 따라서, 클라이언트에 의해 전송될 수 있고 상기 리스트에 포함될 수 있는 패킷 타입들은 아래와 같다:

66 - 클라이언트 기능 패킷.

133 - 알파 커서 이미지 기능 패킷.

139 - 클라이언트가 전송할 수 있는 기능 및 상태 패킷들의 정확한 타입들을 식별하는 유효 상태 응답 리스트 패킷.

140 - 패킷 처리 지연 파라미터 패킷.

141 - 개인용 디스플레이 기능 패킷.

142 - 클라이언트 에러 보고 패킷.

143 - 스케일 비디오 스트림 기능 패킷

144 - 클라이언트 식별 패킷.

145 - 대안 디스플레이 성능 패킷

패킷 타입 56 내지 63 - 제조자 별로 특정된 기능 및 상태 식별자들을 위해 사용될 수 있음.

CRC 필드들은 패킷 길이를 포함하는 패킷의 모든 바이트들의 CRC를 포함한다.

42. 패킷 처리 지연 파라미터 패킷

패킷 처리 지연 파라미터 패킷은 특정 패킷 타입의 수신과 관련된 프로세싱을 완료하기 위해 요구되는 시간을 호스트가 계산하도록 허용하기 위한 파라미터 세트를 제공한다. 호스트에 의해 전송된 몇몇 명령들은 제로 타임에서 클라이언트에 의해 완료될 수 없다. 호스트는 특정 함수들이 클라이언트에 의해 완료되었는지 여부를 결정하기 위해 디스플레이 요청 및 상태 패킷에 있는 상태 비트들을 폴링(poll)하거나, 또는 패킷 처리 지연 파라미터 패킷에 있는 클라이언트에 의해 리턴된 파라미터들을 이용하여 완료 시간을 계산할 수 있다. 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에 있는 140의 파라미터 값을 이용하여 패킷 처리 지연 파라미터 패킷을 지원하기 위한 자신의 기능을 표시한다.

패킷 처리 지연 파라미터 패킷의 일 실시예에 대한 포맷은 일반적으로 도 85A에 도시되어 있다. 도 85a에 도시된 바와 같이, 패킷 처리 지연 파라미터 패킷은 패킷 길이, 패킷 타입, cClient ID, 리스트 아이템 개수, 지연 파라미터 리스트 및 CRC 필드들을 포함하여 구성된다. 상기 패킷 타입에 대한 패킷 길이는 일반적으로 10의 값으로 고정되어 있으며, 140의 타입 값은 패킷 처리 지연 파라미터 패킷으로서 패킷을 식별한다. cClient ID 필드는 클라이언트 ID로서 나중에 사용하기 위해 예약되며, 일반적으로 0으로 설정된다. 2 바이트 리스트 아이템 개수 필드는 다음 유효 파라미터 응답 리스트에 있는 아이템들의 개수를 특정한다.

지연 파라미터 리스트 필드는 하나 이상의 지연 파라미터 리스트 아이템들을 포함하는 리스트이다. 하나의 지연 파라미터 리스트 아이템의 일 실시예에 대한 포맷은 도 85b에 도시되어 있으며, 여기에 지연, 픽셀 지연, 수형 픽셀 지연, 수직 픽셀 지연 및 고정 지연 필드들에 대한 패킷 타입이 도시되어 있다.

각각의 지연 파라미터 리스트 아이템은 일반적으로 정확하게 6 바이트의 길이를 가지도록 제한되며, 추가적으로 아래와 같이 정의된다. 지연 필드에 대한 2 바이트 패킷 타입은 다음 지연 파라미터들이 적용되는 패킷 타입을 특정한다. 픽셀 지연 필드(1 바이트)는 지연 값에 대한 인덱스를 포함한다. 상기 테이블로부터 판독된 값은 패킷의 목적지 필드에서 픽셀들의 총 개수에 의해 곱해진다. 픽셀들의 총 개수는 패킷에 의해 참조된 비트맵의 목적지 영역의 폭과 높이의 곱이다. 1 바이트 수평 픽셀 지연 필드는 지연 값 테이블(DPVL과 동일한 테이블)에 대한 인덱스인 값을 포함한다. 상기 테이블로부터 판독된 값은 패킷의 목적지 필드에서 (픽셀들의) 폭에 의해 곱해진다. 1 바이트 수직 픽셀 지연 필드는 지연 값 테이블(DPVL과 동일한 테이블)에 대한 인덱스인 값을 포함한다. 상기 테이블로부터 판독된 값은 패킷의 목적지 필드에서 (픽셀들의) 높이에 의해 곱해진다.

고정 지연 필드는 지연 값 테이블(DPVL과 동일한 테이블)에 대한 인덱스로서 1 바이트를 사용한다. 상기 테이블로부터 판독된 값은 패킷에서 특정된 임의의 파라미터 값들과 관련이 없는 패킷을 처리하기 위해 요구되는 시간을 나타내는 고정 지연 파라미터이다. 전체 지연 또는 패킷 처리 완료 시간 지연은 다음과 같은 관계에 따라 결정된다:

$$\text{Delay} = (\text{PacketProcessingDelay}(\text{PixelDelay}) \cdot \text{TotalPixels}) + (\text{PacketProcessingDelay}(\text{HorizontalPixelDelay}) \cdot \text{Width}) + (\text{PacketProcessingDelay}(\text{VerticalPixelDelay}) \cdot \text{Height}) + \text{PacketProcessingDelay}(\text{FixcdDelay})$$

몇몇 패킷들에 있어서, 전체 픽셀, 폭 또는 높이는 이러한 파라미터들이 대응하는 패킷에서 참조되지 않기 때문에 이용되지 않는다. 이러한 경우들에서, 대응하는 픽셀 지연 파라미터들은 일반적으로 0으로 설정된다.

43. 개인용 디스플레이 기능 패킷

개인용 디스플레이 기능 패킷은 헤드-마운트 디스플레이 또는 디스플레이 안경들과 같은, 개인용 디스플레이 장치의 기능들을 설명하는 파라미터 세트를 제공한다. 이것은 호스트가 클라이언트의 특정한 기능들에 따라 디스플레이 정보를 맞춤형으로 제공할 수 있도록 한다. 한편, 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에 있는 대응하는 파라미터를 이용함으로써 개인용 디스플레이 기능 패킷을 전송하기 위한 기능을 표시한다.

개인용 디스플레이 기능 패킷의 일 실시예에 대한 포맷은 일반적으로 도 86에 도시되어 있다. 도 86에 도시된 바와 같이, 개인용 디스플레이 기능 패킷은 패킷 길이, 패킷 타입, cClient ID, 서브-픽셀 레이아웃, 픽셀 형태, 수평 보기(Horizontal Field of View), 수직 보기(Vertical Field of View), 시각 축 크로싱, Lft. /Rt. 이미지, 투명성(See Through), 최대 밝기, 광학 기능, 최소 IPD, 최대 IPD, 필드 곡률(curvature) 포인트 리스트 및 CRC 필드들을 포함하여 구성된다. 일 실시예에서, 패킷 길이 필드 값은 68로 고정되어 있다. 141의 패킷 타입 값은 개인용 디스플레이 기능 패킷으로서 패킷을 식별한다. cClient ID 필드는 나중의 사용을 위해 예약되며 일반적으로 0으로 설정된다.

서브-픽셀 레이아웃 필드는 상부에서 하부로, 그리고 왼쪽에서 우측으로 서브-픽셀의 물리적 레이아웃을 특정하며, 다음과 같은 값들을 이용한다: 0은 서브-픽셀 레이아웃이 정의되지 않은 것을 표시한다; 1은 적색, 녹색, 청색 줄무늬를 표시한다; 2는 청색, 녹색, 적색 줄무늬를 표시한다; 3은 쿼드-픽셀(quad-pixel)을 표시하며, 쿼드 픽셀은 상부 좌측에 적색, 하부 우측에 청색, 하부 좌측과 상부 우측에 2개의 녹색이 배치된 2 대 2(2-by-2) 서브-픽셀을 가진다; 4는 쿼드 픽셀을 표시하며, 쿼드 픽셀은 하부 좌측에 적색, 상부 우측에 청색, 상부 좌측과 하부 우측에 2개의 녹색이 배치된 2 대 2 서브-픽셀을 가진다; 5는 델타(트라이어드(triad))를 표시한다; 6은 오버레이된 적색, 청색 및 녹색을 가진 모자이크(예를 들어, 필드-시퀀스 컬러를 가진 LCOS 디스플레이)를 표시한다; 값들 7 내지 255는 일반적으로 나중의 사용을 위해 예약되어 있다.

픽셀 형태 필드는 특정한 구성 서브-픽셀들로 구성된 각각의 픽셀의 형태를 특정하며, 다음과 같은 값들을 이용한다: 0은 서브-픽셀 형태가 정의되지 않은 것을 표시한다; 1은 원형을 표시한다; 2는 정사각형을 표시한다; 3은 직사각형을 표시한다; 4는 달걀 형태(oval)를 표시한다; 5는 타원 형태를 표시한다; 값들 6 내지 255는 원하는 형태를 표시하기 위한 나중의 사용을 위해 예약되며, 이는 당업자에 의해 이해될 수 있을 것이다.

1 바이트 수평 보기(HFOV) 필드는 0.5 도의 증가량으로 수평 보기를 특정한다(예를 들어, HFOV가 30도이면, 이러한 값은 60이다). 상기 값이 0이면, HFOV는 특정되지 않는다.

1 바이트 수직 보기(VFOV) 필드는 0.5 도의 증가량으로 수직 보기를 특정한다(예를 들어, VFOV가 30도이면, 이러한 값은 60이다). 상기 값이 0이면, VFOV는 특정되지 않는다.

1 바이트 시각 축 크로싱 필드는 0.01 디옵터(diopeter)(1/m) 증가량으로 시각 축 크로싱을 특정한다(예를 들어, 시각 축 크로싱이 2.22 미터이면, 이러한 값은 45이다). 상기 값이 0이면, 시각 축 크로싱은 특정되지 않는다.

1 바이트 좌측/우측 이미지 오버랩 필드는 좌측과 우측 이미지의 오버랩 백분율을 특정한다. 허용가능한 이미지 오버랩 범위는 1%부터 100%까지이다. 101에서 255까지의 값은 유효하지 않으므로 일반적으로 사용되지 않아야 한다. 상기 값이 0이면, 이미지 오버랩은 특정되지 않는다.

1 바이트 투명도 필드는 이미지의 투명도 백분율을 특정한다. 허용가능한 투명도의 범위는 0%에서 100%까지이다. 101에서 254까지의 값은 유효하지 않으므로 사용되지 않아야 한다. 상기 값이 255이면, 투명도 백분율은 특정되지 않는다. 1 바이트 최대 밝기 필드는 20 니트(nit)의 증가량으로 최대 밝기를 특정한다(예를 들어, 최대 밝기가 100 니트이면, 상기 값은 5이다). 상기 값이 0이면, 최대 밝기는 특정되지 않는다.

2 바이트 광학 기능 플래그 필드는 디스플레이의 광학 기능들을 특징하는 다양한 필드들을 포함한다. 이러한 비트 값들은 일반적으로 다음과 같이 할당된다:

비트들 15 내지 5는 나중의 사용을 위해 예약되며, 일반적으로 논리 0 상태로 설정된다.

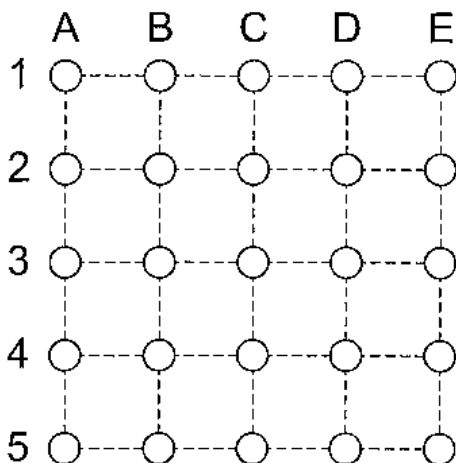
비트 4는 접안 렌즈 초점 조절을 선택하는데, '0'값은 디스플레이에 접안 렌즈 초점 조절 기능이 없는 것을 의미하며, '1'값은 디스플레이에 접안 렌즈 초점 조절 기능이 없는 것을 의미한다.

비트들 3 내지 2는 다음의 값들에 따라 쌍안경 기능을 선택한다: 0값은 디스플레이가 쌍안용이며 2차원(2D) 이미지들만을 디스플레이할 수 있다는 것을 의미한다; 1값은 디스플레이가 쌍안용이며 3차원(3D) 이미지들을 디스플레이할 수 있다는 것을 의미한다; 2값은 디스플레이가 단안용인 것을 의미하며; 3값은 나중의 사용을 위해 예약되어 있다.

비트들 1 내지 0은 좌측-우측 필드 곡률 대칭을 선택하며, 여기서 0값은 필드 곡률이 정의되지 않은 것을 의미한다. 상기 필드가 0이면, A1 내지 E5의 모든 필드 곡률 값들은 포인트 C3를 제외하고 0으로 설정되어야 하며, 여기서 포인트 C3는 디스플레이의 초점 거리를 특징하거나 또는 초점 거리가 특정되지 않은 것을 표시하기 위해 0으로 설정되어야 한다. 1값은 좌측 및 우측 디스플레이들이 동일한 대칭을 가지는 것을 의미한다; 2값은 좌측 및 우측 디스플레이들이 수직 축(열 C)을 통해 반사(mirrored)되는 것을 의미한다; 3값은 나중의 사용을 위해 예약되어 있다.

1 바이트 동공간 거리(IPD: Inter-Pupillary Distance) 최소 필드는 밀리미터(mm) 단위로 최소 동공간 거리를 특징한다. 상기 값이 0이면, 최소 동공간 거리는 특정되지 않는다. 1 바이트 동공간 거리(IPD) 최대 필드는 밀리미터(mm) 단위로 최대 동공간 거리를 특징한다. 상기 값이 0이면, 최대 동공간 거리는 특정되지 않는다.

필드 곡률 포인트 리스트 필드는 1부터 65535까지의 범위를 가지고 디오퍼(1/m)의 천 분의 일 단위로 초점 거리를 특징하는 25개의 2 바이트 파라미터들의 리스트를 포함한다(예를 들어, 1이 0.001 디오퍼이고, 65535는 65.535 디오퍼이다). 필드 곡률 포인트 리스트에 있는 25개의 엘리먼트들은 아래에 도시된 바와 같이 A1 내지 E5로 라벨링되어 있다. 포인트들은 디스플레이의 액티브 영역에 균일하게 분포되어야 한다. 열 C는 디스플레이의 수직 축에 대응하며, 행 3은 디스플레이의 수평 축에 대응한다. 열들 A 및 E는 각각 디스플레이의 좌측 및 우측 에지에 대응한다. 행들 1 및 5는 각각 디스플레이의 상부 및 하부 에지에 대응한다. 리스트에 있는 25개의 포인트들의 순서는 다음과 같다: A1, B1, C1, D1, E1, A2, B2, C2, D2, E2, A3, B3, C3, D3, E3, A4, B4, C4, D4, E4, A5, B5, C5, D5, E5.



CRC 필드는 패킷 길이를 포함하는 패킷의 모든 바이트들의 CRC를 포함한다.

44. 클라이언트 에러 보고 패킷

클라이언트 에러 보고 패킷은 클라이언트가 호스트로 동작 에러들의 리스트를 제공하도록 허용하기 위한 메커니즘 또는 수단으로서 동작한다. 클라이언트는 호스트로부터 특정 명령들을 수신한 결과로서 자신의 정상적인 동작 중에 발생하는 넓은 범위의 에러들을 탐지할 수 있다. 이러한 에러들의 예들은 다음과 같은 사항들을 포함한다: 클라이언트는 자신이 지원하지 않는 모드에서 동작하도록 지시받았을 수 있고, 클라이언트는 범위를 벗어나거나 또는 클라이언트의 기능을 벗어

난 특정 파라미터들을 포함하는 패킷을 수신하였을 수 있으며, 클라이언트는 부적당한 시퀀스로 모드에 진입하도록 지시 받았을 수 있다. 클라이언트 에러 보고 패킷은 정상적인 동작 동안에 에러들을 탐지하기 위해 사용될 수 있으며, 호스트 및 클라이언트 시스템들을 개발하고 통합하는데 있어서 발생하는 문제들을 진단하기 위해 시스템 설계자 및 통합자들에게 가장 유용하다. 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에 있는 142의 파라미터 값을 이용하여 클라이언트 에러 보고 패킷을 전송하기 위한 자신의 기능을 표시한다.

클라이언트 에러 보고 패킷의 일 실시예에 대한 포맷은 도 87a에 도시되어 있다. 도 87a에 도시된 바와 같이, 클라이언트 에러 보고 패킷은 패킷 길이, 패킷 타입, cClient ID, 리스트 아이템 개수, 에러 코드 리스트 및 CRC 필드들을 포함하여 구성된다. 142의 패킷 타입 값은 클라이언트 에러 보고 패킷으로서 패킷을 식별한다. cClient ID 필드는 나중의 사용을 위해 예약되어 있으며 일반적으로 현재 0으로 설정되어 있다. 리스트 아이템 개수 필드(2 바이트)는 다음 에러 코드 리스트에 있는 아이템들의 개수를 특정한다. 에러 코드 리스트 필드(여기서, 8 바이트)는 하나 이상의 에러 보고 리스트 아이템들을 포함하는 리스트이다. 단일 에러 보고 리스트 아이템의 포맷은 도 87b에 도시되어 있다.

일 실시예에서, 도 87b에 도시된 바와 같이, 각각의 에러 보고 리스트 아이템은 정확하게 4 바이트의 길이를 가지며, 일 실시예에서 다음과 같은 필드들을 포함하는 구조를 가진다: 보고되는 에러의 타입을 특정하는 2 바이트의 디스플레이 에러 코드 필드, 클라이언트 에러 보고 패킷에 의해 정의된 에러와 관련하여 보다 상세한 레벨로 특정하는 2 바이트의 에러 서브-코드 필드. 각각의 클라이언트 에러 코드에 대한 특정한 정의는 클라이언트의 제조자에 의해 정의된다. 에러 서브-코드는 모든 디스플레이 에러 코드에 대하여 정의될 필요는 없으며, 에러 서브-코드가 정의되지 않은 경우에는 값이 0으로 설정된다. 각각의 에러 서브-코드에 대한 특정한 정의는 클라이언트의 제조자에 의해 정의된다.

45. 클라이언트 식별 패킷

클라이언트 식별 패킷은 요청 특정 상태 패킷에 응답하여 클라이언트가 식별 데이터를 리턴하도록 허용한다. 일 실시예에서, 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에 있는 144의 파라미터 값을 이용하여 디스플레이 식별 패킷을 전송하기 위한 기능을 표시한다. 호스트가 클라이언트로부터 상기 데이터를 판독함으로써 클라이언트 장치 제조자 명칭 및 모델 번호를 결정할 수 있도록 하는 것은 호스트에 유용하다. 상기 정보는 클라이언트가 클라이언트 기능 패킷에서 기술될 수 없는 특별한 기능들을 가지고 있는지 여부를 결정하기 위해 사용될 수 있다. 클라이언트로부터 식별 정보를 판독하기 위한 두 가지의 방법들, 수단들 또는 메커니즘들이 존재할 수 있다. 하나는 베이스 EDID 구조의 필드들과 유사한 필드들을 포함하는 클라이언트 기능 패킷을 사용하는 것이다. 다른 방법은 클라이언트 기능 패킷의 유사한 필드들과 비교되는 보다 풍부한 정보 세트를 포함하는 클라이언트 식별 패킷을 사용하는 것이다. 이것은 호스트가 3-캐릭터 EISA 코드로 할당되지 않았던 제조자들을 식별하도록 허용하며, 시리얼 번호들이 영숫자 캐릭터들을 포함하도록 허용한다.

클라이언트 식별 패킷의 일 실시예에 대한 포맷은 일반적으로 도 88에 도시되어 있다. 도 88에 도시된 바와 같이, 클라이언트 식별 패킷은 패킷 길이, 패킷 타입, cClient ID, 제조주(Week of Manufacture), 제조 연도, 제조자 명칭 길이, 제품 명칭 길이, 시리얼 번호 길이, 제조자 명칭 스트링, 제품 명칭 스트림, 시리얼 번호 스트링 및 CRC 필드들을 포함하여 구성된다.

2 바이트의 패킷 타입 필드는 디스플레이 식별 패킷으로서 패킷을 식별하는 값을 포함한다. 상기 값은 일 실시예에서 144가 되도록 선택된다. cClient ID 필드(2 바이트)는 다시 클라이언트 ID를 위한 나중의 사용을 위해 예약되어 있으며, 일반적으로 0으로 설정되어 있다. CRC 필드(2 바이트)는 패킷 길이를 포함하는 패킷의 모든 바이트들의 16 비트 CRC를 포함한다.

1 바이트의 제조주 필드는 디스플레이의 제조주를 정의하는 값을 포함한다. 적어도 하나의 실시예에서, 상기 값은 클라이언트에 의해 지원되는 경우에 1에서 53까지의 범위에 있다. 상기 필드가 클라이언트에 의해 지원되지 않으면, 상기 필드는 일반적으로 0으로 설정된다. 1 바이트의 제조 연도 필드는 클라이언트(디스플레이)의 제조 연도를 정의하는 값을 포함한다. 상기 값은 다른 기준 년도들이 사용될 수 있음에도 불구하고, 시작 포인트로서 1990년부터 오프셋되어 있다. 상기 필드에 의해 1991년부터 2245년까지의 범위에 있는 연도가 표현될 수 있다. 예를 들어, 2003년도는 13의 제조 연도 값에 대응한다. 상기 필드가 클라이언트에 의해 지원되지 않으면, 0값으로 설정되어야 할 것이다.

제조자 명칭 길이, 제품 명칭 길이 및 시리얼 번호 길이 필드들은 각각 임의의 널(null) 종결 또는 널 패드 캐릭터들을 포함하는 제조자 명칭 스트링 필드의 길이를 특정하고, 임의의 널 종결 또는 널 패드 캐릭터들을 포함하는 제품 명칭 스트링 필드의 길이를 특정하고, 임의의 널 종결 또는 널 패드 캐릭터들을 포함하는 시리얼 번호 스트링 필드의 길이를 특정하는 2 바이트 값들을 포함한다.

제조사 명칭 스트링, 제품 명칭 스트링 및 시리얼 번호 스트링 필드들은 각각 디스플레이의 제조자, 제품 명칭 및 영숫자 시리얼 번호를 특정하는 ASCII 스트링을 포함하는, 각각의 제조자 명칭 길이, 제품 명칭 길이 및 시리얼 번호 길이 필드들에 의해 특정된 가변적인 바이트 개수를 포함한다. 이러한 스트링들 각각은 적어도 하나의 널 캐릭터에 의해 종결된다.

46. 대안 디스플레이 기능 패킷

대안 디스플레이 기능 패킷은 MDDI 클라이언트 제어기에 부착된 대안 디스플레이들의 기능을 표시한다. 대안 디스플레이 기능 패킷은 요청 특정 상태 패킷에 응답하여 전송된다. 요청되면, 클라이언트 장치는 지원되는 각각의 대안 디스플레이에 대하여 대안 디스플레이 기능 패킷을 전송한다. 클라이언트는 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트에 있는 145의 파라미터 값을 통해 대안 디스플레이 기능 패킷을 전송하기 위한 자신의 기능을 표시한다.

내부 모드에서 동작되는 MDDI 시스템들에 대하여, MDDI 클라이언트 제어기에 연결된 하나 이상의 디스플레이를 포함하는 것이 일반적일 수 있다. 예시적인 애플리케이션은 플립(flip)의 안쪽에 큰 디스플레이를 가지며 바깥쪽에 작은 디스플레이를 가지는 모바일 폰이다. 두 가지 잠재적인 이유로, 내부 모드가 반드시 대안 디스플레이 기능 패킷으로 돌아갈 필요는 없다. 첫째, 호스트는 이미 프로그래밍 되었을 수도 있고 아니면 기능들이 공통 장치나 하우징에 사용되기 때문에 제조 동안에 기능을 통보받을 수도 있다. 둘째, 2개의 어셈블리로 인해, 클라이언트는 호스트로의 접속이 쉽게 단절되거나 분리될 수 없으며, 호스트는 클라이언트 기능의 하드-코딩된 사본을 포함할 수도 있고, 또는 달리 일어날 수도 있지만, 적어도 클라이언트의 변경에 따라 변하지 않는다는 점을 알 수 있다.

클라이언트 기능 패킷의 대안 디스플레이 필드의 수는 하나 이상의 디스플레이 부착된 것을 보고하기 위해 이용될 수 있으며 대안 디스플레이 기능 패킷은 각각의 대안 디스플레이의 기능을 보고한다. 비디오 스트림 패킷은 클라이언트 장치로 각각의 대안 디스플레이를 어드레싱하기 위해 픽셀 데이터 속성 필드에 4 비트를 포함한다.

대안 디스플레이 기능 패킷의 일 실시예에 대한 포맷은 일반적으로 도 89에 도시되어 있다. 도 89에 도시된 바와 같이, 대안 디스플레이 기능 패킷은 패킷 길이, 패킷 타입, cClient ID, 대안 디스플레이 번호, 예약 1, 비트맵 폭, 비트맵 높이, 디스플레이 윈도우 폭, 디스플레이 윈도우 높이, 컬러 맵 RGB 폭, 컬러 맵 RGB 높이, RGB 기능, 단색 용량, 예약 2, Y Cb Cr 기능, 디스플레이 특성 기능, 예약 3 및 CRC 필드들을 포함하여 구성된다. 145의 패킷 타입 값은 대안 디스플레이 기능 패킷으로서 패킷을 식별한다. cClient ID는 클라이언트 ID를 위한 나중의 사용을 위해 예약되어 있으며 일반적으로 0으로 설정되어 있다.

대안 디스플레이 번호 필드는 0부터 15까지의 범위에 있는 정수를 이용하여 대안 디스플레이에 대한 식별을 표시하기 위해 1 바이트를 사용한다. 제 1 대안 디스플레이는 번호 0이 될 것이며 다른 대안 디스플레이들은 대안 디스플레이들의 총 개수에서 1을 뺀 값으로 사용되는 최대값을 가지고 유일한 대안 디스플레이 번호 값들을 통해 식별될 것이다. 대안 디스플레이들의 총 개수에서 1을 뺀 값보다 큰 값은 사용되지 않을 것이다. 예를 들어, 제 1 디스플레이와 MDDI 클라이언트에 연결된 콜러(caller)-ID 디스플레이를 가지는 모바일 폰은 하나의 대안 디스플레이를 가지고 있으며, 여기서 콜러-ID 디스플레이의 대안 디스플레이 번호는 0이며 디스플레이 기능 패킷의 대안 디스플레이 필드의 번호는 1 값을 가진다.

예약 1 필드(1 바이트)는 나중의 사용을 위해 예약되어 있다. 상기 필드에 있는 모든 비트들은 0으로 설정되어 있다. 상기 필드의 하나의 목적은 모든 다음 2 바이트 필드들이 16 비트 워드 주소로 할당되도록 하고 4 비트 필드들이 32 비트 워드 주소로 할당되도록 하는 것이다.

비트맵 폭 필드는 다수의 픽셀로서 표현되는 비트맵의 폭을 특정하는 2 바이트를 사용한다. 비트맵 높이 필드는 다수의 픽셀로서 표현되는 비트맵의 높이를 특정하는 2 바이트를 사용한다. 디스플레이 윈도우 폭 필드는 다수의 픽셀로서 표현되는 디스플레이 윈도우의 폭을 특정하는 2 바이트를 사용한다. 디스플레이 윈도우 높이 필드는 다수의 픽셀로서 표현되는 디스플레이 윈도우의 높이를 특정하는 2 바이트를 사용한다.

컬러 맵 RGB 폭 필드는 컬러 맵(팔레트) 디스플레이 모드에서 디스플레이될 수 있는 적색, 녹색 및 청색 컬러 컴포넌트들의 비트들의 수를 특정하는 2 바이트를 사용한다. 각 컬러 컴포넌트(적색, 녹색 및 청색)에 대하여 최대 8 비트가 사용될 수 있다. 각각의 컬러 컴포넌트의 8 비트가 컬러 맵 패킷으로 전송되더라도, 상기 필드에서 정의된 각각의 컬러 컴포넌트의 최하위 비트들의 수만이 사용된다. 디스플레이 클라이언트가 컬러 맵(팔레트) 포맷을 사용할 수 없다면, 상기 값은 0이다. 컬러 맵 RGB 폭 워드는 세 개의 개별적인 양의 값들로 구성되어 있다:

비트들 3 내지 0은 유효하게 간주되는 0부터 8까지의 값들을 이용하여 각각의 픽셀에 있는 청색 비트들의 최대 개수를 정의한다. 비트들 7 내지 4는 유효하게 간주되는 0부터 8까지의 값들을 이용하여 각각의 픽셀에 있는 녹색 비트들의 최대 개수를 정의한다. 비트들 11 내지 8은 유효하게 간주되는 0부터 8까지의 값들을 이용하여 각각의 픽셀에 있는 적색 비트들의 최대 개수를 정의한다. 비트들 14 내지 12는 나중의 사용을 위해 예약되어 있으며 일반적으로 0으로 설정되어 있다. 비트 15는 패킹되거나 또는 패킹되지 않은 포맷으로 구성된 컬러 맵 픽셀 데이터를 수용하기 위한 클라이언트의 기능을 표시하기 위해 사용된다. 비트 15가 로직-1 레벨로 설정되면, 이것은 클라이언트가 패킹되거나 또는 패킹되지 않은 포맷으로 구성된 컬러 맵 픽셀 데이터를 수용할 수 있다는 것을 표시한다. 비트 15가 로직-0으로 설정되면, 이것은 클라이언트가 오직 패킹되지 않은 포맷으로 구성된 컬러 맵 픽셀 데이터를 수용할 수 있다는 것을 표시한다.

RGB 기능 필드는 RGB 포맷으로 디스플레이될 수 있는 해상도의 비트들의 수를 특정하기 위해 2 바이트를 사용한다. 일 실시예에서, 클라이언트가 RGB 포맷을 사용할 수 없다면, 상기 값은 0으로 설정된다. RGB 기능 워드는 세 개의 개별적인 양의 값들로 구성된다: 비트들 3 내지 0은 각 픽셀에 있는 청색에 대한 비트들의 최대 개수(청색 밀도)를 정의하고, 비트들 7 내지 4는 각 픽셀에 있는 녹색에 대한 비트들의 최대 개수(녹색 밀도)를 정의하고, 비트들 11 내지 8은 각 픽셀에 있는 적색에 대한 비트들의 최대 개수(적색 밀도)를 정의한다. 비트들 14 내지 12는 나중의 사용을 위해 예약되어 있으며 0으로 설정되어 있다. 비트 15는 패킹되거나 또는 패킹되지 않은 포맷으로 구성된 RGB 픽셀 데이터를 수용하기 위한 클라이언트의 기능을 표시하기 위해 사용된다. 비트 15가 로직-1 레벨로 설정되면, 이것은 클라이언트가 패킹되거나 또는 패킹되지 않은 포맷으로 구성된 RGB 픽셀 데이터를 수용할 수 있다는 것을 표시한다. 비트 15가 로직-0으로 설정되면, 이것은 클라이언트가 오직 패킹되지 않은 포맷으로 구성된 RGB 픽셀 데이터를 수용할 수 있다는 것을 표시한다.

1 바이트의 단색 용량 필드는 단색 포맷으로 디스플레이될 수 있는 해상도의 비트들의 수를 특정하기 위한 값 또는 정보를 포함한다. 클라이언트가 단색 포맷을 사용할 수 없다면, 상기 값은 0으로 설정된다. 비트들 6 내지 4는 나중의 사용을 위해 예약되어 있으며 일반적으로 0으로 설정되어 있다. 비트들 3 내지 0은 각각의 픽셀에 존재할 수 있는 그레이스케일의 비트들의 최대 개수를 정의한다. 이러한 네 개의 비트들은 각 픽셀이 1개부터 15개까지의 비트들로 구성되는 것을 특정할 수 있도록 한다. 상기 값이 0이면 단색 포맷은 클라이언트에 의해 지원되지 않는다. 비트 7이 1로 설정되면, 비트 7은 클라이언트가 패킹되거나 또는 패킹되지 않은 포맷으로 구성된 단색 픽셀 데이터를 수용할 수 있다는 것을 표시한다. 비트 7이 0으로 설정되면 이것은 클라이언트가 오직 패킹되지 않은 포맷으로 구성된 단색 픽셀 데이터를 수용할 수 있다는 것을 표시한다.

예약 2 필드는 나중의 사용을 위해 예약된 1 바이트의 필드이며 상기 필드의 모든 비트들은 일반적으로 로직-0 레벨로 설정되어 있다. 일 실시예에서, 상기 필드의 하나의 목적은 모든 다음 2 바이트 필드들이 16 비트 워드 주소로 할당되도록 하고 4 바이트 필드들이 32 비트 워드 주소로 할당되도록 하는 것이다.

2 바이트의 Y Cb Cr 기능 필드는 Y Cb Cr 포맷으로 디스플레이될 수 있는 해상도의 비트들의 수를 특정한다. 클라이언트가 Y Cb Cr 포맷을 사용할 수 없다면, 상기 값은 0이다. Y Cb Cr 기능 워드는 세 개의 개별적인 양의 값들로 구성된다: 비트들 3 내지 0은 Cb 샘플을 특정하는 비트들의 최대 개수를 정의하고, 비트들 7 내지 4는 Cr 샘플을 특정하는 비트들의 최대 개수를 정의하고, 비트들 11 내지 8은 Y 샘플을 특정하는 비트들의 최대 개수를 정의하며, 비트들 14 내지 12는 나중의 사용을 위해 예약되며 0으로 설정되어 있다. 비트 15가 0으로 설정되면, 비트 15는 클라이언트가 패킹되거나 또는 패킹되지 않은 포맷으로 구성된 Y Cb Cr 픽셀 데이터를 수용할 수 있다는 것을 표시한다. 비트 15가 0으로 설정되면, 클라이언트가 오직 패킹되지 않은 포맷으로 구성된 Y Cb Cr 픽셀 데이터를 수용할 수 있다는 것을 표시한다.

1 바이트의 베이어(bayer) 기능 필드는 베이어 포맷으로 전달될 수 있는 해상도의 비트들의 수, 픽셀 그룹 및 픽셀 순서를 특정한다. 클라이언트가 베이어 포맷을 사용할 수 없다면, 상기 값은 0으로 설정된다. 베이어 기능 필드는 다음의 값들로 구성된다: 비트들 3 내지 0은 각각의 픽셀에 존재하는 강도의 비트들의 최대 개수를 정의하고, 비트들 5 내지 4는 요구될 수 있는 픽셀 그룹 패턴을 정의하고, 비트들 8 내지 6은 요구되는 픽셀 순서를 정의하며, 비트들 14 내지 9는 나중의 사용을 위해 예약되고 0으로 설정되어 있다. 비트 15가 1로 설정되면, 비트 15는 클라이언트가 패킹되거나 또는 패킹되지 않은 포맷으로 구성된 베이어 픽셀 데이터를 수용할 수 있다는 것을 표시한다. 비트 15가 0으로 설정되면, 이것은 클라이언트가 오직 패킹되지 않은 포맷으로 구성된 베이어 픽셀 데이터를 수용할 수 있다는 것을 표시한다.

2 바이트 CRC 필드는 패킷 길이를 포함하는 패킷의 모든 바이트들의 16 비트 CRC를 포함한다.

47. 레지스터 액세스 패킷

레지스터 액세스 패킷은 MDDI 링크의 반대쪽 엔드에 있는 구성 및 상태 레지스터들로 액세스하기 위한 수단, 메커니즘 또는 방법을 호스트 또는 클라이언트로 제공한다. 이러한 레지스터들은 각각의 디스플레이 또는 장치 제어기에 대하여 유일할 수 있다. 이러한 레지스터들은 이미 설정 구성, 동작 모드를 요구하는 많은 디스플레이 내에 존재하며, 다른 유용하고 필요한 설정들을 포함한다. 레지스터 액세스 패킷은 MDDI 호스트 또는 클라이언트가 MDDI 링크를 이용하여 레지스터에 기록하고 레지스터의 판독을 요청하도록 허용한다. 호스트 또는 클라이언트가 레지스터의 판독을 요청하면, 반대쪽 엔드는 동일한 패킷 타입의 레지스터 데이터를 전송함으로써, 또한 판독/기록 정보 필드의 사용을 통해 이것이 특정 레지스터로부터 판독된 데이터임을 표시함으로써 응답하여야 한다. 레지스터 액세스 패킷은 1보다 큰 레지스터 카운터를 특정함으로써 다수의 레지스터에 대하여 판독 또는 기록을 수행하기 위해 사용될 수 있다. 클라이언트는 클라이언트 기능 패킷의 클라이언트 특성 기능 필드의 비트 22를 이용하여 레지스터 액세스 패킷을 지원하기 위한 기능을 표시한다.

레지스터 액세스 패킷의 일 실시예에 대한 포맷은 도 90에 도시되어 있다. 도 90에 도시된 바와 같이, 레지스터 액세스 패킷은 패킷 길이, 패킷 타입, bClient ID, 판독/기록 플래그, 레지스터 주소, 파라미터 CRC, 레지스터 데이터 리스트 및 레지스터 데이터 CRC 필드들을 포함하여 구성된다. 146의 패킷 타입 값은 레지스터 액세스 패킷으로서 패킷을 식별한다. bClient ID는 나중의 사용을 위해 예약되어 있으며 일반적으로 현재 0으로 설정되어 있다.

2 바이트의 판독/기록 플래그 필드는 기록 또는 판독, 또는 판독에 대한 응답으로서 특정 패킷을 특정하며, 데이터 값들에 대한 카운트를 제공한다.

비트들 15 내지 14는 판독/기록 플래그로서 동작한다. 비트들 [15:14]가 '00'이면, 상기 패킷은 레지스터 주소 필드에 의해 어드레싱된 레지스터로 기록될 데이터를 포함한다. 특정 레지스터들로 기록될 데이터는 레지스터 데이터 리스트 필드에 포함된다. 비트들 [15:14]가 '10'이면, 이것은 레지스터 주소 필드에 의해 어드레싱된 하나 이상의 레지스터들의 데이터에 대한 요청이다. 비트들 [15:14]가 '11'이면, 상기 패킷은 '10'으로 설정된 판독/기록 플래그들의 비트들 15:14를 가지는 레지스터 액세스 패킷에 응답하여 요청되었던 데이터를 포함한다. 레지스터 주소 필드는 제 1 레지스터 데이터 리스트 아이템에 대응하는 레지스터의 주소를 포함하며, 레지스터 데이터 리스트 필드는 상기 주소 또는 주소들로부터 판독되었던 데이터를 포함한다. 비트들 [15:14]가 '01'이면, 이것은 유효하지 않은 값으로 취급되며, 상기 값은 나중의 사용을 위해 예약되고 사용되지 않는다.

비트들 13:0은 레지스터 데이터 리스트 필드에서 전달될 32 비트 레지스터 데이터 아이템들의 개수를 특정하기 위해 14 비트의 양의 정수를 사용한다. 비트들 15:14가 '00'이면, 비트들 13:0은 레지스터 주소 필드에 의해 특정된 레지스터에서 시작하는 레지스터들로 기록되도록 하기 위해 레지스터 데이터 리스트 필드에 포함된 32 비트 레지스터 데이터 아이템들의 개수를 특정한다. 비트들 15:14가 '10'이면, 비트들 13:0은 수신 장치가 레지스터들이 판독되도록 요청하는 장치로 전송하는 32 비트 레지스터 데이터 아이템들의 개수를 특정한다. 상기 패킷의 레지스터 데이터 리스트 필드는 아이템들을 포함하지 않으며 길이가 0이다. 비트들 15:14가 '11'이면, 비트들 13:0은 레지스터 데이터 리스트 필드에 포함된 레지스터들로부터 판독되었던 32 비트 레지스터 데이터 아이템들의 개수를 특정한다. 비트들 15:14는, 유효하지 않은 값으로 취급되는, '01'로 현재 설정되어 있지 않으며, 그렇지 않으면 나중의 지정 또는 사용을 위해 예약된다.

레지스터 주소 필드는 기록하거나 판독할 레지스터 주소를 표시하기 위해 4 바이트를 사용한다. 어드레싱이 32 비트보다 적은 어드레싱 레지스터들에 있어서, 상위 비트들은 0으로 설정된다.

2 바이트의 파라미터 CRC 필드는 패킷 길이부터 레지스터 주소까지의 모든 바이트들의 CRC를 포함한다. 이러한 CRC 검사가 실패하면, 전체 패킷이 버려지게 된다.

레지스터 데이터 리스트 필드는 클라이언트 레지스터들로 기록될 4 바이트 레지스터 데이터 값들의 리스트 또는 클라이언트 장치 레지스터들로부터 판독되었던 값들을 포함한다.

2 바이트의 레지스터 데이터 CRC 필드는 오직 레지스터 데이터 리스트의 CRC를 포함한다. 이러한 CRC 검사가 실패하면, 레지스터 데이터는 여전히 사용될 수 있으나, CRC 에러 카운트는 증가하게 된다.

D. 패킷 CRC

CRC 필드들은 패킷들의 끝 부분에 위치하고 때때로 상당히 큰 데이터 필드를 가질 수 있는 패킷들의 특정한 중요 파라미터들 뒤에 위치할 수 있으며, 그리하여 전달하는 동안에 에러들이 발생할 가능성이 커진다. 두 개의 CRC 필드들을 가지는 패킷들에서, 하나의 CRC 필드만이 사용되는 경우에, CRC 생성기는 제 1 CRC 후에 재-초기화되며, 그 결과 긴 데이터 필드 뒤에 오는 CRC 계산들은 패킷의 시작 부분에서 파라미터들에 의해 영향을 받지 않는다.

예시적인 실시예에서, CRC 계산을 위해 사용되는 다항식은 CRC-16 또는 $X^{16} + X^{15} + X^2 + X^0$ 으로 알려져 있다. 본 발명을 구현하기 위해 유용한 CRC 생성기 및 검사기(3600)의 샘플 구현은 도 36에 도시되어 있다. 도 36에서, CRC 레지스터(3602)는 Tx_MDDI_Data_Before_CRC 라인을 통한 입력인 패킷의 제 1 비트를 전달하기 전에 0x0001의 값으로 초기화되며, 그 후에 패킷의 바이트들이 LSB와 함께 먼저 시작하는 레지스터로 쉬프트된다. 상기 도면의 레지스터 비트 번호들은 사용되는 다항식의 순서에 대응하며, MDDI에 의해 사용되는 비트 위치들에 대응하지 않는다는 것을 유의하도록 한다. CRC 레지스터를 한쪽 방향으로 쉬프트하는 것이 보다 효율적이며, 이것은 CRC 비트 15가 MDDI CRC 필드의 비트 위치 0에 위치하도록 하고 CRC 레지스터 비트 14가 MDDI CRC 필드의 비트 위치 1에 위치하도록 하며, MDDI 비트 위치 14에 도달할 때까지 수행된다.

예시적으로, 디스플레이 요청 및 상태 패킷들에 대한 패킷 콘텐츠가 0x07, 0x46, 0x000400, 0x00이고 (또는 0x07, 0x00, 0x46, 0x00, 0x04, 0x00, 0x00과 같은 바이트 시퀀스로서 표현되고), 멀티플렉서들(3604 및 3606) 및 NAND 게이트(3608)의 입력들을 이용하여 제공되면, Tx_MDDI_Data_With_CRC 라인을 통한 결과적인 CRC 출력은 0x0ea1이다 (또는 0xa1, 0x0e의 시퀀스로서 표현된다).

CRC 생성기 및 검사기(3600)가 CRC 검사기로서 구성되면, Rx_MDDI_Data 라인을 통해 수신된 CRC는 멀티플렉서(3604)와 NAND 게이트(3608)에 대한 입력이며, NOR 게이트(3610), 배타적-OR(XOR) 게이트(3612) 및 AND 게이트(3614)를 사용하여 CRC 레지스터에서 검색된 값과 비트 단위로 비교된다. AND 게이트(3614)에 의한 출력으로서, 임의의 에러들이 존재하면, CRC는 게이트(3614)의 출력을 레지스터(3602)의 입력으로 연결함으로써 CRC 에러를 포함하는 모든 패킷에 대하여 한 번씩 증가하게 된다. 도 36의 다이어그램에서 도시된 예시적인 회로는 주어진 CHECK_CRC_NOW 원도우(도 37b 참조) 내에서 하나 이상의 CRC 에러 신호를 출력할 수 있다. 그러므로 CRC 에러 카운터는 일반적으로 CHECK_CRC_NOW가 액티브 상태인 각각의 간격 내에서 제 1 CRC 에러 인스턴스만을 카운트한다. CRC 생성기로서 구성된 경우에, CRC는 패킷의 끝 부분과 일치하는 시점에서 CRC 레지스터로부터 클로킹(clocked)된다.

입력 및 출력 신호들과 인에이블 신호들에 대한 타이밍은 도 37a 및 37b에 도표로 도시되어 있다. CRC의 생성과 데이터 패킷의 전송은 Gen_Reset, Check_CRC_Now, Generate_CRC_Now 및 Sending_MDDI_Data 신호들의 상태(0 또는 1)와 Tx_MDDI_Data_Before_CRC 및 Tx_MDDI_Data_With_CRC 신호들과 함께 도 37a에 도시되어 있다. 데이터 패킷의 수신과 CRC 값의 검사는 Gen_Reset, Check_CRC_Now, Generate_CRC_Now 및 Sending_MDDI_Data 신호들의 상태와 Rx_MDDI_Data 및 CRC 에러 신호들과 함께 도 37b에 도시되어 있다.

E. 패킷 CRC에 대한 에러 코드 오버헤드

오직 데이터 패킷들과 CRC만이 호스트와 클라이언트 사이에서 전달될 때마다, 에러 코드들은 발생하지 않는다. 유일한 에러는 동기화 손실이다. 다시 말하면, 양호한 데이터 전달 경로 또는 파이프라인의 부재로 인한 타임아웃으로 링크를 리셋하고 진행하기 위해 한쪽은 링크를 기다려야 한다. 불행하게도, 이것은 시간 낭비이며 다소 비효율적이다.

일 실시예에서의 사용을 위해, 패킷들의 CRC 부분이 에러 코드 정보를 전달하기 위해 사용되는 새로운 기법이 개발되었다. 이것은 일반적으로 도 65에 도시되어 있다. 즉, 하나 이상의 에러 코드들은 통신 프로세싱 또는 링크에서 발생할 수 있는 특정한 미리 정의된 에러들 또는 흠결들을 표시하는 데이터 전달을 처리하는 프로세서들 또는 장치들에 의해 생성된다. 에러가 발생하면, 적절한 에러 코드가 패킷의 CRC에 대한 비트들을 사용하여 생성되고 전달된다. 즉, CRC 값은 원하는 에러 코드를 통해 오버로딩되거나 겹쳐 쓰이게(overwritten) 되며, 원하는 에러 코드는 CRC 필드의 값들을 모니터링하는 에러 모니터 또는 검사기에 의해 수신측 엔드를 통해 탐지될 수 있다. 에러 코드가 몇몇 이유들을 위해 CRC 값을 매칭시키는 경우들에 있어서, 에러에 대한 보수(compliment)가 혼동을 방지하기 위해 전달된다.

일 실시예에서, 로버스트(robust)한 에러 경고 및 탐지 시스템을 제공하기 위해서, 에러 코드는 전달되는 일련의 패킷들, 일반적으로 전달되는 모든 패킷들을 이용하여 여러 번 전달될 수 있거나 또는 에러가 탐지된 후에 전송될 수 있다. 이것은 에러를 생성하는 조건이 시스템에서 클리어되는 시점까지 발생하게 되며, 상기 시점에서 일반적인 CRC 비트들이 다른 값에 의해 오버로딩되지 않고 전달된다.

CRC 값을 오버로딩하는 이러한 기법은 최소량의 잔여 비트들 또는 필드들을 사용하여 시스템 에러들에 대한 보다 빠른 응답을 제공한다.

도 66에 도시된 바와 같이, 이전에 설명되거나 또는 알려진 다른 회로의 일부를 형성할 수 있는, 에러 탐지기 또는 탐지 수단(6602)을 사용하는 CRC 겹쳐 쓰기 메커니즘 또는 장치(6600)가 도시되어 있으며, 통신 링크 또는 프로세스 내에 존재하는 에러들을 검출한다. 다른 회로의 일부로서 형성될 수 있거나 또는 미리-선택된 에러 메시지들을 저장하기 위한 룩업(look up) 테이블들과 같은 기법들을 사용할 수 있는, 에러 코드 생성기 또는 수단(6604)은 발생하여 검출되었던 특정한 미리 정의된 에러들 또는 홈결들을 표시하기 위해 하나 이상의 에러 코드들을 생성한다. 장치들(6602 및 6604)은 원하는 바에 따라 하나의 회로 또는 장치로 형성될 수 있으며, 또한 다른 알려진 프로세서들 및 엘리먼트들에 대한 프로그래밍된 단계들의 시퀀스의 일부로서 형성될 수 있다는 것을 이해해야 할 것이다.

선택된 에러 코드 또는 코드들이 전달된 CRC 값과 동일한지 여부를 알기 위한 검사를 위한 CRC 값 비교기 또는 비교 수단(6606)이 도시되어 있다. 이러한 경우에, 코드 보수 생성기 또는 생성 수단 또는 장치는 원시 CRC 패턴 또는 값처럼 실수가 없고 탐지 방식이 혼동을 주거나 복잡하지 않도록 에러 코드들의 보수를 제공하기 위해 사용된다. 에러 코드 선택기 또는 선택 수단 엘리먼트 또는 장치(6610)는 삽입 또는 겹쳐 쓰기 위해 필요한 에러 코드 또는 값을 선택하거나, 또는 적절하게 각각의 보수들을 선택한다. 에러 코드 CRC 겹쳐 쓰기 장치 또는 겹쳐 쓰기 메커니즘 또는 수단(6612)은 원하는 에러 코드들을 수신 장치로 전달하기 위해, 데이터 스트림, 패킷들 및 삽입될 원하는 코드들을 수신하고 대응하거나 또는 적절한 CRC 값들을 겹쳐 쓰는 장치이다.

언급된 바와 같이, 에러 코드는 일련의 패킷들을 사용하여 여러 번 전달될 수 있으며, 겹쳐 쓰기 장치(6612)는 프로세싱 동안에 코드들의 복사본들을 유지하기 위해 메모리 저장 엘리먼트들을 사용할 수 있거나 또는 필요에 따라 자신들의 값들을 저장 또는 유지하기 위해 사용될 수 있는 이전의 엘리먼트들 또는 다른 알려진 저장 위치들로부터 상기 코드들을 호출할 수 있다.

도 66의 겹쳐 쓰기 메커니즘이 구현하는 일반적인 프로세싱은 도 67a 및 67b에 추가적으로 상세하게 도시되어 있다. 도 67a에서, 하나 이상의 에러는 단계(6702)에서 통신 데이터 또는 프로세스에서 탐지되며, 에러 코드는 상기 조건을 표시하기 위해 단계(6704)에서 선택된다. 동시에 또는 적절한 시점에서, 교체될 CRC 값은 단계(6706)에서 검사되고, 단계(6708)에서 원하는 에러 코드와 비교된다. 이전에 논의된 바와 같이, 이러한 비교의 결과로서 원하는 코드 또는 다른 대표값들이 존재하는 CRC 값과 동일한지 여부가 결정된다. 이러한 경우에, 프로세싱은 단계(6712)로 진행하며, 여기서 보수 또는 몇몇 경우들에서 원하는 바에 따라 다른 대표값들이 삽입될 코드로서 선택된다. 단계들(6710 및 6714)에서 에러 코드들 또는 값들이 삽입되도록 결정되면, 적절한 코드가 삽입을 위해 선택된다. 이러한 단계들은 명확화를 위해 개별적으로 도시되어 있으나 일반적으로 단계(6708)의 결정에 대한 출력에 기반하는 하나의 선택을 나타낸다. 최종적으로, 단계(6716)에서, 적절한 값들이 프로세스의 타겟이 되는 패킷들과 함께 전달하기 위한 CRC 위치로 겹쳐 쓰이게 된다.

도 67b에 도시된 바와 같이, 패킷 수신 측에서, 패킷 CRC 값들은 단계(6722)에서 모니터링된다. 일반적으로, 데이터 전달 과정에서 에러가 발생하였는지 여부와 패킷 또는 패킷들의 재전송을 요청할 것인지 또는 추가적인 동작들을 금지할 것인지 여부 등을 결정하기 위해(이들 중 몇몇은 위에서 논의됨), CRC 값들은 시스템 내에서 하나 이상의 프로세스들에 의해 모니터링된다. 이러한 모니터링의 일부로서 이러한 정보는 또한 알려지거나 또는 미리 선택된 에러 코드들 또는 대표값들을 CRC 값들과 비교하기 위해 사용될 수 있다. 대안적으로, 개별적인 에러 탐지 프로세스 또는 모니터가 구현될 수 있다. 코드가 나타나면, 코드는 추출되거나 또는 그렇지 않으면 추가적인 프로세싱을 위해 단계(6724)에서 표시될 수 있다. 이것이 실제 코드 또는 보수인지 여부에 대한 결정이 단계(6726)에서 이루어질 수 있으며, 이러한 경우에 추가적인 단계(6728)가 상기 값을 원하는 코드값으로 바꾸기 위해 이용된다. 다른 경우에, 단계(6730)에서 결과적으로 추출된 코드, 보수 또는 다른 복원된 값들은 어떤 에러가 발생하였는지를 탐지하고 전송된 코드를 형성하기 위해 사용된다.

V. 링크 하이베이션(hibernation)

MDDI 링크는 하이베이션 상태에 신속하게 진입하고 하이베이션으로부터 신속하게 웨이크업할 수 있다. 상기와 같은 반응들은 통신 시스템 또는 장치가 MDDI 링크가 전력 소모를 감소시키기 위해 자주 하이베이션이 되도록 하며, 이는 사용을 위해 매우 신속하게 다시 웨이크업할 수 있기 때문이다. 일 실시예에서, 외부 모드로서, 클라이언트는 MDDI_Stb 쌍이 500kHz 레이트로 토글해야 하는 1`Mbps 레이트와 일치하는 스트로브 펄스 타이밍을 가지고 일정 데이터 레이트로 수행하는 최초 시간 동안 하이베이션으로부터 웨이크업한다. 클라이언트의 특성들이 호스트에 의해 발견되거나 호스트로

통신되면, 호스트는 1Mbps 부터 클라이언트가 동작할 수 있는 최대 레이트까지의 임의의 레이트로 링크를 웨이크업할 수 있다. 내부 모드에서 클라이언트들은 호스트와 클라이언트 모두가 동작할 수 있는 임의의 레이트로 웨이크업 할 수 있다. 이는 일반적으로 내부 모드에서 클라이언트가 웨이크업하는 최초 시간에 적용가능하다.

일 실시예에서, 링크가 하이버네이션으로부터 웨이크업하면 호스트 및 클라이언트는 펄스들의 시퀀스를 교환한다. 상기 펄스들은 약간의 전류만을 소비하는저속 라인 수신기들을 사용하여 최대 링크 동작 속도로 신호들을 수신하도록 요구되는 서로다른 수신기들로 검출된다. 호스트 또는 클라이언트는 링크를 웨이크업할 수 있으며, 따라서 웨이크업 프로토콜은 호스트와 클라이언트 모두가 동시에 웨이크업을 시도하는 경우에 발생할 수 있는 가능한 경쟁을 처리하도록 설계된다.

하이버네이션 상태 동안 MDDL_Data 및 MDDL_Stb의 서로다른 드라이버들은 디세이블되고, 모든 서로다른 쌍들을 통한 서로다른 전압은 0볼트이다. 하이버네이션으로부터 웨이크업하는 동안 펄스들의 시퀀스를 검출하는데 사용되는 서로다른 라인 수신기들은 계획적인 전압 오프셋을 갖는다. 일 실시예에서, 상기 수신기들에서 로직-1 및 로직-0 레벨 사이의 임계치는 대략 125mV이다. 이는 구동되지 않은 서로다른 쌍이 링크 웨이크업 시퀀스 동안 로직-0 레벨로 보여지도록 한다.

하이버네이션 상태로 진입하기 위해, 호스트는 링크 셋다운 패킷의 CRC 이후에 64 MDDL_Stb 사이클들을 전송한다. 호스트는 CRC 이후에 16 내지 56 MDDL_Stb 사이클들의 범위에서(출력 디세이블 전파 지연들을 포함하여) 호스트의 MDDL_Data0 출력을 디세이블한다. 호스트는 링크 셋다운 패킷이 웨이크업 시퀀스를 초기화하기 전에 링크 셋다운 패킷의 CRC 이후에 64 MDDL_Stb 사이클들을 전송하는 것을 종료한다. 일 실시예에서, 호스트-초기화된 웨이크업은 MDDL_Stb에 펄스들을 구동하기 전에 MDDL_Data0가 유효 로직-1 레벨에 도달한 후에 호스트가 적어도 100nsec 대기해야 하는 것으로 정의된다.

하이버네이션 상태에서부터 "웨이크업"하기 위해, 몇몇 동작들 및 프로세스들이 필요하다. 클라이언트, 여기에서 디스플레이가 호스트로부터 데이터 또는 통신, 서비스를 요구할 때, MDDL_Stb가 비활성인 동안 약 70 내지 1000 μ s 동안 MDDL_Data0 라인을 로직-1 상태로 구동하고 MDDL_Stb가 활성이 된 후에 약 70 MDDL_Dtb 사이클들 동안(60 내지 80의 범위에 걸쳐서) MDDL_Data0을 로직-1 레벨로 유도하지만, 요구되는 바에 따라 다른 주기들이 사용될 수 있다. 클라이언트는 그후에 MDDL_Data0 드라이버를 고임피던스 상태가 되도록 함으로써 상기 드라이버를 디세이블한다.

만약 MDDL_Stb가 하이버네이션 동안 활성이지만, 불가능하다면, 클라이언트는 MDDL_Data0을 약 70 MDDL_Stb 사이클들 동안(60 내지 80의 범위에 걸쳐) 로직-1 상태로 구동할 수 있다. 상기와 같은 동작은 순방향 링크(208)를 통해 데이터 트래픽을 시작 또는 재시작하며 그 상태동안 클라이언트를 폴링한다.

호스트는 요청 펄스의 존재를 검출해야 하며, 먼저 적어도 약 100nsec 동안 MDDL_Stb를 로직-0 레벨로 구동하고 MDDL_Data0을 로직-하이 레벨로 구동하는 스타트업 시퀀스를 시작한다. 그후에 토글링 동안 MDDL_Stb는 약 150MDDL_Stb 사이클들 동안(140 내지 160의 범위) MDDL_Data0을 로직-1 레벨로 구동하고 약 50MDDL_Stb 사이클들 동안 로직-0로 구동하는 것을 계속한다. 클라이언트는 80 이상의 MDDL_Stb 사이클들 동안 로직-1 상태에서 MDDL_Data0을 검출하는 경우에 서비스 요청 펄스를 전송하지 않아야 한다. 클라이언트가 60 내지 80 MDDL_Stb 사이클들 동안 로직-1 레벨에서 MDDL_Data0을 검출하면 호스트가 50 MDDL_Stb 사이클들 동안 MDDL_Data0을 로직-0 레벨로 구동하는 간격을 탐색하기 시작한다. 호스트가 50 MDDL_Stb 사이클들 동안 MDDL_Data0을 로직-0 레벨로 구동한 후에, 호스트는 링크를 통해 패킷들을 전송하기 시작한다. 전송된 제 1 패킷은 서브-프레임 헤더 패킷이다. 클라이언트는 MDDL_Data0가 50 사이클 간격의 40 MDDL_Stb 사이클들 동안 로직-0 레벨인 이후에 서브-프레임 헤더 패킷을 찾기 시작한다. 하이버네이션 프로세싱 및 스타트업 시퀀스와 연관된 시간 선택 특성 및 시간 간격들의 오차는 하기에서 논의된다. (하기의 도 68A-C 참조.)

호스트는 먼저 MDDL_Stb를 인에이블함으로써 웨이크업을 초기화하고 동시에 이를 로직-0 레벨로 구동할 할 수 있다. MDDL_Stb는 하기에 설명되는 것과 같이 펄스들이 출력될 때까지는 로직-1 레벨로 구동되지 않아야 한다. MDDL_Stb가 로직-0 레벨에 도달한 후에 호스트는 MDDL_Data0을 인에이블하고 동시에 이를 로직-1 레벨로 구동한다. MDDL_Data0은 하기에 설명되는 것과 같이 50 MDDL_Stb 펄스들의 간격 동안 로직-0 레벨로 구동되는 간격 RK지 로직-0 레벨로 구동되지 않아야 한다. 호스트는 MDDL_Data0이 MDDL_Stb에 펄스들을 구동하기 전에 유효한 로직-1 레벨에 도달할 때까지 적어도 100nsec 대기해야 한다. 상기 타이밍 관계는 최악의 출력 인에이블 지연들을 고려하여 발생한다. 이는 실질적으로 클라이언트가 호스트에 의해 구동된 MDDL_Data0에서 로직-1 레벨에 의해 호출된 후에 MDDL_Stb 수신기를 완전히 인에이블하기에 충분한 시간을 가지는 것을 보장한다.

경합이 없는 일반적인 클라이언트 서비스 요청 이벤트(3800)에 대한 처리 단계들의 예가 도 38에 도시되며, 상기 이벤트들은 대문자 A, B, C, D, E, F 및 G를 사용하여 설명하기에 편리하게 표시된다. 프로세스는 포인트 A에서 시작하며, 호스

트는 클라이언트 장치에 링크가 저전력 절전 모드 상태로 변화할 것이라고 통지하는 링크 쉷다운 패킷을 전송한다. 다음 단계에서, 포인트 B에 도시된 것과 같이 호스트는 MDDI_Data0 드라이버를 디세이블하고 MDDI_Stb 드라이버가 로직 0 이 되도록 함으로써 저전력 절전 모드 상태로 진입한다. MDDI_Data0는 고 임피던스의 바이어스 네트워크에 의해 로직 0 레벨이 된다. 임의의 주기들 이후에, 클라이언트는 포인트 C에 도시된 것과 같이 MDDI_Data0가 로직 1 레벨이 되도록 함으로써 호스트에 서비스 요청 펄스를 전송한다. 호스트는 고 임피던스의 바이어스 네트워크를 사용할 때 로직 0 레벨을 주장하지만, 클라이언트 내의 드라이버는 라인이 로직 1 레벨이 되도록 한다. 50 μ s 내에, 호스트는 서비스 요청 펄스를 인식하고, 포인트 D에 도시된 것과 같이 드라이버를 인에이블함으로써 MDDI_Data0에서 로직 1 레벨을 주장한다. 그 후에 클라이언트는 서비스 요청 펄스를 주장하는 것을 시도하는 것을 중단하고, 포인트 E에 도시된 것과 같이 드라이버가 고-임피던스 상태가 되도록 한다. 호스트는 포인트 F에 도시된 것과 같이 MDDI_Data0이 50 μ s 동안 로직 0 레벨이 되도록 하고, MDDI_Data0에서 로직 0 레벨과 일치하는 방식으로 MDDI_Stb를 생성하는 것을 시작한다. 클라이언트는 MDDI_Data0이 40 MDDI_Stb 사이클들동안 로직-0 레벨인 후에 서브-프레임 헤더 패킷을 탐색하기 시작한다. MDDI_Data0가 로직 0 레벨이라 주장하고 MDDI_Stb를 50 μ s 동안 구동한 후에, 호스트는 포인트 G에 도시된 것과 같이 서브 프레임 헤더 패킷을 전송함으로써 순방향 링크를 통해 데이터를 전송하기 시작한다.

유사한 예가 도 39에 도시되며, 서비스 요청은 링크 재시작 시퀀스가 시작한 후에 주장되고 이벤트들은 대문자 A, B, C, D, E, F 및 G를 사용하여 다시 표시된다. 이는 클라이언트로부터의 요청 펄스 또는 신호가 서브 프레임 헤더 패킷을 변조시키도록 한다. 프로세스는 포인트 A에서 시작하여 호스트는 클라이언트 장치에 링크 쉷다운 패킷을 전송하여 링크가 저전력 절전 모드 상태로 변화할 것임을 통지한다. 다음 단계에서, 호스트는 포인트 B에 도시된 것과 같이 MDDI_Data0 드라이버를 디세이블하고 MDDI_Stb 드라이버를 로직 0 레벨로 세팅함으로써 저전력 절전 모드 상태가 된다. 전과 같이, MDDI_Data0는 고-임피던스 바이어스 네트워크에 의해 로직 0 레벨이 된다. 일정 기간 이후에, 호스트는 포인트 C에 도시된 것과 같이 150 μ s 동안 MDDI_Data0를 로직 1 레벨로 구동하여 링크 재시작 시퀀스를 시작한다. 링크 재시작 시퀀스가 시작한 이후 50 μ s가 경과되기 전에, 디스플레이는 포인트 D에 도시된 것과 같이 70 μ s의 지속 기간 동안 MDDI_Data0를 주장한다. 이는 디스플레이가 호스트로부터 서비스를 요청해야 하고 호스트가 이미 링크 재시작 시퀀스를 시작하였음을 인식하지 못하기 때문에 발생한다. 클라이언트는 그 후에 서비스 요청 펄스를 주장하는 것을 시도하는 것을 중단하고 포인트 E에 도시된 것과 같이 드라이버가 고-임피던스 상태가 되도록 한다. 호스트는 MDDI_Data0를 로직 1 레벨로 구동하는 것을 계속한다. 호스트는 포인트 F에 도시된 것과 같이 MDDI_Data0를 50 μ s 동안 로직 0 레벨로 구동하며, MDDI_Data0에서 로직 0 레벨과 일치하는 방식으로 MDDI_Stb를 생성하는 것을 시작한다. MDDI_Data0를 로직 0 레벨로 주장하고 MDDI_Stb를 50 μ s 동안 구동한 후에, 호스트는 포인트 G에 도시된 것과 같이 서브 프레임 헤더 패킷을 전송함으로써 순방향 링크를 통해 데이터를 전송하는 것을 시작한다.

상기 설명으로부터, 종래의 해결책은 웨이크업 시퀀스의 일부로서 호스트가 2가지 상태가 되도록 하는 것과 관련됨을 알게 된다. 제 1 상태에 대하여, 호스트는 MDDI_Data0를 150 μ s 동안 하이(high)로 구동하고, MDDI_Stb 라인을 활성화하는 동안 MDDI_Data0 신호를 50 μ s 동안 로우(low)로 구동하며, MDDI 패킷들을 전송하기 시작한다. 상기 프로세스는 MDDI 장치 및 방법들을 사용하여 달성할 수 있는 데이터 전송률과 관련하여 당업계 수준을 진보시킨다. 그러나 전술된 것과 같이, 다음 단계 또는 프로세스를 조절하거나 더 빠르게 선택할 수 있도록 하기 위한 감소된 응답시간과 관련하여 속도가 빨라지는 것은 프로세싱 또는 엘리먼트들을 간략화하는 능력이며, 항상 요구되는 사항이다.

출원인들은 호스트가 신호 토글링을 위해 클록 사이클 기반의 타이밍을 사용하는 웨이크업 프로세싱 및 타이밍에 대한 새로운 접근 방법을 발견하였다. 상기 구성에서, 호스트가 웨이크업 시퀀스의 시작시 MDDI_Data0 신호를 하이로 구동한 후에 호스트는 0 내지 10 μ s까지 MDDI_Stb를 토글링하는 것을 시작하며, 신호가 로우로 구동될 때까지 대기하지 않는다. 웨이크업 시퀀스 동안, 호스트는 MDDI_Data0 신호가 항상 로직 0 레벨이었던 것처럼 MDDI_Stb를 토글링한다. 이는 클라이언트 측으로부터의 시간 개념을 효과적으로 제거하며, 호스트는 상기 주기들에 대하여 최초 2개의 상태들에 대한 이전의 150 μ s 및 50 μ s 기간으로부터 150 클록 사이클 및 50 클록 사이클들로 변화한다.

호스트는 현재 데이터 라인을 하이로 구동하고 데이터 라인이 0인 경우에 10 클록 사이클 내에 스트로브 신호를 전송하기 시작해야 한다. 호스트가 150 클록 사이클들 동안 데이터 라인을 하이로 구동한 후에, 호스트는 스트로브 신호를 전송하는 것을 계속하면서 50 클록 사이클 동안 데이터 라인을 로우로 구동한다. 상기 프로세스들 모두가 종료된 후에, 호스트는 제 1 서브 프레임 헤더 패킷을 전송하기 시작할 수 있다.

클라이언트 측에서, 클라이언트 실행은 현재 데이터 라인이 최초 하이이고 그 후에 로우인 클록 사이클들의 개수를 계산하기 위해 발생한 클록을 사용할 수 있다. 하이 상태로 구동된 모든 데이터 라인에서 발생해야 하는 클록 사이클들의 개수는 150이며, 로우 상태로 구동된 데이터 라인에서 발생해야 하는 클록 사이클들의 개수는 50이다. 이는 적절한 웨이크업 시퀀스에 대하여, 클라이언트가 하이인 데이터 라인의 적어도 150개의 연속적인 클록 사이클들을 카운트한 후 로우인 데이터 라인의 적어도 50개의 연속적인 클록 사이클들을 카운트할 수 있어야 함을 의미한다. 상기 두 조건들 모두가 만족하면,

클라이언트는 제 1 서브 프레임의 유일한 단어를 탐색하는 것을 시작할 수 있다. 상기 패턴에서의 중단은 클라이언트가 하이인 데이터 라인의 최초 150개의 연속하는 클록 사이클들을 검색하는 초기 상태로 카운트를 복귀시키는 기준으로서 사용된다.

절전 모드로부터의 호스트 기준 웨이크업을 위한 본 발명의 클라이언트 실행은 전술된 것과 같이 클록 레이트가 1 Mbps에서 시작하지 않아야 한다는 점을 제외하고 초기 시작(start-up) 케이스와 매우 유사하다. 통신 링크가 절전 모드가 되었을 때 이전 레이트가 활성화되는 것은 무엇이든지 다시 시작하도록 세팅될 수 있다. 만약 호스트가 전술된 것과 같은 스트로브 신호의 전송을 시작하면, 클라이언트는 하이인 데이터 라인의 적어도 150개의 연속하는 클록 사이클들을 카운팅한 후에 로우인 데이터 라인의 적어도 50개의 연속하는 클록 사이클들을 카운팅할 수 있어야 한다. 상기 두 조건들 모두가 만족하면, 클라이언트는 고유 워드에 대한 탐색을 시작할 수 있다.

절전 모드로부터의 클라이언트 기반 웨이크업을 위한 본 발명의 클라이언트 실행은 클라이언트가 데이터 라인을 구동시킴으로써 시작되는 것을 제외하고는 호스트 기반의 웨이크업과 유사하다. 클라이언트는 호스트 장치를 웨이크업 하기 위해 클록 없이 데이터 라인을 비동기로 구동시킬 수 있다. 호스트가 데이터 라인이 클라이언트에 의해 하이로 구동된다고 인식하면, 웨이크업 시퀀스를 시작할 수 있다. 클라이언트는 호스트 시작에 의해 또는 웨이크업 프로세스 동안 발생한 클록 사이클들의 개수를 카운트할 수 있다. 클라이언트가 하이인 데이터의 70개의 연속하는 클록 사이클들을 카운트하면, 데이터 라인을 하이로 구동하는 것을 중단할 수 있다. 상기 포인트에서, 호스트는 이미 데이터 라인을 하이로 적절히 구동시킬 수 있어야 한다. 클라이언트는 그 후에 하이인 데이터 라인의 또 다른 80개의 클록 사이클들을 카운트하여 하이인 데이터 라인의 150개 클록 사이클들에 도달하며, 로우인 데이터 라인의 50개 클록 사이클들도 검색할 수 있다. 이들 3가지 조건들이 만족하면, 클라이언트는 고유 워드를 검색하기 시작할 수 있다.

상기 새로운 웨이크업 프로세스의 실행의 장점은 시간 측정 장치가 필요하지 않다는 것이다. 상기 장치가 오실레이터인지 또는 커패시터 방전 회로인지, 아니면 다른 공지된 장치들인지 간에, 클라이언트는 스타트 업 조건들을 결정하기 위해 상기 외부 장치들을 더 이상 필요로 하지 않는다. 클라이언트 장치 보드에 제어기들, 카운터들, 등등을 구현할 때 비용과 회로 영역을 절약한다. 이는 클라이언트에게 유리한 것처럼 호스트에 대해 유리할 수 없으며, 상기 기술은 코어 회로를 위해 사용되는 매우 높은 밀도의 로직(VHDL)과 관련하여 호스트를 잠정적으로 간략화해야 한다. 웨이크업 통지 및 측정 소스로서 데이터 및 스트로브 라인들을 사용하는 전력 소비는 어떤 외부 회로도 코어 엘리먼트들이 호스트 기반 웨이크업을 위해 대기해야 할 필요가 없기 때문에 낮아질 것이다. 사용된 사이클들 또는 클록 주기들의 개수는 예시적이며 다른 주기들이 사용될 수 있음이 당업자에게 인식될 것이다.

상기 새로운 웨이크업 프로세스의 실행시 장점은 시간 측정 장치를 필요로 하지 않는 것이다. 상기 장치가 오실레이터인지 또는 커패시터 방전 회로인지, 아니면 다른 공지된 장치들인지 간에, 클라이언트는 스타트 업 조건들을 결정하기 위해 상기 외부 장치들을 더 이상 필요로 하지 않는다. 클라이언트 장치 보드에 제어기들, 카운터들, 등등을 구현할 때 비용과 회로 영역을 절약한다. 이는 클라이언트에게 유리한 것처럼 호스트에 대해 유리할 수 없으며, 상기 기술은 코어 회로를 위해 사용되는 매우 높은 밀도의 로직(VHDL)과 관련하여 호스트를 잠정적으로 간략화해야 한다. 웨이크업 통지 및 측정 소스로서 데이터 및 스트로브 라인들을 사용하는 전력 소비는 어떤 외부 회로도 코어 엘리먼트들이 호스트 기반 웨이크업을 위해 대기해야 할 필요가 없기 때문에 낮아질 것이다.

상기 새로운 기술의 동작을 간략하게 설명하기 위해, MDDI_Data0 및 MDDI_Stb의 타이밍 및 클록 사이클들과 관련된 다양한 동작들은 도 68a, 68b 및 68c에 도시된다.

경합이 없는 일반적인 호스트-개시 웨이크업을 위한 처리 단계들의 예는 도 68a에 도시되며, 이벤트들은 대문자들 A, B, C, D, E, F, 및 G를 사용하여 편리하게 표시된다. 프로세스는 포인트 A에서 시작하며, 호스트는 링크가 저전력 절전 모드 상태로 변화할 것이라고 통지하는 링크 셧다운 패킷을 클라이언트 장치로 전송한다. 다음 단계의 포인트 B에서, 호스트는 클라이언트에 의한 프로세싱이 MDDI_Stb가 토글링하는 것을 중단하기 전에 완료되도록 하며, 이는 클라이언트 장치 내에서 클록이 복원되는 것을 중단시킨다. 호스트는 또한 먼저 MDDI_Data0이 로직 0 레벨로 세팅하고 CRC 이후에 16 내지 48 사이클 범위내에(일반적으로 출력 디세이블 전파 지연들을 포함하는) MDDI_Data0 출력을 디세이블한다. CRC 이후 48 사이클들 이후 및 다음 단계 C 이전의 얼마 동안 클라이언트 내의 MDDI_Data0 및 MDDI_Stb를 위한 고속 수신기들이 저전력 상태가 되도록 하는 것이 바람직할 수 있다. 클라이언트는 MDDI_Data0 및 MDDI_Stb를 위한 고속 수신기들이 링크 셧다운 패킷의 CRC 이후에 48번째 MDDI_Stb 사이클의 상승 에지 이후 임의의 시간에서 하이버네이션이 되도록 한다. 클라이언트가 MDDI_Data0 및 MDDI_Stb를 위한 고속 수신기들이 링크 셧다운 패킷의 CRC 이후에 64번째 MDDI_Stb 사이클의 상승 에지 이후 임의의 시간에서 하이버네이션이 되도록 하는 것이 제안된다.

호스트는 MDDI_Data0 및 MDDI_Stb 드라이버들을 디세이블하고 호스트 제어기를 저전력 절전 모드 상태가 되도록 함으로써 포인트 또는 단계 C에서 저전력 절전 모드 상태가 된다. 또한, MDDI_Stb 드라이버를 로직 0 레벨로 세팅하거나(고 임피던스 바이어스 네트워크를 사용하여) 원하는 경우에 절전 모드 동안 토글링을 계속한다. 클라이언트는 또한 저전력 레벨의 절전 모드 상태가 된다.

임의의 기간 이후에, 호스트는 MDDI_Data0 및 MDDI_Stb 드라이버 출력을 인에이블함으로써 포인트 D에서 링크 재시작 시퀀스를 시작한다. 호스트는 드라이버들이 그들의 개별 출력들을 완전히 인에이블해야 하는 동안 MDDI_Data0를 로직 1 레벨이 되도록 하고 MDDI_Stb를 로직 0 레벨이 되도록 한다. 호스트는 일반적으로 펄스들을 MDDI_Stb에서 구동하기 전에 상기 출력들이 원하는 로직 레벨들에 도달한 이후 약 200나노초 동안 대기한다. 이는 클라이언트 시간이 수신을 준비하도록 한다.

호스트 드라이버들이 인에이블되고 MDDI_Data0가 로직 1 레벨이 되면, 호스트는 포인트 E에 도시된 것과 같이 150MDDI_Stb 사이클들의 지속기간 동안 MDDI_Stb를 토글링하기 시작한다. 호스트는 포인트 F에 도시된 것과 같이 MDDI_Data0가 50 사이클들 동안 로직 0 레벨이 되도록 하고 클라이언트는 MDDI_Data0이 40 MDDI_Stb 사이클들 동안 로직 0 레벨인 이후에 서브 프레임 헤더 패킷을 검색하는 것을 시작한다. 호스트는 포인트 G에 도시된 것과 같이 서브 프레임 헤더 패킷을 전송함으로써 순방향 링크를 통해 데이터를 전송하기 시작한다.

경합이 없는 일반적인 클라이언트-개시 웨이크업에 대한 처리 단계들의 예가 도 68b에 도시되며, 이벤트들은 편리한 설명을 위해 대문자들 A, B, C, D, E, F, G, H 및 I를 사용하여 표시된다. 전송된 것과 같이, 프로세스는 포인트 A에서 시작하며, 호스트는 링크가 저전력 상태로 변화할 것이라고 통지하는 링크 섯다운 패킷을 클라이언트에 전송한다.

포인트 B에서, 호스트는 클라이언트에 의한 프로세싱이 MDDI_Stb가 토글링하는 것을 중단하고, 클라이언트 장치 내에서 클럭이 복원되는 것을 중단하기 전에 완료되도록 하기 위해 MDDI_Stb를 약 64 사이클들 동안(또는 시스템 설계에 의해 원하는 바에 따라) 토글링한다. 호스트는 또한 먼저 MDDI_Data0를 로직 0 레벨로 세팅하고 CRC 이후에 16 내지 48 사이클 범위 내에(일반적으로 출력 디세이블 전과 지연들을 포함하는) MDDI_Data0 출력을 디세이블한다. CRC 이후 및 다음 단계 C 이전의 48 사이클들 이후의 얼마 동안 클라이언트 내의 MDDI_Data0 및 MDDI_Stb를 위한 고속 수신기들이 저전력 상태가 되도록 하는 것이 바람직할 수 있다.

호스트는 MDDI_Data0 및 MDDI_Stb 드라이버들을 디세이블하고 호스트 제어기를 저전력 절전 모드 상태가 되도록 함으로써 포인트 또는 단계 C에서 저전력 절전 모드 상태가 된다. 또한, MDDI_Stb 드라이버를 로직 0 레벨로 세팅하거나(고 임피던스 바이어스 네트워크를 사용하여) 원하는 경우에 절전 모드 동안 토글링을 계속한다. 클라이언트는 또한 저전력 레벨의 절전 모드 상태가 된다.

임의의 기간 이후에, 호스트는 MDDI_Stb 수신기를 인에이블하고, 호스트가 MDDI_Stb 드라이버를 인에이블하기 전에 MDDI_Stb의 수신 버전의 상태가 클라이언트 내에서 로직 0 레벨임을 보장하도록 MDDI_Stb 수신기 내의 오프셋을 인에이블함으로써 포인트 D에서 링크 재시작 시퀀스를 시작한다. 요구되는 것과 같이, 유효한 차동 신호의 수신을 보장하고 여러 신호를 금지하기 위해 수신기를 인에이블하기 전으로 클라이언트가 오프셋을 인에이블하는 것이 바람직할 수 있다. 클라이언트는 MDDI_Data0를 로직 1 레벨이 되도록 하는 동안 MDDI_Data0 드라이버를 인에이블한다. 클라이언트는 MDDI_Data0 라인을 로직 1 레벨로 구동하면서 MDDI_Data0드라이버를 인에이블한다. 오프셋 및 표준 MDDI_Stb 차동 수신기를 인에이블하는 시간이 100nsec 미만인 경우에 MDDI_Data0 및 MDDI_Stb가 동시에 인에이블되는 것이 허용된다.

약 1 msec 이내에 포인트 E에서, 호스트는 클라이언트로부터의 서비스 요청 펄스를 인식하고, 호스트는 MDDI_Data0 및 MDDI_Stb 드라이버 출력들을 인에이블함으로써 링크 재시작 시퀀스를 시작한다. 호스트는 드라이버들이 그들의 개별 출력들을 인에이블해야하는 동안에는 MDDI_Data0를 로직 1 레벨이 되도록 하고 MDDI_Stb를 로직 0 레벨이 되도록 한다. 호스트는 일반적으로 상기 출력들이 MDDI_Stb에서 펄스들을 구동하기 전에 원하는 로직 레벨에 도달한 후에 약 200 나노초를 대기한다. 이는 클라이언트 시간이 수신을 준비하도록 한다.

호스트 드라이버들이 인에이블되고 MDDI_Data0가 로직 1 레벨이 되면, 호스트는 포인트 F에 도시된 것과 같이 150 MDDI_Stb 사이클의 지속 기간 동안 MDDI_Stb에 펄스들을 출력하기 시작한다. 클라이언트가 MDDI_Stb에서의 제 1 펄스를 인식하면, MDDI_Stb 수신기에서 오프셋을 디세이블한다. 클라이언트는 70 MDDI_Stb 사이클들 동안 MDDI_Data0

를 로직 1 레벨로 구동하는 것을 계속하고, 포인트 G에서 MDDI_Data0 드라이버를 디세이블한다. 호스트는 추가의 80 MDDI_Stb 펄스들의 기간 동안 MDDI_Data0을 로직-1로 구동하고 포인트 H에서 MDDI_Data0을 로직-0 레벨로 구동한다.

포인트들 G 및 H에 도시된 것과 같이, 호스트는 50 사이클들 동안 MDDI_Data0을 로직 0 레벨로 구동하고 클라이언트는 MDDI_Data0이 40 MDDI_Stb 사이클 동안 로직 0 레벨인 이후에 서브 프레임 헤더 패킷을 검색하기 시작한다. 호스트는 포인트 I에 도시된 것과 같이 서브 프레임 헤더 패킷을 전송함으로써 순방향 링크를 통해 데이터를 전송하기 시작한다.

클라이언트로부터 경합이 있고 클라이언트가 링크를 웨이크업하기 원하는 일반적인 호스트-개시 웨이크업에 대한 처리 단계들의 예가 도 68c에 도시된다. 이벤트들은 편리한 설명을 위해 대문자들 A, B, C, D, E, F, G, H 및 I를 사용하여 표시된다. 전송된 것과 같이, 프로세스는 포인트 A에서 시작하며, 호스트는 링크가 저전력 상태로 변화할 것이라고 통지하는 링크 셧다운 패킷을 클라이언트에 전송하고, 포인트 B로 진행하며 MDDI_Stb는 클라이언트에 의한 프로세싱이 완료되도록 하기 위해 약 64 사이클들 동안(또는 시스템 설계에 의해 원하는 바에 따라) 토글링되며, 그 후에 포인트 C에서 호스트는 MDDI_Data0 및 MDDI_Stb를 디세이블하고 호스트 제어기가 저전력 절전 모드 상태가 되도록 함으로써 저전력 절전 모드 상태가 된다. 일정 기간 이후에, 호스트는 포인트 D에서 MDDI_Data0 및 MDDI_Stb 드라이버 출력을 인에이블함으로써 링크 재시작 시퀀스를 시작하고, 포인트 E에 도시되는 것과 같이 150MDDI_Stb 사이클의 지속 기간 동안 MDDI_Stb를 토글링하기 시작한다.

포인트 E 이후 70까지의 MDDI_Stb 사이클들, 포인트 F에서, 클라이언트는 아직 호스트가 MDDI_Data0을 로직 1 레벨로 구동하고 있음을 인식하지 못하며, 따라서, 클라이언트는 MDDI_Data0을 로직 1 레벨로 구동한다. 이와 같은 상황은 클라이언트가 서비스를 요청하기를 원하지만 통신하고자 하는 호스트가 링크 재시작 시퀀스를 이미 시작하였음을 인식하지 못하기 때문에 발생한다. 포인트 G에서, 클라이언트는 MDDI_Data0을 구동하는 것을 중단하고 출력을 디세이블함으로써 드라이버가 고 임피던스 상태가 되도록 한다. 호스트는 80개의 추가 사이클들 동안 MDDI_Data0을 로직 1 레벨로 구동하는 것을 계속한다.

호스트는 포인트 H에 도시된 것과 같이 50 사이클들 동안 MDDI_Data0을 로직 0 레벨로 구동하고, 클라이언트는 MDDI_Data0이 40 MDDI_Stb 사이클들 동안 로직 1 레벨인 이후에 서브 프레임 헤더 패킷을 검색하기 시작한다. 호스트는 포인트 I에 도시된 것과 같이 서브 프레임 헤더 패킷을 전송함으로써 순방향 링크를 통해 데이터를 전송하기 시작한다.

VI. 인터페이스의 전기적인 사양

예시적인 실시예들에서, 비제로 복귀(NRZ) 포맷의 데이터가 데이터-스트로브 신호 또는 DATA-STB 포맷을 사용하여 인코딩되며, 이는 클록 정보가 데이터 및 스트로브 신호들에 삽입되도록 한다. 클록은 복잡한 위상 고정 루프 회로 없이 복원될 수 있다. 데이터는 양방향 차동 링크를 통해 전달되며, 이는 일반적으로 유선 케이블을 사용하여 구현되지만 전송된 것과 같이 다른 전도체들, 인쇄된 와이어들, 또는 전송 엘리먼트들이 사용될 수 있다. 스트로브 신호(STB)는 호스트에 의해서만 구동되는 단방향 링크를 통해 전달된다. 스트로브 신호는 데이터 라인 또는 신호에서 동일하게 유지되는 백-투-백 상태, 또는 1 이 존재할 때마다 값(0 또는 1)을 토글링한다.

비트들 "1110001011"과 같은 데이터 시퀀스가 DATA-STB 인코딩을 사용하여 전송될 수 있는 방법의 일 예는 도 40에 도시된다. 도 40에서, 데이터 신호(4002)는 신호 타이밍 차트의 제 1 라인에 도시되고 STB 신호(4004)는 제 2 라인에 도시되며, 각각의 시간은 적절하게(공통 시작 포인트에서) 정렬된다. 시간이 경과함에 따라, DATA 라인(4002;신호)에서 상태의 변경이 발생하면, STB 라인(4004;신호)은 이전 상태를 유지하며, 따라서, DATA 신호의 제 1 '1'상태는 STB 신호의 시작값인 제 1 '0' 상태와 상관된다. 그러나 DATA 신호의 상태, 즉 레벨이 변경되지 않으면, 또는 변경되지 않을 때 STB 신호는 도 40에서 DATA가 또 다른 '1'값을 제공하는 경우와 같이 현재 예에서 상반된 상태 또는 '1'로 토글링한다. 즉, DATA 및 STB 사이에는 1 및 비트 사이클당 1번의 변화만이 존재한다. 따라서, STB 신호가 다시 변화하면, 즉 상기 시간을 '0'으로 변화시키면, 데이터 신호가 '1'을 유지하며 상기 레벨 또는 값을 유지하면 DATA 신호는 레벨을 '0'으로 변화시킨다. DATA 신호가 '1'을 유지하면, STB 신호는 상반된 상태 또는 '1'로 토글링하며, DATA 신호가 변화하거나 레벨들 또는 값들을 변화시키는 경우에도 상기와 같다.

상기 신호들을 수신하면, DATA 및 STB 신호들에 배타적 논리합(XOR) 연산이 수행되어 클록 신호(4006)를 발생하며, 이는 원하는 데이터 및 스트로브 신호들과의 상대적인 비교를 위한 타이밍 차트의 하부에 도시된다. 호스트에서 입력 데이터로부터의 DATA 및 STB 출력들 또는 신호들을 생성하고, 클라이언트에서 DATA 및 STB 신호들로부터 데이터를 복원 또는 재포착하기 위한 회로의 일 예가 도 41에 도시된다.

도 41에서, 송신부(4100)는 중간 신호 경로(4102)를 통해 원래의 DATA 및 STB 신호들을 발생하여 전송하는데 사용되며, 수신부(4120)는 상기 신호들을 수신하여 데이터를 복원하는데 사용된다. 도 41에 도시된 것과 같이, 호스트로부터 클라이언트로 데이터를 전송하기 위해, DATA 신호는 회로들을 트리거하기 위한 클록 신호와 함께 2개의 D형 플립 플롭 회로 엘리먼트(4104 및 4106)에 입력된다. 2개의 플립 플롭 회로 출력들(Q)은 각각 두개의 차동 라인 드라이버들(4108 및 4110; 전압 모드)을 사용하여 서로 다른 신호 쌍들 MDDI_Data0+, MDDI_Data0-, MDDI_Stb+, MDDI_Stb-로 분할된다. 3입력 배타적 NOR(XNOR) 게이트, 회로, 또는 로직 엘리먼트(4112)는 DATA 및 두 플립 플롭 모두의 출력들을 수신하기 위해 접속되며, 차례로 MDDI_Stb+, MDDI_Stb- 신호들을 생성하는 제 2 플립 플롭에 대한 데이터 입력을 제공하는 출력을 발생한다. 편리함을 위해, XNOR 게이트는 스트로브를 발생하는 플립 플롭의 Q 출력을 효율적으로 반전하고 있음을 나타내도록 배치된 반전 표시를 갖는다.

도 41의 수신부(4120)에서, MDDI_Data0+, MDDI_Data0-, MDDI_Stb+, MDDI_Stb- 신호들은 두 개의 차동 라인 수신기들(4122 및 4124)의 각각에 의해 수신되며, 차동 신호들로부터 단일 출력들을 발생한다. 증폭기들의 출력들은 클록 신호를 발생하는 2입력 배타적 OR(XOR) 게이트, 회로 또는 로직 엘리먼트(4126)의 각각의 입력들에 입력된다. 클록 신호는 지연 엘리먼트(4132)를 통해 데이터 신호의 지연 버전을 수신하는 2개의 D형 플립 플롭 회로들(4128 및 4130)의 각각을 트리거하는데 사용되며, 이중 하나(4128)는 데이터 '0' 값들을 발생하고 다른 하나(4130)는 데이터 '1' 값들을 발생한다. 클록은 XOR 로직으로부터 독립적인 출력을 갖는다. 클록 정보가 DATA 및 STB 라인들 사이에서 분산되기 때문에, 클록 레이트의 1/2 보다 빠른 상태를 사이에서 신호 변화는 존재하지 않는다. 클록은 DATA 및 STB 신호들의 배타적 OR 프로세싱을 사용하여 재발생되기 때문에, 시스템은 클록 신호가 단일의 지정된 데이터 라인을 통해 직접 전송되는 상황과 비교하여 입력 데이터 및 클록 사이의 큐의 양의 2배까지 허용한다.

MDDI Data 쌍들, MDDI_Stb+, MDDI_Stb- 신호들은 잡음의 부정적인 영향으로부터 면역성을 최대화하기 위해 서로 다른 모드에서 동작된다. 각각의 서로다른 쌍들은 신호들을 전송하는데 사용되는 케이블 또는 도전체의 특성 임피던스를 사용하여 동시에 종결된다. 일반적으로 모든 동시의 종결들은 클라이언트 내에서 발생된다. 이는 순방향 트래픽(호스트로부터 클라이언트로부터 전송된 데이터)을 위한 서로다른 수신기에 인접하지만 역방향 트래픽(클라이언트로부터 호스트로 전송된 데이터)을 위한 케이블 또는 다른 도전체들 또는 전송 엘리먼트들의 구동 단부에 존재한다. 역방향 트래픽에 대하여, 신호는 클라이언트에 의해 구동되고, 호스트에서 고임피던스 수신기에 의해 반사되며 클라이언트에서 종결된다. 이는 전류 소비를 증가시키는 이중 종결에 대한 필요성을 차단한다. 이는 케이블내의 라운드-트립 지연의 역수보다 큰 데이터 레이트로 기능한다. MDDI_Stb+, MDDI_Stb- 도전체들 또는 신호들은 호스트에 의해서만 구동된다.

본 발명의 MDD 인터페이스의 일부로서 신호들을 전송하기 위한 드라이버들, 수신기들 및 종단부들을 실행하기 위해 사용할 수 있는 엘리먼트들의 예시적인 구성은 도 42에 도시된다. 상기 예시적인 인터페이스는 1V 미만의 전력 스윙들 및 저전력 드레인을 가지고 낮은 감지 전압, 여기에서 200mV를 사용한다. 각각의 신호 쌍이 드라이버는 서로다른 전류 출력을 갖는다. MDDI 패킷들을 수신하는 동안, MDDI_Data 및 MDDI_Stb 쌍들은 0V의 전압 임계치를 가지는 종래의 서로다른 수신기를 사용한다. 하이버네이션 상태에서 드라이버 출력들은 디세이블되고 동시-종결 레지스터들은 각각의 신호 쌍에서 전압을 0V로 풀링한다. 하이버네이션 동안 MDDI_Data0 쌍에서 특정 수신기는 양의 125mV인 오프셋 입력 임계치를 가지며, 상기 값은 하이버네이션 라인 수신기가 구동되지 않은 신호쌍을 로직-0 레벨로 해석하도록 한다.

때때로 호스트 또는 클라이언트는 서로다른 쌍을 로직-0 레벨 또는 로직-1 레벨로 구동하여 데이터 흐름의 방향이 변화할 때(호스트로부터 클라이언트로 또는 클라이언트로부터 호스트로) 상기 쌍에서 유효한 로직-레벨을 보장한다. 출력 전압 범위 및 출력 규격들은 여전히 동일한 로직 레벨에서 동시에 구동된 출력들을 충족한다. 몇몇 시스템들에서 하이버네이션 동안 및 링크가 하이버네이션 상태에서부터 웨이크업하는 특정 시간에 적은 오프셋 전압을 생성하기 위해 적은 전류를 종결된 서로다른 쌍으로 인가하는 것이 필요할 수 있다. 상기와 같은 상황들에서, 인에이블된 오프셋-전류 바이어스 회로들은 하기와 같이 참조되는 전력 레벨들을 구동한다: $I_{\text{ESD-and-Rx}}$ - 일반적으로 $I_{\text{ESD-and-Rx}} \leq 1\mu\text{A}$ 인 내부 ESD 다이오드 및 서로다른 수신기 입력; $I_{\text{Tx-Hi-Z}}$ - 일반적으로 $I_{\text{Tx-Hi-Z}} \leq 1\mu\text{A}$ 인 고임피던스 상태에서 서로다른 드라이버 출력; 및 $I_{\text{external-ESD}}$ - 일반적으로 $I_{\text{external-ESD}} \leq 3\mu\text{A}$ 인 외부 ESD 방지 다이오드를 통한 누설 전류.

상기 누설 전류들의 각각이 도 101에 도시된다. 풀-업 및 풀-다운 회로들은 최악의 경우 전술된 누설 조건들이 동시에 발생하는 경우에 최소의 서로다른 전압을 달성해야 한다. 전체 누설은 외부 ESD 방지 다이오드들이 없는 내부 모드에 대하여 $\leq 4\mu\text{A}$ 이고 외부 ESD 방지 다이오드들이 있는 외부 모드에 대하여 $\leq 10\mu\text{A}$ 이다.

차동 라인 드라이버들 및 라인 수신기들의 전기적 파라미터들 및 특성들이 표 8에 도시된다. 기능적으로, 드라이버는 입력을 통해 로직 레벨을 양의 출력에 전송하고, 입력의 반전을 음의 출력에 전송한다. 입력으로부터 출력들로의 지연은 차동

적으로 구동되는 차동 라인에 우수하게 매칭된다. 대부분의 실행에서, 출력들에서의 전압 스윙은 전력 소비 및 전자기 방사를 최소화하기 위해 입력에서의 스윙 미만이 된다. 표 8은 최소 전압 스윙을 약 0.5V로 표시한다. 그러나 당업자에게 공지된 것과 같이 다른 값들이 사용될 수 있고, 본 발명자는 설계 제약들과 상관없이 몇몇 실시예들에서 더 작은 값을 고려할 수 있다.

차동 라인 수신기는 고속 전압 비교기와 동일한 특성을 갖는다. 도 41에서, 방울형 표시가 없는 입력은 양의 입력이고 방울형 표시가 있는 입력은 음의 입력이다. 출력은 $(V_{input+}) - (V_{input-})$ 이 0보다 큰 경우에 로직 1이다. 이를 설명하는 또 다른 방식은 로직 0 및 1의 전압 레벨들에 고정된 출력을 가지고 매우 큰(가상적으로는 무한한) 이득을 가지는 차동 증폭기이다.

서로 다른 쌍들 사이의 지연 스튜(skew)는 최고 전위 속도에서 서로 다른 전송 시스템을 동작하기 위해 최소화되어야 한다.

도 42에서, 호스트 제어기(4202) 및 클라이언트 또는 디스플레이 제어기(4204)는 통신 링크(4206)를 통해 패킷들을 전송하는 것으로 도시된다. 호스트 제어기는 전송될 호스트 DATA 및 STB 신호들을 수신할 뿐만 아니라 전송될 클라이언트 데이터 신호들을 수신하기 위해 3개의 드라이버들의 시리즈(4210, 4212, 4214)를 사용하며, 클라이언트는 3개의 드라이버들(4230, 4232, 4234)을 사용한다. 호스트 DATA(4212)를 통과시키는 드라이버는 일반적으로 호스트로부터 클라이언트로의 전송이 요구될 때만 통신 링크의 작동을 허용하도록 하기 위해 인에이블 신호 입력을 사용한다. STB 신호가 데이터 전송의 일부로서 형성되기 때문에, 드라이버(4212)를 위해 어떤 추가 인에이블 신호도 사용되지 않는다. 클라이언트 DATA 및 STB 수신기들(4132, 4230)의 각각의 입력들은 그들을 통해 이동되는 종단 임피던스들 또는 저항들(4218 및 4220)을 포함한다. 클라이언트 제어기내의 드라이버(4234)는 클라이언트로부터 호스트로 전송된 데이터 신호들을 준비하기 위해 사용되며, 상기 드라이버(4214)는 입력측에서 데이터를 처리한다.

특정 수신기(드라이버들; 4216 및 4236)은 DATA 라인들에 접속되며, 추후 논의되는 하이버네이션 제어의 일부로서 전송된 125mV 전압 오프셋을 생성하거나 사용한다. 오프셋들은 하이버네이션 라인 수신기들이 구동되지 않은 신호쌍들을 로직-0 레벨로 해석하도록 한다.

상기 드라이버들 및 임피던스들은 별개의 구성 요소들로서, 또는 회로 모듈의 일부로서, 또는 더 가격 효율적인 인코더 및 디코더 솔루션으로 작동하는 응용 주문형 집적 회로(ASIC)로서 형성될 수 있다.

전력이 한쌍의 도전체들을 통해 HOST_Pwr 및 HOST_Gnd라는 명칭의 신호들을 사용하여 호스트 장치로부터 클라이언트 장치 또는 디스플레이로 전송되는 것이 용이하게 보여질 수 있다. 신호의 HOST_Gnd 부분은 기준 접지 및 클라이언트 장치를 위한 전원 복귀 경로 또는 신호로 작용한다. HOST_Pwr 신호는 호스트 디바이스에 의해 구동된 클라이언트 장치 전원으로 작용한다. 예시적인 구성에서, 저전력 애플리케이션을 위해, 클라이언트 장치는 500mA까지 인가하도록 허용된다. HOST_Pwr 신호는 리튬-이온 형태의 배터리 또는 호스트 장치에 상주하는 배터리 팩과 같으나 이에 제한되지 않는 임시 전원 소스들로부터 제공되며 HOST_Gnd와 관련하여 3.2 내지 4.3V의 범위를 가질 수 있다.

VII. 타이밍 특성들

A. 개요

클라이언트가 호스트로부터 서비스를 확보하고 호스트가 상기 서비스를 제공하기 위해 사용되는 단계들 및 신호 레벨들은 도 43에 도시된다. 도 43에서, 도시된 신호들의 제 1 부분은 호스트로부터 전송되는 링크 섯다운 패킷을 도시하고, 데이터 라인은 고 임피던스 바이어스 회로를 사용하여 로직 0 상태로 구동된다. 어떤 데이터도 클라이언트 디스플레이 또는 드라이버를 디세이블되도록 하는 호스트에 의해 전송되지 않는다. MDDL_Stb 신호 라인에 대한 일련의 스트로브 펄스들은 하부에 도시될 수 있으며, 이는 MDDL_Stb가 링크 섯다운 패킷 동안 활성화기 때문이다. 호스트가 바이어스 회로 및 로직을 0으로 구동함에 따라 상기 패킷이 종료하고 로직 레벨이 0으로 변화하면, MDDL_Stb 신호 라인은 적절히 로직 0 레벨로 변화한다. 이는 호스트로부터 최종 신호의 전송 또는 서비스의 종료를 표시하고, 이전의 임의의 시점에서 발생할 수 있으며, 서비스의 이전의 중단 및 서비스 시작 이전의 신호들의 상태를 도시하도록 포함된다. 만약 원한다면, 상기 신호는 상기 호스트 장치에 의해 수행된 '공지된' 이전의 통신 없이 적절한 상태로 통신 링크를 리셋하기 위해 전송될 수 있다.

도 43에 도시된 것과 같이, 클라이언트로부터 출력된 신호는 먼저 0의 로직 레벨로 세팅된다. 다시 말해서, 클라이언트 출력은 고 임피던스이고, 드라이버는 디세이블된다. 서비스가 요청되면, 서비스는 드라이버를 인에이블하고 서비스 요청을 호스트로 전송하며, 상기 호스트는 라인이 로직 1 레벨로 구동되는 시간 동안 지정된 t서비스가 된다. 특정 시간량이 경과

하거나 호스트가 로직 1 레벨로 신호를 구동함으로써 링크 개시 시퀀스에 응답하는 요청을 검출(t호스트-검출이라 명명됨)하기 전에 요구될 수 있다. 상기 시점에서, 클라이언트는 요청을 무효가 되도록 하고 서비스 요청 드라이버를 디세이블하여 클라이언트로부터의 출력 라인이 다시 0 로직 레벨이 되도록 한다. 상기 시간 동안, MDDI_Stb 신호는 로직 0 레벨이 된다.

호스트는 호스트가 로직 레벨을 0으로 구동한 후의 t재시작-하이로 명명된 기간 동안 호스트 데이터 출력을 '1'레벨로 구동하고, 제 1 순방향 트래픽이 서브 프레임 헤더 패킷에서 시작한 후에 t재시작-로우로 명명된 기간 동안 MDDI_Stb를 활성화하며, 이후에 순방향 트래픽 패킷들은 전송된다. MDDI_Stb 신호는 t재시작-로우 기간 및 후속하는 서브 프레임 헤더 패킷 동안 활성이 된다.

표 7 및 표 8은 전송된 것과 같은 다양한 기간들의 시간 또는 처리 기간을 표시하고, 예시적인 최소 및 최대 데이터 전송률의 관계식을 도시하며, 상기 관계식은 다음과 같다:

$$t_{bit} = \frac{1}{\text{Link_Data_Rate}}, \text{ 상기 Link_Data_Rate는 단일 데이터 쌍의 비트 레이트이다.}$$

[표 7]

파라미터	설명	최소	타입	최대	단위
t _{service}	디스플레이 서비스 요청 펄스의 간격	60	70	80	μs
t _{restart-high}	호스트 링크 재시작 하이 펄스의 간격	140	150	160	μs
t _{restart-low}	호스트 링크 재시작 로우 펄스의 간격	40	50	60	μs
t _{display-detect}	디스플레이가 링크 재시작 시퀀스를 검출하는 시간	1		50	μs
t _{host-detect}	호스트가 서비스 요청 펄스를 검출하는 시간	1		50	μs
t _{st-stb}	MDDI_Data0 하이로부터 MDDI_Stb 토글링으로의 시간	0		10	μs
1/t _{bit-min-perf}	최소 성능 장치를 위한 링크 데이터 레이트	1		1000	kbps
1/t _{bit-min-perf}	최소 성능 장치를 위한 링크 데이터 레이트	0.001		1.1	Mbps
1/t _{bit-max-perf}	외부 장치용 최대 링크 데이터 레이트 범위	0.001		400	Mbps
1/t _{bit-max-perf}	내부 장치용 최대 링크 데이터 레이트 범위			550	Mbps
	역방향 링크 데이터 레이트	0.0005		50	Mbps
t _{bit}	외부 모드의 순방향 링크 데이터 비트 기간	2.5		10 ⁶	nsec
t _{bit}	내부 모드의 순방향 링크 데이터 비트 기간	1.8		10 ⁶	nsec

[표 8]

파라미터	설명	최소	타입	최대	단위
t _{restart-high}	호스트 링크 재시작 하이 펄스의 간격	140	150	160	Stb clks
t _{restart-low}	호스트 링크 재시작 로우 펄스의 간격	50	50	50	Stb clks
t _{stb-data-enable}	MDDI_Data0 인에이블된 링크 재시작 시퀀스에 완전히 인에이블된 MDDI_Stb	0			μs
t _{client-startup}	MDDI_Data0이 로직-하이 레벨에 도달한 후에 호스트가 MDDI_Stb를 로직-0으로 고정하는 시간	100			nsec
t _{host-detect}	MDDI_Data0 하이로부터 MDDI_Stb 토글링으로의 시간	0		1000	μs
T _{client-detect}	클라이언트가 로직-하이 레벨 성능 장치에서 MDDI_Data0을 검출하는 시간	60		80	Stb clks
t _{stb-startup}	호스트가 MDDI_Stb를 토글링하기 시작하기 전에 호스트가 MDDI_Stb를 로직-0으로 고정하는 시간	100			nsec

당업자는 도 41 및 42에 도시된 개별 엘리먼트들의 기능이 공지된 것이며, 도 42의 엘리먼트들의 기능은 도 43의 타이밍 다이어그램에 의해 증명됨을 용이하게 이해할 것이다. 도 42에 도시된 직렬 중단부들 및 절전 모드 저항들에 대한 세부사항들은 도 41로부터 생략되며, 이는 상기 정보가 데이터-스트로브 인코딩을 수행하고 이로부터 클록을 복원하는 방법을 설명하는데 불필요하기 때문이다.

B. 데이터-스트로브(strobe) 타이밍 순방향 링크

호스트 드라이버 출력으로부터 순방향 링크를 통해 데이터를 전송하기 위한 스위칭 특성들이 표 9에 도시된다. 표 9는 요구되는 최소 및 최대값 대 특정 신호 변경들이 발생하는 시간의 표 형태를 표시한다. 예를 들면, $t_{dd}-(\text{host-output})$ 라고 명명된 데이터 값('0' 또는 '1'의 출력)의 시작으로부터 종료, 즉 Data0으로부터 Data0으로의 변화가 발생하는 일반적인 시간 길이가 t_{tbit} 이며, 최소 시간은 약 $t_{tbit}-0.5\text{nsec}$ 이고, 최대 시간은 약 $t_{tbit}+0.5\text{nsec}$ 이다. Data0, 다른 데이터 라인들(DataX), 및 스트로브 라인들(Stb)에서의 변화 사이의 일반적인 간격은 도 44에 도시되며, 상기 도 44에서 Data0으로부터 Strobe, Strobe로부터 Strobe, Strobe로부터 Data0, Data0로부터 non-Data0, non-Data0로부터 non-Data0, non-Data0로부터 Strobe, 및 Strobe로부터 non-Data0의 변화들이 도시되며, 이들은 각각 $t_{tds}-(\text{host-output})$, $t_{tss}-(\text{host-output})$, $t_{tsd}-(\text{host-output})$, $t_{tddx}-(\text{host-output})$, $t_{tdxdx}-(\text{host-output})$, $t_{tdxs}-(\text{host-output})$ 및 $t_{tsdx}-(\text{host-output})$ 으로 지칭된다.

[표 9]

파라미터	설명	최소	타입	최대	단위
$t_{dd}-(\text{host-output})$	Data0으로부터 Data0로의 변화	$t_{tbit}-0.5$	t_{tbit}	$t_{tbit}+0.5$	nsec
$t_{tds}-(\text{host-output})$	Data0으로부터 Strobe로의 변화	$t_{tbit}-0.8$	t_{tbit}	$t_{tbit}+0.8$	nsec
$t_{tss}-(\text{host-output})$	Strobe로부터 Strobe로의 변화	$t_{tbit}-0.5$	t_{tbit}	$t_{tbit}+0.5$	nsec
$t_{tsd}-(\text{host-output})$	Strobe로부터 Data0로의 변화	$t_{tbit}-0.8$	t_{tbit}	$t_{tbit}+0.8$	nsec
$t_{tddx}-(\text{host-output})$	Data0로부터 non-Data0로의 변화		t_{tbit}		nsec
$t_{tdxdx}-(\text{host-output})$	non-Data0로부터 non-Data0로의 변화	$t_{tbit}-0.5$	t_{tbit}	$t_{tbit}+0.5$	nsec
$t_{tdxs}-(\text{host-output})$	non-Data0로부터 Strobe로의 변화		t_{tbit}		nsec
$t_{tsdx}-(\text{host-output})$	Strobe로부터 non-Data0로의 변화		t_{tbit}		nsec

순방향 링크를 통해 데이터를 전송하는 동일한 신호들에 대한 클라이언트 수신기 입력을 위한 일반적인 MDDI 타이밍 표시들은 표 10에 도시된다. 동일한 신호들이 논의되지만 시간 지연되기 때문에, 당업자에게 이해되는 것과 같은 신호특성들 및 개별 표시들의 의미를 설명하는데 새로운 도면은 필요하지 않다.

[표 10]

파라미터	설명	최소	타입	최대	단위
$t_{dd}-(\text{display-input})$	Data0으로부터 Data0로의 변화	$t_{tbit}-1.0$	t_{tbit}	$t_{tbit}+1.0$	nsec
$t_{tds}-(\text{display-input})$	Data0으로부터 Strobe로의 변화	$t_{tbit}-1.5$	t_{tbit}	$t_{tbit}+1.5$	nsec
$t_{tss}-(\text{display-input})$	Strobe로부터 Strobe로의 변화	$t_{tbit}-1.0$	t_{tbit}	$t_{tbit}+1.0$	nsec
$t_{tsd}-(\text{display-input})$	Strobe로부터 Data0로의 변화	$t_{tbit}-1.5$	t_{tbit}	$t_{tbit}+1.5$	nsec
$t_{tddx}-(\text{host-output})$	Data0로부터 non-Data0로의 변화		t_{tbit}		nsec
$t_{tdxdx}-(\text{host-output})$	non-Data0로부터 non-Data0로의 변화		t_{tbit}		nsec
$t_{tdxs}-(\text{host-output})$	non-Data0로부터 Strobe로의 변화		t_{tbit}		nsec
$t_{tsdx}-(\text{host-output})$	Strobe로부터 non-Data0로의 변화		t_{tbit}		nsec

도 45 및 46은 호스트가 호스트 드라이버를 각각 디세이블하거나 인에이블할 때 이에 응답하여 발생할 수 있는 지연의 존재를 도시한다. 역방향 링크 캡슐화 패킷 또는 왕복 시간 지연 측정 패킷과 같은 특정 패킷들을 전송하는 호스트의 경우에, 호스트는 원하는 패킷들, 도 45에 도시된 파라미터 CRC, 스트로브 정렬, 및 울 제로 패킷들이 전송된 후에 라인 드라이버를 디세이블한다. 그러나 도 45에 도시된 것과 같이, 라인의 상태는 0으로부터 원하는 더 높은 값으로 즉시 스위칭해야 하는 것이 아니라, 특정 제어 또는 회로 엘리먼트들이 존재하는 것을 일시적으로 달성할 수 있지만 응답하는데 호스트 드라이버 디세이블 지연 기간이라 명명된 기간이 걸린다. 상기 기간이 0 나노초(nsec.) 길이가 되도록 순간적으로 발생할 수 있지만, 보호 시간 1 또는 전환 1 패킷 주기들 동안 발생하는 d입의의 최대 길이의 주기가 되는 10nsec의 더 긴 주기로 확장될 수 있다.

도 46에서, 호스트 드라이버가 역방향 링크 캡슐화 패킷 또는 왕복 시간 지연 측정 패킷과 같은 패킷을 전송하기 위해 인에이블될 때 신호 레벨의 변경이 수행됨을 볼 수 있다. 여기에서, 보호 시간 2 또는 전환 2 패킷 주기들 이후에, 호스트 드라이버는 인에이블되거나 레벨을 '0'으로 구동하기 시작하며, 상기 값은 제 1 패킷이 전송되기 이전의 드라이버 재-인에이블 기간 동안 발생하는 호스트 드라이버 인에이블 지연 기간이라는 명칭의 기간을 통해 접근되거나 도달된다.

유사한 프로세스가 드라이버들에 대하여 발생하며, 신호는 클라이언트 장치, 여기에서 장치를 위해 전송된다. 상기 기간들의 길이에 대한 일반적인 가이드 라인들 및 그들 각각의 관계들이 하기의 표 11에 도시된다.

[표 11]

설명	최소	최대	단위
호스트 드라이버 디세이블 지연	0	10	nsec
호스트 드라이버 인에이블 지연	0	2.0	nsec
디스플레이 드라이버 디세이블 지연	0	10	nsec
디스플레이 드라이버 인에이블 지연	0	2.0	nsec

C. 데이터-스트로브 타이밍 역방향 링크

클라이언트 드라이버 출력으로부터 역방향 링크를 통해 데이터를 전송하기 위해 사용되는 데이터 및 스트로브 신호들에 대한 스위칭 특성들 및 타이밍 관계들이 도 47 및 48에 도시된다. 특정 신호 변화들을 위한 일반적인 시간이 하기에 도시된다. 도 47은 호스트 수신기 입력에서 전송되는 데이터의 타이밍 및 스트로브 펄스들의 선단 에지들과 후미 에지들 사이의 관계를 도시한다. 즉, 스트로브 신호들의 상승 또는 선단 에지에 대한 셋업 시간으로서 tsu-sr이 지정되고, 스트로브 신호들의 후미 또는 하강 에지에 대한 셋업 시간으로서 tsu-sf가 지정된다. 상기 셋업 기간들에 대한 시간의 길이는 최소 8 나노초와 비슷하다.

도 48은 역방향 데이터 타이밍에 의해 발생하는 스위칭 특성들 및 상응하는 클라이언트 출력 지연을 도시한다. 도 48에서, 전송되는 데이터의 타이밍 및 유도된 지연을 계산하는 스트로브 펄스들의 선단 에지와 후미 에지 사이의 관계를 도시한다. 즉, 스트로브 신호들의 상승 또는 선단 에지와 데이터(유효) 사이의 전파 지연으로서 tpd-sr이 지정되고, 스트로브 신호들의 후미 또는 하강 에지와 데이터 사이의 전파 지연으로서 tpd-sf가 지정된다. 상기 전파 지연 기간들에 대한 시간의 길이는 최소 8 나노초와 비슷하다.

VIII. 링크 제어의 실행(링크 제어 동작)

A. 상태 머신 패킷 프로세서

MDDI 링크를 통해 전송되는 패킷들은 일반적으로 300Mbps 또는 그 이상과 비슷한 레이트로 매우 신속하게 처리되지만, 원하는 경우에 더 낮은 레이트가 수용될 수 있다. 상기 형태의 버스 또는 전송 링크 속도는 현재 상업적으로 이용가능한 (경제적인) 범용 마이크로 프로세서들 또는 유사물이 제어하기에 매우 크다. 따라서, 상기와 같은 신호 전송을 수용하기 위한 일반적인 구현은 적절한 오디오-비주얼 서브 시스템에 전송되거나 재전송되는 패킷들을 생성하기 위해 입력 패킷 스트림을 해석하는 프로그램가능한 상태 머신을 사용하는 것이다. 상기 장치들은 공지되어 있으며, 요구되는 고속 또는 매우 고속의 동작을 달성하기 위해 동작들, 기능들 또는 상태들의 수가 제한된 회로들을 사용한다.

범용 제어기들, 프로세서들, 또는 처리 엘리먼트들은 더 낮은 속도 요구들을 가지는 제어 또는 상태 패킷들과 같은 임의의 정보에서 더 적절히 동작하거나 상기 정보를 조종하는데 사용될 수 있다. 상기 패킷들(제어, 상태 또는 다른 미리 정의된 패킷들)이 수신될 때, 상태 머신은 이들을 데이터 버퍼 또는 유사한 처리 엘리먼트를 통해 범용 프로세서로 통과시켜야 하며, 따라서 패킷들은 오디오 및 비주얼 패킷들이 그들의 적정한 작동을 위한 목적지로 전송되는 경우에 원하는 결과(효과)를 제공하도록 작동될 수 있다. 만약 향후에 마이크로프로세서들 또는 다른 범용 제어기들, 프로세서들 또는 처리 엘리먼트들이 더 높은 데이터 전송률의 처리 용량들을 달성하도록 제작되면, 하기에 논의되는 상태들 또는 상태 머신은 저장 엘리먼트 또는 매체에 저장된 프로그램들과 같이 상기 장치의 소프트웨어 제어를 사용하여 실행될 수 있다.

범용 프로세서 기능은 처리 전력을 이용하거나 몇몇 모델들 또는 그래픽 프로세서들이 몇몇 기능들을 수행하고 하드웨어 복잡성 및 비용을 감소하기 위해 컴퓨터에서 사용되는 CPU들의 처리 전력을 사용하는 것과 유사한 방식으로 컴퓨터 애플리케이션들 내부의 마이크로프로세서들(CPU들), 또는 제어기들, 프로세서들, 디지털 신호 처리기들(DSP들), 특수용 회로

들 또는 무선 장치들에서 사용되는 ASIC들에 사용가능한 사이클들을 초과함으로써 임의의 실시예들에서 실행될 수 있다. 그러나 상기 사이클 공유 또는 사용은 처리 속도, 타이밍, 또는 상기 엘리먼트들의 전체 동작에 부정적인 영향을 줄 수 있고, 따라서 다수의 애플리케이션에서 전용 회로들 또는 엘리먼트들이 상기 일반적인 처리에 바람직하다.

이미지 데이터가 디스플레이(마이크로-디스플레이)에 표시되거나 호스트 장치에 의해 전송된 모든 패킷들을 실질적으로 수신하기 위해, 디스플레이 신호 처리는 순방향 링크 채널 타이밍에 동기화된다. 즉, 클라이언트 및 클라이언트 회로들에 도달하는 신호들은 적절한 신호 처리가 발생하도록 실질적으로 시간 동기화되어야 한다. 일 실시예에 대한 신호에 의해 실행되는 상태들의 상위 레벨 상태 다이어그램은 도 49에 도시된다. 도 49에서, 상태 머신(4900)에 대하여 가능한 순방향 링크 동기 "상태들"은 하나의 비동기 프레임 상태(4904), 2개의 동기 포착 상태들(4902 및 4906), 및 3개의 동기 상태들(4908, 4910 및 4912)로 구별되어 도시된다.

시작 단계 또는 상태(4902)에 도시된 것과 같이, 디스플레이 또는 표시 장치와 같은 클라이언트는 미리 선택된 "비동기" 상태에서 시작하며, 검출된 제 1 서브 프레임 헤더 패킷에서 고유 워드를 탐색한다. 상기 비동기 상태는 타입 1 인터페이스가 선택되는 최소 통신 세팅 또는 "폴-백(fall-back)" 세팅을 나타내는 것에 유의하여야 한다. 탐색 동안 고유 워드가 발견되면, 디스플레이는 서브 프레임 길이의 필드를 저장한다. 상기 제 1 프레임에 대한 처리 동안 또는 동기화가 획득될 때까지 CRC 비트들의 검사는 수행되지 않는다. 만약 상기 서브 프레임 길이가 0이면, 동기 상태 처리는 동기화가 아직 수행되지 않았음을 나타내는 "비동기 프레임" 상태라 표시된 상태(4904)로 진행한다. 상기 처리 단계는 도 49에서 cond 3 또는 조건 3으로 표시된다. 그렇지 않으면, 만약 프레임 길이가 0 보다 크면, 동기 상태는 "하나의 동기 프레임이 발견된"이라 세팅된 인터페이스 상태인 상태(4906)로 진행한다. 상기 처리 단계는 도 49에서 cond 5 또는 조건 5로 표시된다. 부가적으로, 상태 머신들이 프레임 헤더 패킷 및 0보다 큰 프레임 길이에 대한 우수한 CRC 결정을 도출하면, 처리 단계는 "하나의 동기 프레임이 발견된" 상태로 진행한다. 상기 처리 단계는 도 49에서 cond 6 또는 조건 6으로 표시된다.

시스템이 "비동기" 상태와는 다른 상태인 경우에, 양호한 CRC 결과를 가지는 패킷이 검출되면, 인터페이스 상태는 "동기" 상태(4908)로 변경된다. 상기 처리 단계는 도 49에서 cond 1 및 조건 1로 표시된다. 다시 말해서, 임의의 패킷 내의 CRC가 정확하지 않으면, 동기 상태 처리 단계는 "비동기 프레임" 상태의 인터페이스 상태(4902)로 진행하거나 복귀한다. 상기 처리 부분은 도 49에서 cond 2 또는 조건 2로 표시된다.

B. 동기에 대한 동기 포착 시간

인터페이스는 동기화가 손실되어 "비동기 프레임" 상태로 복귀하는지를 결정하기 전에 "동기 에러들"의 개수를 수용하도록 구성될 수 있다. 도 49에서, 상태 머신이 "동기 상태"에 도달하고 어떤 에러도 발견되지 않으면, cond 1 결과를 연속적으로 발생하고 "동기" 상태를 유지한다. 그러나 cond 2 결과가 검출되면, 처리는 상태를 "하나의 동기 에러" 상태(4910)로 변화시킨다. 상기 시점에서, 만약 처리 결과가 또 다른 cond 1 결과를 검출하면, 상태 머신은 "동기" 상태로 복귀하거나, 또 다른 cond 2 결과를 발생하며, "두 개의 동기 에러들" 상태(4912)로 이동한다. 다시 말해서, 만약 cond 1이 발생하면, 처리는 상태 머신을 "동기" 상태로 복귀시킨다. 그렇지 않으면, 또 다른 cond 2가 발생하면, 상태 머신은 "비동기" 상태로 복귀한다. 인터페이스가 "링크 섯다운 패킷"을 발생하면, 링크가 데이터 전송들을 종료하고 도 49에서 cond 4 또는 조건 4와 동기화하는 상태는 전혀 없는 것처럼 "비동기 프레임" 상태로 복귀한다.

반복되는 고유 워드인 "오류 카피(false copy)"가 서브 프레임 내의 몇몇 고정된 위치에서 나타날 수 있음이 이해된다. 상기 상황에서, 상태 머신은 서브 프레임에 동기화할 수 있으며, 이는 MDD 인터페이스 처리가 "동기" 상태로 진행하는 순서로 처리될 때 서브 프레임 헤더 패킷의 CRC가 유효해야만 하기 때문이다.

서브 프레임 헤더 패킷에서의 서브 프레임 길이는 링크가 섯다운되기 전에 호스트가 단 하나의 서브 프레임만을 전송할 것임을 표시하도록 0으로 세팅될 수 있고, MDD 인터페이스는 유힬 절전 모드 상태로 설정되거나 구성된다. 상기 경우에, 디스플레이는 서브 프레임 헤더 패킷을 검출한 후에 순방향 링크를 통해 직접 패킷들을 수신해야 하며, 이는 링크가 유힬 상태로 변화하기 전에 단일 서브 프레임만이 전송되기 때문이다. 정상적이거나 일반적인 동작들에서, 서브 프레임 길이는 0이 아니고, 디스플레이는 인터페이스가 도 49에 "동기" 상태들로 도시된 상태들 내에 있는 동안 순방향 링크 패킷들을 처리한다.

외부 모드 클라이언트 장치는 순방향 링크 데이터 시퀀스를 호스트가 이미 전송하는 동안 호스트에 부착될 수 있다. 상기 경우에, 클라이언트는 호스트에 동기화되어야 한다. 디스플레이가 순방향 링크 신호를 동기화하는데 요구되는 시간은 서브 프레임 크기 및 순방향 링크 데이터 전송률에 따라 유효할 수 있다. 순방향 링크에서 랜덤 또는 더 랜덤한 데이터의 일부로서 고유 워드 "오류 카피"를 검출하는 확률은 서브 프레임 크기가 클수록 커진다. 동일한 시간에, 오류 검출로부터의 복원 능력은 낮아지며, 순방향 링크 데이터 전송률이 느려질수록 이를 실행하는데 걸리는 시간은 더 오래 걸린다.

C. 초기화

전송된 것과 같이, "개시" 시점에서, 호스트는 요구되는 최소의 데이터 전송률 1Mbps에서 또는 그 미만에서 동작하는 순방향 링크를 구성하며, 주어진 애플리케이션에 대하여 적절하게 서브 프레임 길이 및 미디어-프레임 레이트를 형성한다. 즉, 순방향 및 역방향 링크들 모두는 타입 1 인터페이스를 사용하여 동작을 시작한다. 상기 파라미터들은 일반적으로 호스트가 클라이언트 디스플레이(또는 다른 타입의 클라이언트 장치)에 대한 용량 또는 요구되는 구성을 결정하는 동안 임시적으로 사용된다. 호스트는 디스플레이 또는 클라이언트가 디스플레이 용량 패킷에 응답하는 것을 요청하기 위해 1의 값으로 세팅된 요청 플래그들의 비트 '0'을 가지는 역방향 링크 캡슐화 패킷 다음에 순방향 링크를 통해 서브 프레임 헤더 패킷을 전송한다. 디스플레이가 순방향 링크를 통해 동기를 포착한 이후에, 역방향 링크 또는 채널을 통해 디스플레이 요청 및 상태 패킷 및 디스플레이 용량 패킷을 전송한다.

호스트는 최적 또는 적절한 레벨의 성능을 위해 링크를 재구성하는 방법을 결정하기 위하여 디스플레이 용량 패킷의 콘텐츠를 검사한다. 호스트는 호스트 및 디스플레이가 서로 호환가능한 프로토콜 버전들을 사용하는지를 확인하기 위해 프로토콜 버전 및 최소 프로토콜 버전 필드들을 검사한다. 프로토콜 버전들은 일반적으로 디스플레이 용량 패킷의 제 1의 2개 필드들로 유지되어, 프로토콜의 다른 엘리먼트들이 호환가능하지 않거나 호환가능한 것으로 이해되지 않을 때 용량이 결정될 수 있다.

D. CRC 처리

모든 패킷 타입들에 대하여, 패킷 프로세서 상태 머신은 CRC 검사기가 적절하게 제어되는 것을 보장한다. 이는 CRC 비교로 인해 하나 또는 그 이상의 에러들이 검출될 때 CRC 에러 카운터를 증가시키고, 각각의 처리된 서브 프레임의 시작시 CRC 카운터를 리셋시킨다.

E. 동기화 검사의 선택적인 손실

상기와 같은 일련의 단계들 또는 상태들은 더 높은 데이터 전송률 또는 스루풋 속도를 발생하도록 동작하지만, 출원인은 클라이언트가 호스트와의 동기화가 손실되었다고 선언하기 위해 사용하는 조건들에서의 선택적인 배치 또는 변경이 더 높은 데이터 전송률 또는 스루풋을 달성하기 위해 효율적으로 사용될 수 있음을 발견하였다. 새로운 실시예는 동일한 기본 구성을 가지지만, 상태가 변경되는 조건들이 다르다. 또한, 새로운 카운터는 서브 프레임 동기화에 대하여 검사하는 것을 지원하도록 구현된다. 상기 단계들 및 조건들은 도 63에 도시되며, 방법 또는 상태 머신의 동작들을 설정하는데 유용한 일련의 상태들 및 조건들을 도시한다. 명확함을 위해 "동기 포착 상태" 및 "동기 상태"만이 도시된다. 또한, 결과적인 상태들이 실질적으로 동일하기 때문에, 상태 머신 자체가 될 수 있도록 동일한 도면부호를 사용한다. 그러나 상태들(및 상태 머신 동작)을 변화시키기 위한 조건들은 다소 변경되며, 따라서 모든 조건들은 차이들을 식별하는 편리함으로 두 도면들(1, 2, 3, 4, 5, 6 대 61, 62, 63, 64, 65) 사이에서 명확함을 위해 다시 번호가 표시된다. 비동기 프레임 상태는 상기 논의에서 고려되지 않기 때문에, 하나의 상태(4904) 및 조건 6은 도면에서 더 이상 사용되지 않는다.

도 63에서, 시스템 또는 클라이언트(디스플레이 및 표시를 위한)는 도 49에서와 같이 미리 선택된 "비동기" 상태(4902)에서 상태 머신(5000)을 시작한다. 비동기 조건(4902)으로부터 상태들을 변화시키기 위한 제 1 상태 변경은 동기 패킷이 발견되는 조건 64가 된다. 서브 프레임 헤더가 상기 패킷(조건 61을 충족하는)을 통과한다고 가정할 때, 패킷 프로세서 상태 머신의 상태는 동기 상태(4908)로 변경될 수 있다. 조건 62의 동기 에러는 상태 머신이 상태(4910)로 시프트하도록 하고, 제 2 발생이 상태(4912)로 시프트하도록 한다. 그러나 MDDI 패킷의 임의의 CRC 실패가 상태 머신이 동기 상태(4908)가 아닌 하나의 동기 에러 상태(4910)가 되도록 하는 것이 발견되었다. 임의의 MDDI 패킷의 또 다른 CRC 실패는 2개의 동기화 실패 상태(4912)로 이동할 것이다. 정확한 CRC 값으로 디코딩된 패킷은 상태 머신이 동기 상태(4908)로 복귀하도록 할 것이다.

변경되는 것은 '모든' 패킷에 대한 결정 또는 CRC 값을 사용하는 것이다. 즉, 상태 머신이 모든 패킷이 서브 프레임 헤더 패킷들을 관찰하는 대신에 동기화의 손실을 결정하기 위해 모든 패킷에 대한 CRC 값을 검색하도록 하는 것이다. 상기 구성 또는 프로세스에서, 동기화의 손실은 고유 워드 또는 서브 프레임 헤더 CRC 값들을 사용할 때 결정되지 않는다.

상기 새로운 인터페이스 구현은 MDD 인터페이스 링크가 동기화 실패들을 매우 빠르게 인식하도록 하며, 따라서 상기 실패들을 더 빨리 복원하도록 한다.

상기와 같은 시스템이 더 견고해지도록 하기 위해, 클라이언트는 서브 프레임 카운터를 부가 또는 사용해야 한다. 클라이언트는 고유 워드가 신호 내에 도달하거나 발생하는 것으로 예측되는 시간에 고유 워드의 존재를 검사한다. 만약 고유 워드가 정확한 시간에 발생하지 않으면, 클라이언트는 서브 프레임 길이 이상인 몇몇의(여기에서 3개의) 패킷 시간 및 기간들을 대기해야하는 경우보다 더 빨리 동기화 실패가 발생하는 것을 인식할 수 있다. 만약 고유 워드에 대한 테스트가 고유 워드가 존재하지 않는다고 표시하면, 다시 말해서 타이밍이 부정확하다고 표시하면, 클라이언트는 즉시 동기화의 링크 손실을 선언하고 동기 상태로 이동할 수 있다. 적절한 고유 워드의 존재를 검사하는 프로세스는 유일한 단어가 부정확하다고 말하는 상태 머신에 조건 65(cond 65)를 부가한다. 만약 서브 프레임 패킷이 클라이언트에서 수신되어 매칭되지 않는 것으로 예측되면, 클라이언트는 즉시 동기 상태(4902)가 될 수 있고, 상태들(4910 및 4912)을 가로지르는 다수의 동기 에러들(조건 62)에 대하여 대기하는 추가의 시간을 절약한다.

상기와 같은 변경은 서브 프레임 길이를 카운트하기 위해 클라이언트 코어에서 추가의 카운터 또는 카운팅 기능을 사용한다. 일 실시예에서, 카운트다운 기능이 사용되고, 현재 처리된 임의의 패킷의 전송은 카운터가 종료되는 경우에 서브 프레임의 유일한 단어를 검사하기 위해 중단된다. 선택적으로, 카운터는 현재 패킷이 검사되는 포인트에서 요구되는 최대값 또는 특정 요구값에 비교되는 카운트를 카운트 업할 수 있다. 상기 프로세스는 대단히 긴 패킷 길이를 가지고 클라이언트에서 부정확하게 수신된 디코딩 패킷들로부터 클라이언트를 보호한다. 임의의 다른 패킷을 차단하기 위해 요구되는 서브 프레임 길이 카운터가 디코딩된 경우에, 동기화의 손실은 어떤 패킷도 서브 프레임 경계를 교차하지 않아야 하기 때문에 결정될 수 있다.

IX. 패킷 처리

상태 머신이 수신하는 전송된 패킷의 각 타입에 대하여, 인터페이스의 동작을 실행하기 위한 특정 처리 단계 또는 일련의 단계들을 수행해야 한다. 순방향 링크 패킷들은 하기의 표 12에 열거된 예시적인 처리에 따라 처리된다.

[표 12]

패킷 타입	패킷 프로세서 상태 머신 응답
서브 프레임 헤더(SH)	우수한 패킷을 확인하고, 서브 프레임 길이 필드를 포착하여, 범용 프로세서에 패킷 파라미터들을 전송한다.
채움(F)	데이터를 무시한다.
비디오 스트림(VS)	비디오 데이터 포맷 기술자 및 다른 파라미터들을 해석하고, 필요한 경우에 팩킹된 픽셀 데이터를 언팩킹하고, 필요한 경우에 컬러 맵을 통해 픽셀들을 해석하고, 픽셀 데이터를 비트맵 내의 적절한 위치들에 기록한다.
오디오 스트림(AS)	오디오 샘플 레이트 세팅을 오디오 샘플 클럭 발생기로 전송하고, 특정 크기의 오디오 샘플들을 구분하고, 필요한 경우에 오디오 샘플 데이터를 언팩킹하며, 오디오 샘플들을 적절한 오디오 샘플 FIFO로 라우팅한다.
컬러 맵(CM)	컬러 맵 크기 및 오프셋 파라미터들을 판독하고, 컬러맵 메모리 또는 저장 위치에 컬러 맵 데이터를 기록한다.
역방향 링크 캡슐화(REL)	적절한 시간에 역방향으로 패킷들을 전송하는 것을 용이하게 한다. 역방향 링크 플래그들이 검사되고, 디스플레이 용량 패킷들이 필요한 경우에 전송된다. 디스플레이 요청 및 상태 패킷들이 적절하게 전송된다.
클라이언트 용량(CC)	역방향 링크 캡슐화 패킷의 역방향 링크 플래그 필드를 사용하여 호스트에 의해 요청되는 경우에 상기 패킷 타입을 전송한다.
키보드(K)	만약 존재한다면 사용이 요구되는 키보드 타입의 장치와 통신하는 범용 프로세서로 상기 패킷들을 통과시킨다.
포인팅 장치(PD)	만약 존재한다면 사용이 요구되는 포인팅 타입의 장치와 통신하는 범용 프로세서로 상기 패킷들을 통과시킨다.
링크 섣다운(LS)	링크가 섣다운된 사실을 기록하고 범용 프로세서에 통지한다.
클라이언트 서비스 요청 및 상태(CSRs)	상기 패킷을 역방향 링크 캡슐화 패킷 내의 제 1 패킷으로서 전송한다.
비트 블록 전송(BPT)	비디오 데이터 포맷 기술자와 같은 패킷 파라미터들을 해석하고, 어떤 픽셀들이 가장 먼저 이동할 것인지를 결정하여 요구되는 경우에 비트맵 내의 픽셀들을 이동시킨다.
비트맵 영역 채움(BAF)	패킷 파라미터들을 해석하고, 필요한 경우에 컬러 맵을 통해 픽셀들을 번역하여 픽셀 데이터를 비트맵 내의 적절한 위치들에 기록한다.
비트맵 패턴 채움(BPF)	패킷 파라미터들을 해석하고, 필요한 경우에 팩킹된 픽셀 데이터를 언팩킹하며, 필요한 경우에 컬러 맵을 통해 픽셀들을 번역하여 픽셀 데이터를 비트맵 내의 적절한 위치들에 기록한다.
통신 링크 채널(CLC)	상기 데이터를 범용 프로세서에 직접 전송한다.
절전 모드 동안의 클라이언트 서비스 요청(CSR)	범용 프로세서는 요청을 전송하는 하위 레벨 기능들을 제어하고, 자기 자신을 통해 재시작하는 링크와의 경합을 감출한다.
인터페이스 타입 핸드 오프 요청(ITHR) 및 인터페이스 타입 확인 응답(ITA)	상기 패킷들을 범용 프로세서로 통과시킬 수 있다. 상기 타입의 패킷을 수신하고 확인 응답에 대한 응답을 공식화하기 위한 로직은 실질적으로 최소가 된다. 따라서, 상기 동작은 패킷 프로세서 상태 머신 내에 구현될 수 있다. 결과적인 핸드오프는 하위 레벨 물리 계층 동작으로서 발생하고, 범용 프로세서의 기능에 영향을 미칠 수 없다.
수행 타입 핸드오프(PTH)	상기 패킷들을 범용 프로세서에 직접 전송함으로써 상기 패킷들에 작용할 수 있고, 하드웨어에 모드 변경을 수행할 것을 명령한다.

X. 역방향 링크 데이터 레이트 감소

매우 바람직한, 최대 또는 최적(스케일) 역방향 링크 데이터 레이트를 달성하기 위해서 임의의 방식으로 호스트 링크 제어기에서 사용되는 임의의 파라미터들이 조정 또는 구성될 수 있음이 본 발명가들에 의해 파악되었다. 예를 들어, 역방향 링크 인캡슐레이션 패킷의 역방향 데이터 패킷 필드를 전달하는데 사용되는 시간 동안, MDDI_Stb 신호 쌍은 순방향 링크 데이터 레이트의 절반에 해당하는 주기적인 데이터 클럭을 생성하기 위해서 토글링한다. 이는 올 제로들을 전송하는 것처럼, MDDI_Data0에 대응하는 MDDI_Stb 신호를 호스트 링크 제어기가 생성하기 때문에 발생한다. MDDI_Stb 신호는 호스트로부터 클라이언트로 전달되며, 클라이언트에서 클라이언트로부터 역방향 링크 데이터를 전달하는 클럭 신호를 생성하기 위해서 사용되며, 이를 통해 역방향 데이터가 호스트로 다시 전달된다. MDDI를 사용하는 시스템에서 순방향 및 역방향 경로들 상에서 신호 전달 및 처리에서 직면하는 전형적인 지연량의 예는 도50에 제시된다. 도50에서, 일련의 지연 값들 1.5nsec, 8.0nsec, 2.0nsec, 1.0nsec, 1.5nsec, 8.0nsec, 및 2.5nsec가 Stb+/- 생성, 케이블 전달 대 클라이언트, 클라이언트 수신기, 클럭 생성, 신호 클럭킹, 데이터0+/- 생성, 케이블 전달 대 호스트, 및 호스트 수신기단 각각에 대한 처리 부 근처에서 제시된다.

순방향 링크 데이터 및 직면되는 신호 처리 지연에 기반하여, 이러한 "왕복시간" 효과 또는 완료될 한 세트의 이벤트들에 대한 MDDL_Stb 신호상의 1 사이클 보다 많은 시간을 필요로 하고, 이는 바람직하지 않은 소비 시간 또는 사이클 량을 초래한다. 이러한 문제를 극복하기 위해서, 역방향 레이트 디바이저는 역방향 링크 상에서 1 비트가 다수의 MDDL_Stb 신호 사이클들을 스캔(span)하는 것을 가능하게 한다. 이는 역방향 링크 데이터 레이트가 순방향 링크 레이트 보다 작다는 것을 의미한다.

인터페이스를 통한 신호 지연들의 실제 길이는 사용되는 각각의 호스트-클라이언트 시스템 또는 하드웨어에 따라 다르다. 비록 요구되지는 않지만, 각각의 시스템은 일반적으로 시스템의 실제 지연을 측정하기 위해서 왕복 지연 측정 패킷을 사용함으로써 보다 양호한 성능을 제공할 수 있고, 이를 통해 역방향 레이트 디바이저는 최적값으로 설정될 수 있다. 호스트는 간단하지만 낮은 속도에서 동작하는 기본 데이터 샘플링 또는, 복잡하지만 보다 높은 역방향 데이터 레이트를 지원하는 향상된 데이터 샘플링 중 하나를 지원한다. 이러한 방법들 둘 모두를 지원하는 클라이언트 기능이 동일하게 고려된다.

왕복 지연은 호스트가 클라이언트로 왕복 지연 측정 패킷을 전송하도록 함으로써 측정된다. 클라이언트는 소위 측정 주기 필드로 지칭되는 그 패킷 내의 미리 선택된 측정 윈도우 내에서, 또는 측정 윈도우 기간 동안 호스트로 일련의 1들을 전송함으로써 이러한 패킷에 응답한다. 이러한 측정의 상세한 타이밍은 앞에서 설명되었다. 왕복 지연은 역방향 링크 데이터가 안전하게 샘플링되는 레이트를 결정하는데 사용된다.

왕복 지연 측정은, 0xff, 0xff, 0x00 응답 시퀀스가 클라이언트로부터 다시 호스트로 수신될 때, 시간 주기의 시작에서 측정 주기 필드의 시간 사이에서 발생하는 순방향 데이터 클록 인터벌의 수를 결정, 검출, 또는 카운팅하는 것을 포함한다. 클라이언트로부터의 응답이, 측정 카운트가 증분되기 전의 순방향 링크 클록 주기의 일부에서 수신될 수 있음을 주목하여야 한다. 이러한 수정되지 않은 값이 역방향 레이트 디바이저를 계산하는데 사용되면, 이는 신뢰성 없는 데이터 샘플링으로 인한 역방향 링크 상의 비트 에러를 초래할 수 있다. 이러한 상황의 예는 도51에 제시되고, 여기서 호스트에서 MDDL_Data, 호스트에서 MDDL_Stb, 호스트 내의 순방향 데이터 클록, 및 지연 카운트를 나타내는 신호들은 그래픽 형태로 제시된다. 도51에서, 응답 시퀀스는, 지연 카운트가 6에서 7로 증분되기 바로 전에, 순방향 링크 클록 주기의 일부에서 클라이언트로부터 수신되었다. 지연이 6으로 가정되면, 호스트는 비트 전이 바로 후에, 또는 가능하게는 비트 전이 도중에 역방향 데이터를 샘플링할 것이다. 이는 호스트에서 에러 샘플링을 초래할 수 있다. 이러한 이유로, 측정된 지연은 역방향 데이터 디바이저를 계산하는데 사용되기 전에 1만큼 증분되어야 한다.

역방향 링크 디바이저는 역방향 링크 데이터 샘플링 전에 호스트가 대기하여야하는 MDDL_Stb 사이클들의 수이다. MDDL_Stb는 순방향 링크 레이트의 1/2인 레이트에서 사이클되기 때문에, 수정된 왕복 지연 측정은 2로 나뉘지고, 그리고 나서 다음 정수로 반올림되어야 할 필요가 있다. 수식으로 표현하면, 관계식은 다음과 같다:

$$reverse_rate_divisor = RoundUpToNextInteger\left(\frac{round_trip_delay + 1}{2}\right)$$

예를 들어, 이는 다음과 같다.

$$reverse_rate_divisor = RoundUpToNextInteger\left(\frac{6+1}{2}\right) = 4$$

이러한 예에서 사용되는 왕복 지연 측정치가 6이 아니라, 7이라 할지라도, 역방향 레이트 디바이저는 동일하게 4가 된다.

역방향 링크 데이터는 역방향 링크 클록의 상승 가장자리에서 호스트에 의해 샘플링된다. 역방향 링크 클록을 생성하기 위해서, 호스트 및 클라이언트 (디스플레이) 모두에서 존재하는 카운터 또는 공지된 유사한 회로(장치)가 존재한다. 카운터들은 초기화되어, 역방향 링크 클록의 제1 상승 가장자리는 역방향 링크 인캡슐레이션 패킷의 역방향 링크 패킷 필드의 제1 비트의 시작에서 발생한다. 예를 들어, 이는 도52에서 제시된다. 카운터들은 MDDL_Stb 신호의 각각의 상승 가장자리에서 증분되고, 카운터들이 랩 어라운드(wrap around)할 때까지 발생하는 카운터들의 수는 역방향 링크 인캡슐레이션 패킷

의 역방향 레이트 디바이저 파라미터에 의해 설정된다. MDDL_Stb 신호는 순방향 링크 레이트의 1/2에서 토글링하기 때문에, 역방향 링크 레이트는 역방향 레이트 디바이저에 의해 나뉘진 순방향 링크 레이트의 1/2이다. 예를 들어, 순방향 링크 레이트가 200Mbps이고, 역방향 레이트 디바이저가 4이면, 역방향 링크 데이터 레이트는 다음과 같다:

$$1/2 * 200Mbps/4 = 25Mbps.$$

역방향 링크 인캡슐레이션 패킷에서 MDDL_Data0 및 MDDL_Stb 신호 라인들의 타이밍을 보여주는 예는 도52에 제시되며, 여기서 예로서 사용되는 패킷 파라미터들은 다음과 같은 값들을 갖는다:

패킷 길이 = 1024(0x0400) 턴 어라운드 1 길이 = 1

패킷 타입 = 65(0x41) 턴 어라운드 2 길이 = 1

역방향 링크 플래그 = 0 역방향 레이트 디바이저 = 1

파라미터 CRC = 0xdb43 올 제로 는 0x00

패킷 길이 및 파라미터 CRC 사이의 패킷 데이터는 다음과 같다.

0x00, 0x04, 0x41, 0x00, 0x02, 0x01, 0x01, 0x43, 0xdb, 0x00,...

클라이언트로부터 리턴되는 제1 역방향 링크 패킷은 패킷 길이 7 및 패킷 타입 70을 갖는 클라이언트 요청 및 상태 패킷이다. 이러한 패킷은 바이트 값 0x07, 0x00, 0x46, ... 등이다. 그러나, 단지 제1 바이트(0x07)만이 도52에서 보여진다. 이러한 제1 역방향 링크 패킷은 실제적인 역방향 링크 지연을 도시하기 위해서 도면에서 대략 1 역방향 링크 클록 주기만큼 시간-쉬프팅된다. 제로를 갖는 호스트 대 클라이언트 왕복 지연에 대한 이상적인 파형은 점선 트레이스로 제시된다.

패킷 타입 이후에, 파라미터 CRC 필드의 MS 바이트가 전달되고, 그리고 나서 올 제로 필드가 전달된다. 호스트로부터의 데이터가 레벨을 변경할 때, 호스트로부터의 스트로브는 1에서 0으로, 그리고 다시 1로 스위칭되어, 보다 넓은 펄스들을 형성한다. 데이터가 0으로 진행하면, 스트로브는 보다 높은 레이트로 스위칭하고, 데이터 라인 상의 데이터의 변경만이 일 라인먼트 필드의 끝 부근에서 변경을 초래한다. 스트로브는 연장된 시간 주기들 동안 데이터 신호의 고정된 0 또는 1 레벨들로 인해 도면의 나머지 부분에 대해 보다 높은 레이트에서 스위칭하고, 그 전이들은 펄스 패턴으로(가장자리) 떨어진다.

클록이 역방향 링크 패킷들을 수용하도록 개시될 때, 호스트에 대한 역방향 링크 클록은 턴 어라운드 1 주기의 끝까지 0이다. 도면의 하단부의 화살표는 데이터가 샘플링될 때를 표시한다. 전달되는 패킷 필드의 제1 바이트(여기서 11000000)는 턴 어라운드 1 후에 시작하는 것으로 제시되고, 라인 레벨은 디스에이블되는 호스트 드라이버로부터 안정화된다. 제1 비트의 이동에서의 지연(비트 3에 대해 제시된 것과 같이)은 데이터 신호에 대한 점선 라인에서 보여진다.

도53에서, 순방향 링크 데이터 레이트에 기반하여 역방향 레이트 디바이저의 일반적인 값들을 관측할 수 있다. 실제 역방향 레이트 디바이저는 적절한 역방향 링크 동작을 보장하기 위해서 왕복 링크 측정의 결과로서 결정된다. 제1 영역(5302)은 안전한 동작 영역에 대응하고, 제2 영역(5304)은 주변(marginal) 성능 영역에 대응하며, 제3 영역(5306)은 적절하게 동작할 것 같지 않은 세팅들을 표시한다.

왕복 지연 측정 및 역방향 디바이저 세팅은, 순방향 또는 역방향 링크 중 하나에서 임의의 인터페이스 타입 세팅들을 가지고 동작하는 동안 동일한데, 왜냐하면 이들은 송신 또는 수신되는 비트들의 수가 아니라, 실제 클록 주기들의 유닛들(units)의 관점에서 표현 및 동작되기 때문이다.

XI. 턴-어라운드 및 보호 시간들

상술한 바와 같이, 역방향 링크 인캡슐레이션 패킷의 턴 어라운드 1 필드 및 왕복 지연 측정 패킷의 보호 시간 1 필드는 클라이언트 인터페이스 드라이버들이 인에이블되기 전에 호스트 인터페이스 드라이버들이 디스에이블되는 것을 허용하는 시간 길이들에 대한 값들을 지정한다. 턴 어라운드 2 및 보호 시간 2 필드들은 호스트 드라이버들이 인에이블되기 전에, 클라이언트 드라이버들이 디스에이블되는 것을 허용하는 시간 값들을 제공한다. 보호 시간 1 및 보호 시간 2 필드들은 일반적으로, 조정이 의도되지 않는 길이들에 대한 미리 설정 또는 선택된 값들로 채워진다. 사용되는 인터페이스 하드웨어에 기반하여, 이러한 값들은 실험적 데이터를 사용하여 전개되고, 동작을 개선하기 위해서 일부 예들에서 조정된다.

수개의 인자들이 턴 어라운드 1의 길이 결정에 기여하고, 이들은 순방향 링크 데이터 레이트, 및 호스트의 MDDI_Data 드라이버들의 최대 디스에이블 시간이다. 최대 호스트 드라이버 디스에이블 시간은 테이블 XI에서 규정되고, 여기서 드라이버는 디스에이블에 최대 10nsec 및 인에이블에 대략 2nsec가 소요된다. 디스에이블되는 호스트 드라이버에 요구되는 순방향 링크 클록들의 최소 수는 다음 관계식으로 표현된다:

$$Clocks_to_disable_{TA1} = \frac{ForwardLinkDataRate}{InterfaceTypeFactor_{FWD}} \cdot HostDriverDisableDelay_{max}$$

턴 어라운드 1의 허용되는 값의 범위는 다음 관계식에 따라 표현된다:

$$Turn_Around_1 \geq RoundUpToNextInteger \left(\frac{Clocks_to_disable_{TA1}}{8} \cdot InterfaceTypeFactor_{FWD} \right)$$

여기서, 인터페이스 타입 인자는 타입 1에 대해 1이고, 타입 2에 대해 2이며, 타입 3에 대해 4이고, 타입 4에 대해 8이다.

상기 2개의 등식들을 결합하면, 인터페이스 항은 소거되고, 턴 어라운드 1은 다음과 같이 정의된다:

$$Turn_Around_1 = RoundUpToNextInteger \left(\frac{ForwardLinkDataRate \cdot HostDriverDisableDelay_{max}}{8} \right)$$

예를 들어, 1500Mbps 타입 3 순방향 링크는 다음과 같은 턴 어라운드 1을 사용한다:

$$Turn_Around_1 = RoundUpToNextInteger \left(\frac{1500Mbps \cdot 10nsec}{8} \right) = 2Bytes$$

왕복 지연이 증가하면, 클라이언트가 인에이블되는 시간에서 호스트가 디스에이블되는 시간의 포인트로부터 타이밍 마진이 개선된다.

턴 어라운드 2에서 일반적으로 사용되는 타이밍 길이를 결정하는 인자들은 순방향 링크 데이터 레이트, 클라이언트의 MDDI_Data 드라이버의 최대 디스에이블 시간, 및 통신 링크의 왕복 지연이다. 클라이언트 드라이버를 디스에이블하는데 필요한 시간의 계산은 상술한 호스트 드라이버에서 설명한 것과 동일하고, 다음 관계식에 따라 정의된다:

$$Clocks_to_disable_{TA2} = \frac{ForwardLinkDataRate}{InterfaceTypeFactor_{FWD}} \cdot DisplayDriverDisableDelay_{max}$$

그리고, 턴 어라운드 2에 대한 허용되는 값의 범위는 다음과 같이 표현된다:

$$Turn_Around_2 \geq RoundUpToNextInteger \left(\frac{Clocks_to_disable_{TA2} + round_trip_delay + 1}{\left(\frac{8}{InterfaceTypeFactor_{FWD}} \right)} \right)$$

예를 들어, 10 순방향 링크 클록 지연을 갖는 1500Mbps 타입 3 순방향 링크는 다음과 같이 턴 어라운드 2 지연을 사용한다:

$$Clocks_to_disable_{TA2} = \frac{1500Mbps}{4} \cdot 10nsec = 3.75$$

$$Turn_Around_2 \geq RoundUpToNextInteger \left(\frac{3.75 + 10 + 1}{\left(\frac{8}{4} \right)} \right) = 8$$

XII. 대안적인 역방향 링크 타이밍

상술한 타이밍 및 보호 밴드들의 사용이 높은 데이터 전송 레이트 인터페이스를 달성하도록 작용하지만, 본 발명가들은 역방향 타이밍 발견을 변경함으로써, 왕복 시간 보다 짧은 역방향 비트 길이들을 허용하는 기술을 발견하였다.

상술한 바와 같이, 역방향 링크 타이밍에 대한 이전 방법은 제1 비트가 IO 클록의 상승 가장자리에서 샘플링될 때까지, 클록 사이클들의 수가 역방향 타이밍 패킷의 보호 시간 1의 최종 비트로부터 카운트되도록 구성된다. 이것은 MDDD 인터페이스에 대한 입력 및 출력들을 타이밍하는데 사용되는 클록 신호(들)이다. 역방향 레이트 디바이저에 대한 계산은 다음과 같이 주어진다:

$$reverse_rate_divisor = RoundUpToNextInteger \left(\frac{round_trip_delay + 1}{2} \right)$$

이는 매우 신뢰성있는 역방향 링크를 초래하는 왕복 지연과 동일한 비트폭을 제공한다. 그러나, 역방향 링크는, 본 발명가들이 이용하기를 원하는, 고속 데이터 전송 레이트에서, 또는 고속에서 실행될 수 있음이 보여졌다. 새로운 독창적인 기술들은 보다 높은 속도에 도달하기 위해서 인터페이스의 추가적인 기능들을 이용하는 것을 허용한다.

이는 하나가 샘플링될 때까지 호스트가 클록 사이클들의 수를 카운트함으로써 달성될 수 있지만, 호스트는 역방향 타이밍 패킷 동안 상승 및 하강 가장자리 모두에서 데이터 라인을 샘플링한다. 이는 호스트가 역방향 비트 내에서 가장 유용한 또는 최적의 샘플링 포인트를 선택할 수 있도록 하여주고, 이를 통해 비트가 안정화되는 것을 보장한다. 즉, 역방향 트래픽 역방향 인캡슐레이션 패킷들에 대한 데이터를 샘플링하는 최적의 상승 가장자리를 발견하는 것이다. 최적 샘플링 포인트는 역방향 링크 디바이저, 및 첫 번째 것이 상승 가장자리에서 검출되었는지 또는 하강 가장자리에서 검출되었는지에 의존한다. 새로운 타이밍 방법은, 역방향 인캡슐레이션 패킷에서 샘플링할 장소를 결정하기 위해서, 역방향 링크에 대해 클라이언트에 의해 전송된 OxFF OxFF OxOO 패턴의 제1 가장자리를 호스트가 단순히 찾는 것을 허용한다.

역방향 비트의 도달 및 어떻게 그 비트가 다양한 역방향 레이트 디바이저들을 찾는지에 대한 예들이, 최종 보호 시간 1 비트 이후로 발생한 클록 사이클들의 수와 함께 도64에서 제시된다. 도64에서, 상승 및 하강 가장자리(상승/하강으로 표시됨) 사이에서 제1 가장자리가 발생하면, 1의 역방향 레이트 디바이저에 대한 최적 샘플링 포인트, 최적 샘플 포인트는 "b"로 표시되는 클록 사이클 가장자리이고, 이는 역방향 비트 주기 내에서 발생하는 유일한 상승 가장자리이기 때문이다. 2의 역방향 레이트 디바이저에 있어서, 최적 샘플링 포인트는 여전히 클록 사이클 리딩 가장자리 "b"이며, 이는 사이클 가장자리 "c"가 "b" 보다 비트 가장자리에 근접하기 때문이다. 4의 역방향 레이트 디바이저에 있어서, 최적 샘플링 포인트는 클록 사이클 가장자리 "d"이며, 이는 그 값이 안정화되는 역방향 비트의 백 가장자리에 더 근접하기 때문이다.

도64를 살펴보면, 제1 가장자리가 상승 및 하강 가장자리(상승/하강으로 표시됨) 사이에서 발생하면, 1의 역방향 레이트 디바이저에 대한 최적 샘플링 포인트는 샘플링 포인트 클록 사이클 가장자리 "a"이며, 이는 역방향 비트 시간 주기 내의 유일한 상승 가장자리이기 때문이다. 2의 역방향 레이트 디바이저에 있어서, 최적 샘플링 포인트는 가장자리 "b"이며, 4의 역방향 레이트 디바이저에 있어서, 최적 샘플링 포인트는 가장자리 "c"이다.

역방향 레이트 디바이저가 커짐에 따라, 최적 샘플링 포인트는 확인 또는 선택하기가 더욱 쉬워짐을 알 수 있고, 이는 중간에 가장 근접한 상승 가장자리이어야 한다.

타이밍 패킷 데이터의 상승 데이터 가장자리가 데이터 라인에서 관측되기에 앞서, 상승 클록 가장자리들의 수를 발견하기 위해서, 호스트는 이러한 기술을 사용할 수 있다. 그리고 나서, 상승 및 하강 가장자리 또는 하강 및 상승 가장자리 사이에 가장자리가 발생하는지 여부, 및 역방향 레이트 디바이저가 무엇인지에 기반하여, 비트가 항상 가능한 한 중간에 근접하여 샘플링되는 것을 보장하기 위해서, 카운터의 추가될 추가적인 클록 사이클들의 수를 결정할 수 있다.

호스트가 클록 사이클들의 수를 선택 또는 결정하면, 특별한 역방향 레이트 디바이저가 잘 동작하는지 여부를 결정하기 위해서, 클라이언트를 통해 다양한 역방향 레이트 디바이저를 "실험"할 수 있다. 호스트(및 클라이언트)는 1 디바이저로 시작할 수 있고, 이러한 역방향 레이트가 데이터를 전달하는데 적절하게 기능하는지 여부를 결정하기 위해서 클라이언트로부터 수신된 역방향 상태 패킷의 CRC를 검사할 수 있다. CRC가 파손되면, 샘플링 에러의 가능성이 존재하고, 호스트는 역방향 레이트 디바이저를 증가시키고, 상태 패킷의 요청을 다시 시도할 수 있다. 제2 요청된 패킷이 파손되면, 디바이저는 다시 증가되고, 요청이 다시 이뤄진다. 이러한 패킷이 정확하게 디코딩되면, 이러한 역방향 레이트 디바이저가 모든 장래의 역방향 패킷들에 대해 사용될 수 있다.

이러한 방법은 효과적이고 유용한데, 왜냐하면 역방향 타이밍은 초기 왕복 타이밍 추정치로부터 변경되어서는 안 되기 때문이다. 순방향 링크가 안정적이면, 클라이언트는, 비록 역방향 링크 고장이 존재하더라도, 순방향 링크 패킷들의 디코딩을 계속하여야 한다. 물론, 이러한 방법이 완벽한 역방향 링크를 보장하지는 않기 때문에, 그 링크에 대한 역방향 링크 디바이저를 설정하는 것은 호스트의 책임이다. 또한, 디바이저는 IO 클록을 생성하는데 사용되는 클록의 품질에 주로 의존할 것이다. 그 클록이 상당량의 지터를 가지면, 샘플링 에러의 확률이 높아진다. 이러한 에러 확률은 왕복 지연에서 클록 사이클들의 양을 증가시킨다.

이러한 구현은 타입 1 역방향 데이터에서 가장 잘 동작하는 것처럼 보이지만, 단지 하나의 데이터 쌍들에 대해 가장 잘 동작하는 레이트에서 링크를 실행하는데 너무 큰 데이터 라인들 사이의 스큐로 인해 타입 2 내지 타입 4 역방향 데이터에 대해서는 문제점을 초래한다. 그러나, 데이터 레이트는 동작을 위해, 타입 2 내지 타입 4인 경우에도 이전 방법으로 감소될 필요는 없다. 이러한 방법은 이상적인 또는 최적의 클록 샘플 위치를 선택하기 위해서 각 데이터 라인 상에서 중복되는 경우, 가장 잘 동작한다. 이들이 각 데이터 쌍에 대해 동일한 샘플 시간에 있으면, 이러한 방법은 계속해서 동작한다. 이들이 상이한 샘플 주기들에 있으면, 2개의 상이한 방법들이 사용될 수 있다. 제1 방법은 각 데이터 쌍에 대해 동일하지 않더라도, 각 데이터 포인트에 대한 보다 최적화된 샘플 위치를 선택하는 것이다. 그리고 나서, 호스트는 데이터 쌍들 세트(타입 2에 대한 2비트, 타입 3에 대한 4비트, 타입 4에 대한 8비트)로부터 비트들에 대한 모든 샘플링 후에 데이터 스트림을 재구성할 수 있다. 다른 옵션은 매 데이터 쌍이 동일한 클록 가장자리에서 샘플링될 수 있도록 역방향 레이트 디바이저를 호스트가 증가시키는 것이다.

XIII. 링크 지연 및 스큐의 효과들

MDDI_Data 쌍들 및 MDDI_Stb 사이의 순방향 링크 상의 지연 스큐는 지연 스큐 보상이 사용되지 않는 경우, 최대 가능 데이터 레이트를 제한할 수 있다. 타이밍 스큐를 초래하는 지연에서의 이러한 차이들은 아래에서 설명되는 제어기 로직, 라인 드라이버들 및 수신기들, 및 케이블 및 커넥터들에 기인한다.

A. 스큐에 의해 제한되는 링크 타이밍 분석(MDDI 타입-1)

1. 타입 1 링크에 대한 지연 및 스큐 예

도41에 제시된 것과 유사하며, 타입 1 인터페이스 링크를 수용하는 전형적인 인터페이스는 도57에 제시된다. 도57에서, 전파 지연 및 스큐에 대한 예시적인 또는 전형적인 값들이 MDDI 타입 1 순방향 링크에 대한 수개의 처리 또는 인터페이스 단들 각각에 대해 제시된다. MDDI_Stb 및 MDDI_Data0 사이의 지연에서의 스큐는 출력 클록의 듀티 사이클을 왜곡되게 한다. 플립플롭(5728, 5732)을 사용하는 수신기 플립플롭(RXFF) 단의 D 입력에서의 데이터는, 신뢰성있게 샘플링될 수 있도록 하기 위해서, 클록 가장자리 후에 약간 변경된다. 도면은 이러한 타이밍 관계식을 생성하는데 있어서 2개의 상이한 문제점들을 해결하기 위해서 사용되는 2개의 직렬 연결된 지연 라인들(5732a, 5732b)을 보여준다. 실제 구현에서, 이들은 하나의 지연 엘리먼트로 결합될 수 있다.

이러한 인터페이스를 통한, 예시적인 신호 처리를 위한 타입 1 링크의 데이터, Stb, 및 클록 복원 타이밍은 도58에서 제시된다.

상당한 총 지연 스큐는 일반적으로 다음 단계에서의 스큐의 총합에서 비롯된다: 플립플롭(5704,5706)을 갖는 송신기 플립플롭(TXFF); 드라이버(5708,5710)를 갖는 송신기 드라이버(TXDRVR), 수신기(5722,5724)를 갖는 수신기 라인 수신기(RXRCVR); 및 수신기 XOR 로직(RXXOR). 지연 1(5732a)는 다음 관계식에 의해 결정되는 RXXOR 단의 XOR 게이트(5736)의 지연과 매칭되거나 또는, 이러한 지연을 초과하여야 한다:

$$t_{PD-\min}(\text{Delay } 1) \geq t_{PD-\max}(\text{XOR})$$

수신기 플립플롭(5728,5732)의 D 입력이 그 클럭 입력 전에 변경되지 않도록 하기 위해서, 이러한 요구조건을 만족시키는 것이 바람직하다. 이는 RXFF의 홀드 타임이 0인 경우에 유효하다.

지연 2의 목적 또는 기능은 다음 관계식에 따라 RXFF 플립플롭의 홀드 타임을 보상하는 것이다:

$$t_{PD-\min}(\text{Delay } 2) = t_{H}(\text{RXFF})$$

많은 시스템에서, 홀드 타임은 0이기 때문에 이는 0이 될 것이며, 물론, 그 경우에서 지연 2의 최대 지연 역시 0이 될 수 있다.

수신기 XOR 단에서 스큐에 대한 최악의 기여는 늦은-데이터/이른-스트로브(data-late/strobe-early)인 경우이고, 여기서 지연 1은 최대 값에 존재하며, XOR 게이트로부터의 클럭 출력은 다음 관계식에 따라 가능한 한 일찍 오게 된다:

$$t_{SKEW-\max}(\text{RXXOR}) = t_{PD-\max}(\text{Delay } 1) - t_{PD-\min}(\text{XOR})$$

이러한 상황에서, 데이터는 그 시간에서 매우 근접한, 2개의 비트 주기들, n 및 n+1 사이에서 변경될 수 있고, 여기서 비트 n+1은 수신기 플립플롭 내에서 클럭된다.

MDDI 타입 1 링크의 최대 데이터 레이트(최소 비트 주기)는 MDDI 링크에서 모든 드라이버들, 케이블, 및 수신기들을 통해 직면하는 최대 스큐에 RXXFF 단의 총 데이터 셋업을 더한 함수이다. RXRCVR 단의 출력까지의 링크에서의 총 지연 스큐는 다음과 같이 표현될 수 있다:

$$t_{SKEW-\max}(\text{LINK}) = t_{SKEW-\max}(\text{TXFF}) + t_{SKEW-\max}(\text{TXDRVR}) + t_{SKEW-\max}(\text{CABLE}) + t_{SKEW-\max}(\text{RXRCVR})$$

그리고, 최소 비트 주기는 다음과 같이 주어진다:

$$t_{BIT-\min} = t_{SKEW-\max}(\text{LINK}) + 2 \cdot t_{B-TP4} + t_{Asymmetry} + t_{SKEW-\max}(\text{RXXOR}) + t_{jitter-host} + t_{PD-\max}(\text{Delay } 2) + t_{SU}(\text{RXFF})$$

외부 모드를 위해 도57에 제시된 예에서, $t_{SKEW-\max}(\text{LINK}) = 1000 \text{ psec}$ 이고, 최소 비트 주기는 다음과 같이 표현될 수 있다:

$$t_{BIT-\min} = 1000 + 2 \cdot 125 + 625 + 125 + 200 + 0 + 100 = 2300 \text{ psec},$$

또는 대략 434Mbps로 간주될 수 있다.

B. MDDI 타입 2, 3, 및 4에 대한 링크 타이밍 분석

도41 및 57과 유사하며, 타입 2, III, 및 IV 인터페이스 링크들을 수용하는, 전형적인 인터페이스 회로는 도59에 제시된다. 추가적인 신호 처리를 수용하기 위해서, 추가적인 엘리먼트들이 TXFF(5904), TXDRVR(5908), RXRCVCR(5922), 및 RXFF(5932,5928,5930) 단들에서 사용된다. 도59에서, 전파 지연 및 스큐에 대한 예시적인 또는 전형적인 값들이 MDDI 타입 2 순방향 링크의 수개의 처리 또는 인터페이스 단들 각각에 대해서 제시된다. 출력 클록의 듀티 사이클에 영향을 미치는 MDDL_Stb 및 MDDL_Data0 사이의 지연에서의 스큐 뿐만 아니라, 이러한 2개의 신호들 모두 및 다른 MDDL_Data 신호들 사이에 스큐가 존재한다. 플립플롭(5928,5930)들로 구성되는 수신기 플립플롭 B(RXFFB)의 D 입력에서의 데이터는, 신뢰성있게 샘플링될 수 있도록 하기 위해서, 클록 가장자리 후에 약간 변경된다. MDDL_Data가 MDDL_Data0 또는 MDDL_Stb 보다 일찍 도달하면, MDDL_Data1은 적어도 지연 스큐량만큼 샘플링되는데 있어서 지연되어야 한다. 이를 위해서, 데이터는 지연 3 지연 라인을 사용하여 지연된다. MDDL_Data1이 MDDL_Stb 및 MDDL_Data0 보다 늦게 도달하고 지연 3에 의해 지연되면, MDDL_Data1이 변경되는 지점은 다음 클록 가장자리에 보다 근접하도록 이동된다. 이러한 처리는 MDDI 타입 2, 3, 또는 4 링크에 대한 데이터 레이트의 상한을 결정한다. 서로에 대한 2개의 데이터 신호들 및 MDDL_Stb의 타이밍 또는 스큐 관계식에 대한 예시적인 상이한 가능성들이 도60A, 60B, 및 60C에 제시된다.

MDDL_DataX 가 가능한 한 일찍 도달할 때, RXFFB에서 신뢰성있게 데이터를 샘플링하기 위해서, 지연 3은 다음 관계식에 따라 설정된다:

$$t_{PD-\min}(\text{Delay3}) \geq t_{SKEW-\max}(\text{LINK}) + t_{H}(\text{RXFFB}) + t_{PD-\max}(\text{XOR})$$

최대 링크 속도는 최소 허용가능 비트 주기에 의해 결정된다. 이는 MDDL_DataX가 가능한 한 늦게 도달하는 경우에 가장 많은 영향을 받는다. 이러한 경우, 최대 허용가능한 사이클 타임은 다음과 같이 주어진다:

$$t_{BIT-\min} = t_{SKEW-\max}(\text{LINK}) + t_{PD-\max}(\text{Delay3}) + t_{SU}(\text{RXFFB}) - t_{PD-\min}(\text{XOR})$$

링크 속도의 상한은 다음과 같다:

$$t_{PD-\max}(\text{Delay3}) = t_{PD-\min}(\text{Delay3})$$

그리고 다음과 같은 가정이 주어진다:

$$t_{BIT-\min}(\text{lower-bound}) = 2 \cdot t_{SKEW-\max}(\text{LINK}) + t_{PD-\max}(\text{XOR}) + t_{SU}(\text{RXFFB}) + t_{H}(\text{RXFFB})$$

상기 예에서, 최소 비트 주기에 대한 하한은 다음 관계식에 의해 주어진다:

$$t_{BIT-\min}(\text{lower-bound}) = 2 \cdot (1000 + 2 \cdot 125 + 625 + 200) + 1500 + 100 + 0 = 5750 \text{ psec},$$

그리고 이는 대략 174Mbps이다.

이는 타입 1 링크에서 사용될 수 있는 최대 데이터 레이트 보다 훨씬 느린 레이트이다. MDDI의 자동 지연 스큐 보상 기능은 최대 링크 레이트 인자에서 갖는 지연 스큐가 유효 데이터 셋업의 가장자리에 바로 존재하도록 하는 효과를 상당히 감소시킨다. MDDL_Data0 및 MDDL_Stb 사이의 조정된 스큐 및 최소 비트 주기는 다음과 같다:

$$t_{BIT-\min-\text{Calibrated}} = t_{SKEW-\max}(\text{Calibrated}) + 2 \cdot t_{B-TP4} + t_{Asymmetry} + t_{jitter-host} + t_{SKEW-\max}(\text{RXAND+RX XOR}) + t_{SU}(\text{RX})$$

도8-5에 제시된 예에서, $t_{SKEW-\max}(\text{Data0-Stb-Calibrated}) = 300 \text{ psec}$ 이고, 최소 비트 주기는 다음과 같다:

$t_{BIT-min-Calibrated} = 300 + 2 \cdot 125 + 625 + 200 + 175 + 100 = 1650 psec$, 그리고, 이는 대략 606Mbps이다.

MDDI_Data가 가능한 한 일찍 도달하는 경우, RXFFB에서 데이터를 신뢰성있게 샘플링하기 위해서, 관련된 프로그램가능한 지연은 1 탭의 정확도로 최적 세팅으로 조정되며, 추가적인 탭 지연이 안전성을 위해 추가된다. 최대 링크 속도는 최소 허용가능한 비트 주기에 의해 결정된다. 이는, MDDI_Data1 이 가능한 한 늦게 도달하는 경우에, 가장 영향을 많이 받는다. 이 경우, 최소 허용가능한 사이클 타임은 다음과 같다:

$$t_{BIT-min-Data1-Calibrated} = 2 \cdot t_{TAP-Spacing-max} + 2 \cdot t_{TA-TP4}$$

도8-5에 주어진 예에서, 샘플링 MDDI_Data에 기반하는, 최소 비트 주기의 하한은 다음과 같다:

$$t_{BIT-min-Data1-Calibrated} = 2 \cdot 150 + 2 \cdot 125 = 550 psec$$

XIV. 물리 계층 상호연결

본 발명에 따른 인터페이스를 구현하는데 유용한 물리적 연결들은 클라이언트 장치 측에서 Hirose Electronic Company Ltd에 의해 제조되는 부품번호 3240-8P-C 및, 호스트측에서 Hirose Electronic Company Ltd에서 제조되는 부품번호 3260-8S(01)과 같은 상업적으로 이용가능한 부품들을 사용하여 구현될 수 있다. 타입 1/타입 2 인터페이스들에서 사용되는 커넥터들에 대한 예시적인 인터페이스 핀 할당 또는 "핀아웃(pinout)"은 테이블 XIII에 리스트되어 있고, 도61에서 제시된다.

[표 13]

신호 이름	핀번호	색상	신호 이름	핀번호	색상
MDDI_Pwr	1	White	MDDI_Gnd	2	Black paired w/White
MDDI_Stb+	3	Green	MDDI_Stb-	4	Black paired w/Green
MDDI_Data0+	7	Blue	MDDI_Data0-	8	Black paired w/Blue
MDDI_Data1+	11	Brown	MDDI_Data1-	12	Black paired w/Brn
MDDI_Data2+	15	Red	MDDI_Data2-	16	Black paired w/Red
MDDI_Data3+	19	Orange	MDDI_Data3-	20	Black paired w/Org
MDDI_Data4+	17	TBD1	MDDI_Data4-	18	Black paired w/TBD1
MDDI_Data5+	13	TBD2	MDDI_Data5-	14	Black paired w/TBD2
MDDI_Data6+	9	TBD3	MDDI_Data6-	10	Black paired w/TBD3
MDDI_Data7+	5	TBD4	MDDI_Data7-	6	Black paired w/TBD4
			Shield		

셸드는 호스트 인터페이스의 HOST_Gnd에 연결되고, 케이블의 셸드 드레인 와이어는 클라이언트 커넥터의 셸드에 연결된다. 그러나, 셸드 및 드레인 와이어는 클라이언트 내부의 회로 그라운드에 연결되지 않는다.

상호연결 엘리먼트들 및 장치들은, PDA 및 무선 전화기, 또는 휴대용 게임 장치와 같은 이동용 통신 및 컴퓨팅 장치에서 사용될 수 있을 만큼 충분히 작게 설계 및 선택된다. 임의의 커넥터 및 케이블링은 일반적인 소비자 환경에서 사용하기에 충분할 만큼 내구성이 있어야 하고, 특히 케이블링에 있어서 작은 사이즈를 허용하여야 하며, 비용이 저렴하여야 한다. 전달 엘리먼트들은 타입 1 및 타입 2에 대해 최대 450 Mbps, 그리고 8비트 병렬 타입 4 버전에서 최대 3.6Gbps의 전달 레이트를 갖는 차분 NRZ 데이터인, 스트로브 신호들 및 데이터를 수용하여야 한다.

내부 모드 애플리케이션에서, 사용되는 컨덕터들에 대한 동일한 의미에서의 커넥터가 존재하지 않거나, 이러한 연결 엘리먼트들은 매우 소형화되는 경향이 있다. 하나의 예는, 호스트 또는 클라이언트 장치 중 하나를 하우징하는 집적 회로 또는 엘리먼트들을 수신하는 제로 삽입 포스 "소켓들"이다. 또 다른 예에서, 다양한 상호연결 컨덕터들과 함께 호스트 및 클라이언트가 인쇄 회로 기판에 존재하고, 집적 회로들의 상호접속을 위한 컨덕터들과의 접촉부와 납땜되어지는 하우징들로부터 연장하는 접촉부 또는 "핀들"을 갖는다.

XV. 동작

도 55에서 패킷들을 처리하는 인터페이스 장치에 대한 개관과 함께, 본 발명의 실시예들을 사용하는 인터페이스의 동작 기간 동안 데이터 및 패킷들을 처리하는 일반적인 단계들의 요약이 도 54A 및 54B에 제시된다. 이러한 도면들에서, 처리는 통신 경로(여기서 케이블)를 사용하여 클라이언트 및 호스트가 연결되는지 여부를 결정하는 단계(5402)에서 시작한다. 이는 커넥터 또는 케이블의 존재, 또는 호스트(USB 인터페이스로서 제시됨)로의 입력에서의 신호들의 존재를 검출하는 소프트웨어 또는 하드웨어 등을 사용하여, 호스트에 의한 주기적인 폴링의 사용을 통해 이뤄질 수 있다. 호스트에 연결되는 클라이언트가 존재하지 않으면, 애플리케이션에 따라 소정 길이를 갖는 대기 상태로 진입하여, 동면 상태로 진행하거나 또는 호스트를 재활성화하기 위해 사용자에게 어떠한 액션을 요구하는 차후 사용을 대기하기 위해서 비활성화된다. 예를 들어, 호스트가 컴퓨터 타입 장치에 상주하는 경우, 클라이언트를 찾기 위해서 사용자는 호스트 처리를 활성화하는 프로그램을 요청하거나, 스크린 아이콘을 클릭하여야 한다. 또 다시, USB 타입 연결의 간단한 플러그 인은 호스트의 구성 및 기능 또는 상주하는 호스트 프로그램에 따라 호스트 처리를 활성화할 수 있다.

클라이언트가 호스트에 연결되거나, 또는 호스트가 클라이언트에 연결되거나, 또는 존재하는 것으로 검출되면, 클라이언트 또는 호스트 중 하나는 단계(5404 및 5406)에서 서비스를 요청하는 적절한 패킷들을 전송한다. 클라이언트는 단계(5404)에서 클라이언트 서비스 요청 또는 상태 패킷 중 하나를 전송할 수 있다. 상술한 바와 같이, 링크는 이전에 폐쇄되었거나 동면 상태일 수 있고, 따라서 이는 뒤이은 통신 링크의 완전한 초기화가 아닐 수 있음을 주의하여야 한다. 통신 링크가 동기화되고 호스트가 클라이언트와의 통신을 시도하면, 클라이언트는 단계(5408)에서 클라이언트 기능 패킷을 호스트로 제공한다. 호스트는 이제 전송 레이트를 포함하는 지원 타입을 결정할 수 있고, 클라이언트는 수용될 수 있다.

일반적으로, 호스트 및 클라이언트는 사용될 서비스 모드 타입(레이트/속도), 예를 들면 타입 1, 타입 2 등을 단계(5410)에서 협상한다. 서비스 타입이 설정되면, 호스트는 정보 전달을 개시할 수 있다. 또한, 호스트는 단계(5411)에서 제시되는 바와 같이, 다른 신호 처리와 병렬적으로 통신 링크들의 타이밍을 최적화하기 위해서 왕복 지연 측정 패킷을 사용할 수 있다.

전술한 바와 같이, 모든 전송들은 서브-프레임 헤더 패킷에서 시작하고(단계 5412에서 전달되는 것을 제시됨), 뒤이어 데이터 타입, 비디오 및 오디오 스트림 패킷, 및 채움 패킷이 뒤따른다(단계 5414에서 전달되는 것으로 제시됨). 오디오 및 비디오 데이터는 이전에 준비되었거나 또는 패킷들로 매핑되었고, 채움 패킷들은 필요에 따라 삽입되어 미디어 프레임들의 필요한 수의 비트들을 채운다. 호스트는 사운드 장치들을 활성화하기 위해서 순방향 오디오 채널 인에이블 패킷과 같은 패킷들을 전송할 수 있다. 또한, 호스트는 전술한 다른 패킷 타입들을 사용하여 명령 및 정보를 전달할 수 있고, 여기서 컬러 맵, 비트 블록 전달 또는 다른 패킷들의 전달로 단계(5416)에서 제시된다. 또한, 호스트 및 클라이언트는 적절한 패킷들을 사용하여 키보드 또는 포인팅 장치들과 관련된 데이터를 교환할 수 있다.

동작 기간 동안, 상이한 데이터 레이트 또는 인터페이스 모드 타입을 요구하는 호스트 또는 클라이언트를 초래하는, 수개의 상이한 이벤트들 중 하나가 발생할 수 있다. 예를 들어, 컴퓨터 또는 데이터를 통신하는 다른 장치는 패킷의 제공 또는 준비에서 속도 저하를 초래하는 처리 데이터에서의 로딩 상태들을 직면할 수 있다. 데이터를 수신하는 클라이언트 장치는 전용 AC 전력 소스로부터 보다 제한된 배터리 전력 소스로 변경될 수 있고, 빨리 데이터를 전달할 수 없거나, 즉시 명령들을 처리할 수 없거나, 또는 보다 제한된 전력 세팅들 하에서 동일한 정도의 해상도 또는 컬러 깊이를 사용하지 못할 수 있다. 대안적으로, 제한적인 상태는 경감 또는 사라져서, 장치가 보다 높은 레이트에서 데이터를 전달할 수 있도록 할 수 있다. 이는 보다 바람직하고, 보다 높은 전송 레이트 모드로 변경하기 위해서 요청이 이뤄질 수 있다.

알려진 상태들의 이러한 또는 다른 타입들이 발생 또는 변경될 수 있으면, 호스트 또는 클라이언트는 이들을 탐지하여 인터페이스 모드를 재협상할 수 있다. 이는 단계(5420)에서 제시되고, 여기서 호스트는 다른 모드로의 핸드오프를 요청하는 클라이언트로 인터페이스 타입 핸드오프 요청 패킷들을 전송하고, 클라이언트는 변경이 추구됨을 확인하는 인터페이스 타입 확인 패킷을 전송하며, 호스트는 특정 모드로 변경하도록 하는 수행 타입 핸드오프 패킷을 전송한다.

비록 처리에 대한 특별한 명령을 요구하지는 않지만, 클라이언트 및 호스트는 포인팅 장치, 키보드, 또는 클라이언트와 주로 관련되는 다른 사용자 타입 입력 장치(비록 이러한 엘리먼트들이 호스트측에 존재하더라도)로부터 수신되는, 또는 이들로 향하는 데이터와 관련된 패킷들을 교환할 수 있다. 이러한 패킷들은 상태 머신(5502)이 아니라, 일반적으로 범용 프로세서 타입 엘리먼트에 의해 처리된다. 또한, 전술한 명령들 중 일부는 범용 프로세서(5504, 5508)에 의해 처리된다.

호스트 및 클라이언트 사이에 데이터 및 명령들이 교환된 후에, 어떤 포인트에서, 추가적인 데이터가 전달될 것인지 또는 호스트 또는 클라이언트가 전달 서비스를 중단할 것인지에 대한 결정이 이뤄진다. 이는 단계(5422)에서 제시된다. 링크가 동면 상태로 진입하거나, 완전히 폐쇄되면, 호스트는 클라이언트로 링크 폐쇄 패킷을 전송하고, 양측은 데이터 전달을 종료한다.

전술한 동작 프로세싱에서 전달되는 패킷들은 호스트 및 클라이언트 제어기들과 관련하여 앞에서 설명된 드라이버들 및 수신기들을 사용하여 전달될 것이다. 이러한 라인 드라이버들 및 다른 논리 엘리먼트들은 전술한 상태 머신 및 범용 프로세서에 연결되고, 이는 도55에서 제시된다. 도55에서, 상태 머신(5502) 및 범용 프로세서(5504 및 5508)들은 전용 USB 인터페이스, 메모리 엘리먼트, 또는 링크 제어기 외부에 상주하는 다른 컴포넌트들과 같이 도시되지 않는 다른 엘리먼트들에 추가로 연결될 수 있다.

프로세서들 및 상태 머신은 통신 링크의 효율적인 설정 및 종료, 및 패킷들의 전달을 보장하기 위해서, 보호 시간 등과 관련하여 전술한 드라이버들의 인에이블링 및 디스에이블링에 대한 제어를 제공한다.

XVI. 디스플레이 프레임 버퍼들

비디오 데이터 버퍼링 요구조건들은 컴퓨터 그래픽들과 비교하여 동영상 비디오 이미지에 대해서는 다르다. 화소 데이터는 클라이언트의 로컬 프레임 버퍼에 가장 빈번히 저장되어, 클라이언트에 대한 이미지는 로컬적으로 리프레쉬될 수 있다.

풀-모션 비디오가 디스플레이되면(각 미디어 프레임은 디스플레이의 거의 매 화소에서 변경됨), 디스플레이 상의 이미지가 제2 프레임 버퍼로부터 리프레쉬되면서, 하나의 프레임 버퍼에서 인입 화소 데이터를 저장하는 것이 바람직하다. 2개를 초과하는 디스플레이 버퍼들이 아래에서 설명되는 바와 같이 시각적인 가공물들을 제거하기 위해서 사용될 수 있다. 하나의 프레임 버퍼에서 전체 이미지가 수신되었다면, 버퍼들의 역할들은 교환될 수 있고, 그리고 나서 새롭게 수신된 이미지가 디스플레이를 리프레쉬하기 위해서 허용되고, 다른 버퍼는 이미지의 다음 프레임으로 채워진다. 이러한 개념은 도 91A에 제시되어 있고, 여기서 화소 데이터는 디스플레이 업데이트 비트를 "01"로 설정함으로써 오프라인 이미지 버퍼에 기록된다.

다른 애플리케이션들에서, 호스트는 전체 이미지를 리페인트(repaint)할 필요 없이 이미지의 일부분만을 갱신하면 된다. 이러한 상황에서, 도91B에 제시된 바와 같이, 디스플레이를 리프레쉬하기 위해서 사용되는 버퍼로 직접 새로운 화소를 기록하는 것이 바람직하다.

작은 비디오 윈도우를 구비한 고정된 이미지를 갖는 애플리케이션에서, 도91에 제시된 바와 같이, 이들 버퍼 모두에 고정된 이미지를 기록하고(디스플레이 업데이트 비트는 "11"), 뒤이어 디스플레이 업데이트 비트를 "01"로 설정함으로써 오프라인 버퍼에 동영상 이미지의 화소를 기록하는 것이 가장 쉽다.

다음 규칙들은 동시에 새로운 정보를 기록하고 디스플레이를 리프레쉬하면서, 버퍼 포인터들에 대한 유용한 조작을 설명한다. 3개의 버퍼 포인터들이 존재한다: `current_fill`은 MDDI 링크 상에서 데이터로부터 현재 채워지는 버퍼를 지시한다; `just_filled`는 가장 최근에 채워진 버퍼를 지시한다; `being_displayed`는 디스플레이들을 리프레쉬하기 위해서 현재 사용되는 버퍼를 지시한다. 이러한 모든 3개의 버퍼 포인터들은 0 내지 $N-1$ 까지의 값들을 포함하며, 여기서 N 은 디스플레이 버퍼들의 수이고, $N \geq 2$ 이다. 버퍼 포인터들에 대한 산수(arithmetic)는 mod N 이고, 예를 들어 $N=3$ 이고 `current_fill=2`이면, 증분 `current_fill`은 `current_fill`이 0으로 설정되게 한다. $N=2$ 인 간단한 경우에서, `just_filled`은 항상 `current_fill`의 보수이다. 매 MDDI에서, 미디어 프레임 경계(0과 같은 서브-프레임 카운트 필드를 갖는 서브-프레임 헤더 패킷)는 규정된 순서로 다음 연산을 수행한다: `just_filled`를 `current_fill`과 동일하게 설정하고, `current_fill`을 `current_fill+1`과 동일하게 설정.

MDDI 비디오 스트림 패킷들은 다음과 같은 방법론 또는 구조에 따라 버퍼들을 업데이트한다: 디스플레이 업데이트 비트들이 "01"이면, 화소 데이터는 `current_fill`에 의해 규정된 버퍼로 기록됨; 디스플레이 업데이트 비트들이 "00"이면, 화소 데이터는 `just_filled`에 의해 규정된 버퍼로 기록됨; 디스플레이 업데이트 비트들이 "11"이면, 모든 버퍼들에 화소 데이터

가 기록됨. 디스플레이는 being_displayed 포인터에 의해 규정된 버퍼로부터 리프레쉬된다. 1 프레임 리프레쉬 에포크(epoch)의 최종 화소를 디스플레이가 리프레쉬한 후에, 그리고 다음 프레임 리프레쉬 에포크의 제1 화소를 리프레쉬하기 전에, 디스플레이 업데이트 처리는 being_refreshed를 just_filled와 동일하게 설정하는 동작을 수행한다.

비디오 스트림 패킷은 화소 데이터가 기록되는 프레임 버퍼를 규정하는 한 쌍의 디스플레이 업데이트 비트들을 포함한다. 클라이언트 기능 패킷은 디스플레이 업데이트 비트들의 어떤 조합이 클라이언트에서 지원되는지를 표시하는 3개의 추가적인 비트들을 갖는다. 많은 경우에서, 컴퓨터에 의해 생성되는 이미지들은 사용자 입력에 따라, 또는 컴퓨터 네트워크로부터 수신되는 정보에 따라 증분적으로 업데이트될 필요가 있다. 디스플레이 업데이트 비트 조합 "00" 및 "11"은 디스플레이되는 프레임 버퍼로 또는 프레임 버퍼 둘 모두로 화소 데이터가 기록되도록 함으로써, 이러한 동작 모드를 지원한다.

비디오 이미지들을 수용할 때, 도 92는 "01"과 동일한 디스플레이 업데이트 비트들을 가지고 MDDI 링크 상에서 비디오 데이터가 전송되는 경우, 프레임 버퍼들을 사용하여 비디오 이미지들이 어떻게 디스플레이되는지를 보여준다. MDDI 링크에서 미디어-프레임 경계가 검출된 후에, 현재 리프레쉬되는 프레임에 대한 리프레쉬 처리가 완료될 때, 디스플레이 리프레쉬 처리는 다음 프레임 버퍼로부터 리프레쉬를 시작한다.

도 92와 관련되는 중요한 가정은 디스플레이를 리프레쉬하기 위해서 프레임 버퍼로부터 클라이언트가 화소들을 판독하는데 사용되는 것과 동일한 순서로(일반적으로, 로우 단위로, 스크린의 좌측 상부에서 우측 하부로 판독됨) 전송되는 연속적인 화소들의 스트림으로서, 이미지가 호스트로부터 수신된다는 것이다. 이는 디스플레이 리프레쉬 및 이미지 전달 동작들이 동일한 프레임 버퍼를 참조하는 경우에 중요한 내용이다.

부분적인 이미지들을 디스플레이하는 것을 방지하기 위해서, 디스플레이 리프레쉬 프레임 레이트가 이미지 전송 프레임 레이트 보다 크게 하는 것이 필요하다. 도 93은 이미지 전송 보다 디스플레이 리프레쉬 레이트가 느린 경우, 이미지 프레임테이션이 어떻게 발생하는지를 보여준다.

컴퓨터 그래픽 이미지들 및 동영상 비디오의 조합을 포함하는 이미지에서, 비디오 화소 데이터는 미디어-프레임의 작은 부분을 점유한다. 이는 디스플레이 리프레쉬 동작 및 이미지 전달이 동일한 프레임 버퍼를 참조하는 경우에 중요할 수 있다. 이러한 상황은 도 94에서 교차된 평행선 음영부로 제시되고, 여기서 디스플레이를 리프레쉬하기 위해서 버퍼로부터 판독되는 화소들은 2 프레임 이전에 버퍼에 기록된 화소들이거나, 또는 이들은 동일한 프레임 버퍼에 즉시 기록된 프레임에 대응한다.

클라이언트에서 3개의 프레임 버퍼들의 사용은 도 95에 제시된 바와 같이, 프레임 버퍼에 대한 액세스를 위한 경쟁의 작은 윈도우 문제를 해결할 것이다.

그러나, 도 96에 제시된 바와 같이, MDDI 링크 상에서 디스플레이 리프레쉬 레이트가 미디어-프레임보다 작으면, 여전히 문제점이 존재한다.

동영상 비디오 이미지에 대한 하나의 버퍼의 사용은 도 97에 제시된 바와 같이 다소 문제가 있다. 디스플레이 리프레쉬가 버퍼로의 이미지 전달보다 빠르면, 리프레쉬되는 이미지는 종종 기록되는 프레임의 상부(upper portion)를 보여줄 것이고, 이미지의 하부(lower portion)는 이전에 전달된 프레임일 것이다. 디스플레이 리프레쉬가 이미지 전달보다 빠르면(선호되는 동작 모드), 유사한 스플릿된 이미지를 보여주는 프레임들의 보다 빈번한 인스턴스들이 존재할 것이다.

XVII. 지연 값 테이블

패킷 처리 지연 파라미터 패킷들은 클라이언트에서의 임의의 명령들을 처리하기 위해서, 예상 지연을 계산하는 테이블-lookup 함수를 사용한다. 테이블에서의 값들은 지연 값의 매운 넓은 동적 범위를 제공하기 위해서 로그 형태로 증가한다. 본 발명의 실시예를 구현하는데 유용한 지연값들의 예시적인 테이블은, 대응하는 3인덱스 값 대 지연 값들을 갖는 하기 표 XX에 제시된다.

[표 XX]

0 - no_delay	37 - 1.5ns	74 - 51ns	111 - 1.8us	148 - 62us	185 - 2.2ms	222 - 75ms
1 - 46ps	38 - 1.6ns	75 - 56ns	112 - 2.0us	149 - 68us	186 - 2.4ms	223 - 83ms
2 - 51ps	39 - 1.8ns	76 - 62ns	113 - 2.2us	150 - 75us	187 - 2.6ms	224 - 91ms
3 - 56ps	40 - 2.0ns	77 - 68ns	114 - 2.4us	151 - 83us	188 - 2.9ms	225 - 100ms
4 - 62ps	41 - 2.2ns	78 - 75ns	115 - 2.6us	152 - 91us	189 - 3.2ms	226 - 110ms
5 - 68ps	42 - 2.4ns	79 - 83ns	116 - 2.9us	153 - 100us	190 - 3.5ms	227 - 120ms
6 - 75ps	43 - 2.6ns	80 - 91ns	117 - 3.2us	154 - 110us	191 - 3.8ms	228 - 130ms
7 - 83ps	44 - 2.9ns	81 - 100ns	118 - 3.5us	155 - 120us	192 - 4.2ms	229 - 150ms
8 - 91ps	45 - 3.2ns	82 - 110ns	119 - 3.8us	156 - 130us	193 - 4.6ms	230 - 160ms
9 - 100ps	46 - 3.5ns	83 - 120ns	120 - 4.2us	157 - 150us	194 - 5.1ms	231 - 180ms
10 - 110ps	47 - 3.8ns	84 - 130ns	121 - 4.6us	158 - 160us	195 - 5.6ms	232 - 200ms
11 - 120ps	48 - 4.2ns	85 - 150ns	122 - 5.1us	159 - 180us	196 - 6.2ms	233 - 220ms
12 - 130ps	49 - 4.6ns	86 - 160ns	123 - 5.6us	160 - 200us	197 - 6.8ms	234 - 240ms
13 - 150ps	50 - 5.1ns	87 - 180ns	124 - 6.2us	161 - 220us	198 - 7.5ms	235 - 260ms
14 - 160ps	51 - 5.6ns	88 - 200ns	125 - 6.8us	162 - 240us	199 - 8.3ms	236 - 290ms
15 - 180ps	52 - 6.2ns	89 - 220ns	126 - 7.5us	163 - 260us	200 - 9.1ms	237 - 320ms
16 - 200ps	53 - 6.8ns	90 - 240ns	127 - 8.3us	164 - 290us	201 - 10ms	238 - 350ms
17 - 220ps	54 - 7.5ns	91 - 260ns	128 - 9.1us	165 - 320us	202 - 11ms	239 - 380ms
18 - 240ps	55 - 8.3ns	92 - 290ns	129 - 10us	166 - 350us	203 - 12ms	240 - 420ms
19 - 260ps	56 - 9.1ns	93 - 320ns	130 - 11us	167 - 380us	204 - 13ms	241 - 460ms
20 - 290ps	57 - 10ns	94 - 350ns	131 - 12us	168 - 420us	205 - 15ms	242 - 510ms
21 - 320ps	58 - 11ns	95 - 380ns	132 - 13us	169 - 460us	206 - 16ms	243 - 560ms
22 - 350ps	59 - 12ns	96 - 420ns	133 - 15us	170 - 510us	207 - 18ms	244 - 620ms
23 - 380ps	60 - 13ns	97 - 460ns	134 - 16us	171 - 560us	208 - 20ms	245 - 680ms
24 - 420ps	61 - 15ns	98 - 510ns	135 - 18us	172 - 620us	209 - 22ms	246 - 750ms
25 - 460ps	62 - 16ns	99 - 560ns	136 - 20us	173 - 680us	210 - 24ms	247 - 830ms
26 - 510ps	63 - 18ns	100 - 620ns	137 - 22us	174 - 750us	211 - 26ms	248 - 910ms
27 - 560ps	64 - 20ns	101 - 680ns	138 - 24us	175 - 830us	212 - 29ms	249 - 1.0sec
28 - 620ps	65 - 22ns	102 - 750ns	139 - 26us	176 - 910us	213 - 32ms	250 - 1.1sec
29 - 680ps	66 - 24ns	103 - 830ns	140 - 29us	177 - 1.0ms	214 - 35ms	251 - 1.2sec
30 - 750ps	67 - 26ns	104 - 910ns	141 - 32us	178 - 1.1ms	215 - 38ms	252 - 1.3sec
31 - 830ps	68 - 29ns	105 - 1.0us	142 - 35us	179 - 1.2ms	216 - 42ms	253 - 1.5sec
32 - 910ps	69 - 32ns	106 - 1.1us	143 - 38us	180 - 1.3ms	217 - 46ms	254 - 1.6s
33 - 1.0ns	70 - 35ns	107 - 1.2us	144 - 42us	181 - 1.5ms	218 - 51ms	255 - indefinite
34 - 1.1ns	71 - 38ns	108 - 1.3us	145 - 46us	182 - 1.6ms	219 - 56ms	
35 - 1.2ns	72 - 42ns	109 - 1.5us	146 - 51us	183 - 1.8ms	220 - 62ms	
36 - 1.3ns	73 - 46ns	110 - 1.6us	147 - 56us	184 - 2.0ms	221 - 68ms	

지연은 테이블 내에 인덱스로서 규정된 파라미터를 사용하여 테이블 룩업을 수행함으로써 계산된다. 이는 지연이 패킷처리 테이블(인덱스)과 동일하다는 것을 의미한다. 예를 들어, 지연 파라미터 리스트 아이템들로부터의 파라미터들 중 하나가 134와 동일한 8비트이면, 지연은 16μsec인 패킷처리테이블(134)과 동일하다. 값 255는 계산에 의해 결정될 수 없는 명령 완료를 표시하고, 호스트가 클라이언트 요청 및 상태 패킷 또는 MCCS VCP 제어 파라미터B7h의 그래픽 비지 플래그들을 검사할 것이라는 것을 표시한다.

일부 경우들에서, 이러한 지연은 높이, 폭, 또는 목적 이미지에서의 화소들의 수에 의해 곱해지고, 다른 지연들이 더해져서 전체 패킷 처리 지연을 계산한다.

XVIII. 다중 클라이언트 지원

현재 프로토콜 버전은 다수의 클라이언트 장치들을 직접 지원하는 것처럼 보이지 않는다. 그러나, 대부분의 패킷들은 다수의 클라이언트들을 갖는 시스템에서 특정 클라이언트 장치들을 어드레스하는데 사용될 수 있는 예비 클라이언트 ID 필드를 포함한다. 현재, 많은 수의 애플리케이션들에 있어서, 이러한 클라이언트 ID 또는 이러한 클라이언트 ID들은 0으로 설정된다. 서버-프레임 헤더 패킷은 또한 호스트가 다수의 클라이언트 시스템을 지원하는지 여부를 표시하기 위한 필드를 포함한다. 따라서, 다수의 클라이언트 호스트들 및 클라이언트들과의 차후 호환성을 시스템 설계자가 계획하는 것을 돕기 위해서, MDD 인터페이스 또는 프로토콜의 차후 애플리케이션들에서 다수의 클라이언트 장치들이 연결 또는 어드레스될 수 있도록 하는 방식이 존재한다.

다수의 클라이언트들을 갖는 시스템에서, 도98에 제시된 바와 같이, 허브를 사용하여 또는 데이터 체인을 사용하여, 또는 도99에 제시된 바와 같이 이러한 기술들의 조합을 사용하여, 클라이언트들이 호스트와 연결되는 것이 바람직하다.

XVIII. 추가 내용

본 발명의 실시예들에 대한 구성 및 프로토콜을 구현하기 위해서 사용되는 다양한 패킷들에 대해, 상술한 포맷들, 구조들, 및 내용들뿐만 아니라, 보다 상세한 필드 콘텐츠들 또는 동작들이 일부 패킷 타입들에 대해 여기서 제시된다. 이들은 당업자가 이러한 각각의 용법 또는 동작을 실행하고, 다양한 애플리케이션들에서 본 발명의 사용을 이해하는 것을 돕기 위해서 제시된다. 아직 논의되지 않은 약간의 필드들은 여기서 추가로 논의된다. 또한, 이러한 필드들에는 상술한 실시예들과 관련된 값들 및 예시적인 정의들이 제공된다. 그러나, 이러한 값들은 본 발명을 제한하는 것으로 간주되어서는 안 되며, 인터

페이스 및 프로토콜을 구현하는데 유용한 하나 이상의 실시예들을 나타내며, 모든 실시예들이 함께 또는 동시에 실시될 필요는 없다. 요구되는 데이터 또는 데이터 레이트 전달 결과들을 달성하기 위해서, 다른 값들이 다른 실시예들에서 사용될 수 있으며, 이는 당업자가 잘 이해할 수 있을 것이다.

A. 비디오 스트림 패킷들에 대해서

일 실시예에서, 화소 데이터 속성 필드(2바이트)는 다음과 같이 해석되는 일련의 비트 값들을 갖는다. 비트 1 및 0은 디스플레이 화소 데이터가 라우팅 어떻게 라우팅되는지를 선택한다. "11" 비트 값들에 있어서, 데이터는 양쪽 눈 모두에게 디스플레이되고, "10" 비트 값에 있어서, 데이터는 단지 왼쪽 눈으로만 라우팅되고, "01" 비트 값에 있어서, 데이터는 오른쪽 눈으로만 라우팅되며, 비트 값 "00"에 있어서, 데이터는 아래에서 논의되는 비트 8 내지 11에 규정되는 대안적인 디스플레이로 라우팅된다.

비트 2는 인터페이스 포맷에서 화소 데이터가 표현되는지를 표시하며, 값 "0"은 화소 데이터가 표준 프로그래시브 포맷으로 존재함을 의미하고, 로우 번호(화소 Y 좌표)는 하나의 로우에서 다음 로우로 진행할 때 1만큼 증분된다. 이러한 "1"의 값을 가질 때, 화소 데이터는 인터레이스 포맷으로 존재하고, 로우 번호는 하나의 로우에서 다음 로우로 진행할 때 2만큼 증분된다. 비트 3은 화소 데이터가 대안적인 화소 포맷으로 존재함으로 나타낸다. 이는 비트 2에 의해 인에이블되는 표준 인터레이스 모드와 유사하지만, 인터레이싱은 수평 대신에 수직이다. 비트 3이 "0"일 때, 화소 데이터는 표준 프로그래시브 포맷이며, 칼럼 번호(화소 X 좌표)는 각각의 연속적인 화소가 수신될 때 1만큼 증분된다. 비트 3이 "1"이면, 화소 데이터는 대안적인 화소 포맷이고, 칼럼 번호는 각 화소가 수신될 때마다 2만큼 증분된다.

비트 4는 화소 데이터가 디스플레이 또는 카메라와 관련되는지 여부를 표시하며, 여기서 데이터는 무선 전화기 또는 유사한 장치 또는 휴대용 컴퓨터, 또는 상술한 다른 장치를 위한 내부 디스플레이로부터 또는 내부 디스플레이로 전달되거나, 또는 데이터는 이러한 장치에 직접 연결되거나 이러한 장치 내부에 구축된 카메라로부터 또는 카메라로 전달된다. 비트 4가 "0"이면, 화소 데이터는 디스플레이 프레임 버퍼로부터 또는 디스플레이 프레임 버퍼로 전달된다. 비트 4가 "1"이면, 화소 데이터는 공지된 임의의 타입의 카메라 또는 비디오 장치로부터 또는 이러한 장치로 전달된다.

비트 5는 화소 데이터가 디스플레이에서 다음 연속적인 화소들의 로우를 포함할 때를 표시하기 위해서 사용된다. 이는 비트 5가 "1"로 설정되는 경우에 고려된다. 비트 5가 "1"로 설정되면, X 좌측 가장자리, Y 상부 가장자리, X 우측 가장자리, Y 하부 가장자리, X 시작, 및 Y 시작 파라미터들이 정의되지 않고, 클라이언트에 의해 무시된다. 비트 15가 논리 1 레벨에서 설정되면, 이는 이러한 패킷의 화소 데이터가 이미지의 최종 화소들 로우임을 표시한다. 클라이언트 기능 패킷의 클라이언트 특징 기능 필드 표시자 필드의 비트 8은 이러한 특징이 지원되는지 여부를 표시한다.

비트 7 및 6은 화소 데이터가 기록될 프레임 버퍼를 규정하는 디스플레이 업데이트 비트들이다. 보다 구체적인 효과들은 다른 곳에서 논의된다. 비트 값 "01"에서, 화소 데이터는 오프라인 이미지 버퍼에 기록된다. 비트 값 "00"에서, 화소 데이터는 디스플레이를 리프레쉬하기 위해서 사용되는 이미지 버퍼에 기록된다. 비트 값 "11"에서, 화소 데이터는 모든 이미지 버퍼들에 기록된다. 비트 값 또는 조합 "10"은 무효한 값 또는 지정(designation)으로 취급되고, 화소 데이터는 무시되며, 어떠한 이미지 버퍼들에도 기록되지 않는다. 이러한 값은 인터페이스의 추가적인 애플리케이션을 위한 사용을 가질 수 있다.

비트 8 내지 11은 화소 데이터가 라우팅되는 대체(alternate) 디스플레이 또는 디스플레이 위치를 규정하는 부호를 갖지 않는 4 비트 정수를 형성한다. 비트 0 및 1은 디스플레이 클라이언트가 비트 8 내지 11을 대체 디스플레이 번호로서 해석하기 위해서 "00"으로 설정된다. 비트 0 및 1이 "00"이 아니면, 비트 8 내지 11은 논리-제로 레벨로 설정된다.

비트 12 내지 14는 추가적인 사용을 위해 예비되고, 일반적으로 논리 제로 레벨로 설정된다. 비트 15는, 상술한 바와 같이, 비트 5와 연결되어 사용되고, 비트 15를 논리 1로 설정하는 것은 화소 데이터 필드의 로우가 데이터 프레임 화소들의 최종 로우임을 표시한다. 논리 1로 설정되는 비트 5를 갖는 다음 비디오 스트림 패킷은 다음 비디오 프레임의 제1 로우에 대응할 것이다.

2 바이트 X 시작 및 Y 시작 필드는 화소 데이터 필드의 제1 화소에 대한 포인트의 절대치 X 및 Y 좌표(X 시작, Y 시작)를 규정한다. 2 바이트 X 좌측 가장자리 및 Y 상부 가장자리 필드들은 화소 데이터 필드에 의해 채워지는 스크린 윈도우의 좌측 가장자리의 X 좌표 및 상부 가장자리의 Y 좌표를 규정하고, X 우측 가장자리 및 Y 하부 가장자리 필드들은 업데이트되는 윈도우의 우측 가장자리 X 좌표 및 하부 가장자리 Y 좌표를 규정한다.

화소 카운트 필드(2 바이트)는 아래의 화소 데이터 필드의 화소들의 수를 규정한다.

파라미터 CRC 필드(2 바이트)는 패킷 길이로부터 화소 카운트까지의 모든 바이트들의 CRC를 포함한다. 이러한 CRC 검사가 실패하면, 전체 패킷은 폐기된다.

화소 데이터 필드는 디스플레이될 원(raw) 비디오 정보를 포함하고, 이는 비디오 데이터 포맷 서술자 필드에 의해 설명되는 방식으로 포맷팅된다. 데이터는 다른 곳에서 설명되는 것과 같이 한 번에 하나의 "로우"에서 전송된다. 화소 데이터 속성 필드의 비트 5가 논리 레벨 1로 설정되면, 화소 데이터 필드는 정확하게 화소들의 하나의 로우를 포함하고, 최 좌측 화소에 대응하는 제1 화소가 전송되고, 최 우측 화소에 대응하는 최종 화소가 전송된다.

화소 데이터 CRC 필드(2 바이트)는 단지 화소 데이터의 16비트 CRC를 포함한다. 이러한 값들의 CRC 검증이 실패하면, 화소 데이터는 여전히 사용될 수 있지만, CRC 에러 카운트가 증분된다.

B. 오디오 스트림 패킷들에 대해서

일 실시예에서, 오디오 채널 ID 필드(1 바이트)는 클라이언트 장치에 의해 오디오 데이터가 전송되는 특정 오디오 채널을 식별하기 위해서, 8비트의 부호를 갖지 않는 정수 값을 사용한다. 물리적인 오디오 채널들은 좌측 전방(front), 우측 전방, 좌측 후방(rear), 우측 후방, 전방 중앙, 서브-우퍼(woofer), 서라운드 좌측, 및 서라운드 우측 채널들을 각각 나타내는, 0,1,2,3,4,5,,6 또는 7 값들로서 이러한 필드에 의해 물리 채널들에 규정 또는 매핑된다. 254 오디오 채널 ID 값은 디지털 오디오 샘플들 중 하나의 스트림이 좌측 전방 및 우측 전방 채널들 모두로 전송된다는 것을 표시한다. 이는 예를 들면 스테레오 헤드셋이 음성 통신을 위해 사용되는 경우의 애플리케이션, 또는 간단한 사용자 인터페이스가 경고 톤들을 생성하는 경우의 다른 애플리케이션에 대한 통신을 간략화한다. 8-253 및 255 범위의 ID 필드에 대한 값들은, 당업자에 의해 기대되는 추가적인 지정들을 새로운 설계가 요구하는 경우에 사용하기 위해서 현재 예비된다.

예비 1 필드(1 바이트)는 일반적으로 차후 사용을 위해 예비되고, 0으로 설정된 이러한 필드 내의 모든 비트들을 갖는다. 이러한 필드의 하나의 기능은 모든 뒤이은 2 바이트 필드들이 16비트 워드 어드레스와 정렬되도록 하고, 4 바이트 필드들이 32 비트 워드 어드레스와 정렬되도록 하는 것이다.

오디오 샘플 카운트 필드(2 바이트)는 이러한 패킷에서 오디오 샘플들의 수를 규정한다.

샘플당 비트들 및 패킹 필드는 오디오 데이터의 패킹 포맷을 규정하는 1 바이트를 포함한다. 일 실시예에서, 일반적으로 사용되는 포맷은 비트 4 내지 0이 PCM 오디오 샘플당 비트들의 수를 정의하도록 하는 것이다. 비트 5는 디지털 오디오 데이터 샘플들이 패킹(pack)되는지 여부를 규정한다. 상술한 바와 같이, 도 12는 패킹 및 바이트-정렬된 오디오 샘플들 사이의 차이를 보여주는 도이다. 비트 5에 대한 값, "0"은 디지털 오디오 데이터 필드의 각각의 PCM 오디오 샘플이 인터페이스 바이트 경계와 바이트-정렬됨을 표시하고, 값 "1"은 각각의 연속적인 PCM 오디오 샘플이 이전 오디오 샘플에 대해 패킹됨을 표시한다. 이러한 비트는 비트 4 내지 0에 정의된 값(PCM 오디오 샘플당 비트들의 수)이 8의 배수가 아닌 경우에만 유효하다. 비트 7 내지 6은, 시스템 설계들이 추가적인 지정들을 필요로 하고, 일반적으로 0 값으로 설정되는 경우에 사용하기 위해서 예비된다.

오디오 샘플 레이트 필드(1 바이트)는 오디오 PCM 샘플 레이트를 규정한다. 사용되는 포맷에서, 값 0은 초당 8,000 샘플들 레이트(sps)를 나타내고, 값 1은 16,000 sps를 나타내며, 값 2는 24,000 sps를 나타내며, 값 3은 32,000 sps를 나타내며, 값 4는 40,000 sps를 나타내며, 값 5는 48,000 sps를 나타내며, 값 6은 11,025 sps를 나타내며, 값 7은 22,050 sps를 나타내며, 값 8은 44,100 sps를 나타내며, 값 9 내지 255는 차후 사용을 위해 예비되어, 이들은 현재 0으로 설정된다.

파라미터 CRC 필드(2 바이트)는 패킷 길이로부터 오디오 샘플 레이트까지의 모든 바이트들의 16 비트 CRC를 포함한다. 이러한 CRC에 대한 검사가 실패하면, 전체 패킷은 폐기된다. 디지털 오디오 데이터 필드는 플레이될 원(raw) 오디오 샘플들을 포함하며, 일반적으로 부호를 갖지 않는 정수로서 선형 포맷 형태로 존재한다. 오디오 데이터 CRC 필드(2 바이트)는 단지 오디오 데이터에 대한 16 비트 CRC를 포함한다. 이러한 CRC가 검사에 실패하면, 오디오 데이터는 여전히 사용될 수 있지만, CRC 에러 카운트가 증분된다.

C. 사용자-정의된 스트림 패킷들에 대해서

일 실시예에서, 2 바이트 스트림 ID 번호 필드는 특정한 사용자 정의된 스트림을 식별하기 위해서 사용된다. 스트림 파라미터들 및 스트림 데이터 필드들의 콘텐츠들은 일반적으로 MDDI 장비 제조자에 의해 정의된다. 2 바이트 스트림 파라미터 CRC 필드는 패킷 길이로부터 오디오 코딩 바이트까지의 스트림 파라미터들의 모든 바이트들에 대한 16 비트 CRC를

포함한다. 이러한 CRC 검사가 실패하면, 전체 패킷은 폐기된다. 스트림 파라미터들 및 스트림 파라미터 CRC 필드들 모두는, MDDI 인터페이스의 말단 애플리케이션에 의해 요구되지 않는 경우 폐기될 수 있으며, 즉 이들은 선택적인 것으로 간주된다. 2 바이트 스트림 데이터 CRC 필드는 단지 스트림 데이터에 대한 CRC를 포함한다. 이러한 CRC가 검사에 실패하면, 스트림 데이터의 사용은 애플리케이션의 요구조건에 따라 선택적이다(optional). 양호한 CRC에 의존하는 스트림 데이터의 사용은 일반적으로 CRC가 양호한 것으로 확인될 때까지 스트림 데이터가 버퍼링될 것을 필요로 한다. CRC 에러 카운트가 CRC가 검사되지 않는 경우 증분된다.

D. 컬러 맵 패킷에 대해서

2 바이트 h클라이언트 ID 필드는 이전에 사용된 바와 같이, 클라이언트 ID를 위해 예비되는 값들의 정보를 포함한다. 이러한 필드는 차후 사용을 위해 예비되기 때문에, 비트들을 "0"으로 설정함으로써 현재 값은 0으로 설정된다.

2 바이트 컬러 맵 아이템 카운트 필드는 컬러 맵 데이터 필드에 포함되는 3 바이트 컬러 맵 아이템들 또는 이러한 패킷의 컬러 맵 데이터에 존재하는 컬러 맵 테이블 엔트리들을 규정하는 값들을 사용한다. 이러한 실시예에서, 컬러 맵 데이터의 바이트들의 수는 컬러 맵 아이템 카운트의 3배이다. 컬러 맵 아이템 카운트는 어떠한 컬러 맵 데이터도 전송하지 않기 위해서 0으로 설정된다. 컬러 맵 사이즈가 0 이면, 컬러 맵 오프셋 값은 일반적으로 여전히 전송되지만, 이는 디스플레이에 의해 무시된다. 컬러 맵 오프셋 필드(4 바이트)는 클라이언트 장치의 컬러 맵 테이블의 시작으로부터 이러한 패킷의 컬러 맵 데이터의 오프셋을 규정한다.

2 바이트 파라미터 CRC 필드는 패킷 길이로부터 오디오 코딩 바이트까지의 모든 바이트들에 대한 CRC를 포함한다. 이러한 CRC가 검사에 실패하면, 전체 패킷은 폐기된다.

컬러 맵 데이터 필드에 있어서, 각 컬러 맵 위치의 폭은 컬러 맵 아이템 사이즈 필드에 의해 규정되고, 일 실시예에서 제1 부분은 청색의 크기를 규정하고, 제2 부분은 녹색의 크기를 규정하며, 제3 부분은 적색의 크기를 규정한다. 컬러 맵 사이즈 필드는 컬러 맵 데이터 필드에 존재하는 3 바이트 컬러 맵 테이블 아이템들의 수를 규정한다. 하나의 컬러 맵이 하나의 비디오 데이터 포맷 및 컬러 맵 패킷과 맞지 않으면, 전체 컬러 맵은 각 패킷에서 상이한 컬러 맵 데이터 및 컬러 맵 오프셋들을 갖는 다수의 패킷들을 전송함으로써 규정된다. 각 컬러 맵 아이템 데이터에서 적, 녹, 청의 비트들의 수는 일반적으로 디스플레이 기능 패킷의 컬러 맵 RGB 폭 필드에서 규정된 것과 동일하다.

2 바이트 컬러 맵 CRC 필드는 단지 컬러 맵 데이터에 대한 CRC를 포함한다. CRC 검사가 실패하면, 컬러 맵 데이터는 여전히 사용될 수 있지만, CRC 에러 카운트가 증분된다.

각각의 컬러 맵 데이터 아이템은 다음 순서로 전송된다: 청, 녹, 적, 각 컴포넌트의 최하위 비트가 먼저 전송됨. 각 컬러 맵 아이템의 각각의 적, 녹, 청 컴포넌트들은 패킹되지만, 각각의 컬러 맵 아이템(청 컴포넌트의 최하위 비트)은 바이트-정렬되어야 한다. 도100은 6 비트 청색, 8비트 녹색, 7비트 적색을 갖는 컬러 맵 데이터 아이템들의 예를 보여준다. 이러한 예에서, 컬러 맵 패킷의 컬러 맵 아이템 사이즈는 21이고, 클라이언트 기능 패킷의 컬러 맵 RGB 폭 필드는 0x0786이다.

E. 역방향 링크 인캡슐레이션 패킷들에 대해서

파라미터 CRC 필드(2 바이트)는 패킷 길이로부터 턴-어라운드 길이까지의 모든 바이트들에 대한 16 비트 CRC를 포함한다. 이러한 CRC가 검사에 실패하면, 전체 패킷은 폐기된다.

일 실시예에서, 역방향 링크 플래그 필드(1 바이트)는 클라이언트로부터 정보를 요청하기 위해서 한 세트의 플래그들을 포함한다. 비트(예를 들면, 비트 0)가 논리 1 레벨로 설정되면, 호스트는 클라이언트 기능 패킷을 사용하여 디스플레이로부터 규정된 정보를 요청한다. 비트가 논리 0 레벨로 설정되면, 호스트는 클라이언트로부터 정보를 필요로 하지 않는다. 나머지 비트들(여기서, 비트 0 내지 7)은 차후 사용을 위해 예비되고, 0으로 설정된다. 그러나, 필요에 따라, 역방향 링크들에 대한 플래그들을 설정하기 위해서 보다 많은 비트들이 사용될 수 있다.

역방향 레이트 디바이저 필드(1 바이트)는 역방향 링크 데이터 클럭에 대해 일어나는 MDDI_Stb 사이클들의 수를 규정한다. 역방향 링크 데이터 클럭은 2 배의 역방향 레이트 디바이저에 의해 나뉘진 순방향 링크 데이터 클럭과 동일하다. 역방향 링크 데이터 레이트는 역방향 링크 데이터 클럭 및 역방향 링크에 대한 인터페이스 타입과 관련된다. 이러한 실시예에서, 타입 1 인터페이스에 대해서, 역방향 데이터 레이트는 역방향 링크 데이터 클럭과 동일하고, 타입 2, 타입 3, 및 타입 4 인터페이스들에 대해서, 역방향 데이터 레이트들은 역방향 링크 데이터 클럭의 2배, 4배, 및 8배와 각각 동일하다.

논리 제로 레벨로 비트들을 설정함으로써 값이 제로와 동일하게 설정되는, 올 제로 1 필드는 한 그룹의 바이트들(여기서 8)을 포함하고, 턴 어라운드 1 필드 동안 호스트 라인 드라이버들을 디스에이블링하기에 앞서 단지 MDDI_Stb만을 사용하여 클라이언트가 클록 복원을 개시할 수 있도록 충분한 시간 동안 모든 MDDI_Data 신호들이 논리 0 레벨에서 존재하는 것을 보장하기 위해서 사용된다. 일 실시예에서, 올 제로 1 필드의 길이는 케이블의 왕복 지연에서 순방향 링크 바이트 전송 횟수들과 동일하거나 크다.

턴 어라운드 1 길이 필드(1 바이트)는 제1 턴 어라운드 주기를 설정하는, 턴 어라운드 1에 할당되는 총 바이트들의 수를 규정한다. 턴 어라운드 1 길이 파라미터에 의해 규정되는 바이트들의 수는, 호스트의 라인 드라이버들이 디스에이블되기 전에, 클라이언트의 MDDI_Data 라인 드라이버들이 인에이블 되도록 하기 위해서 할당된다. 턴 어라운드 1의 비트 0 동안 클라이언트는 MDDI_Data 라인 드라이버들을 인에이블하고, 턴 어라운드 1의 최종 비트에 앞서 완전히 디스에이블 되도록 하기 위해서 호스트는 그 출력을 디스에이블한다. 클라이언트 드라이버의 인에이블링 및 호스트 드라이버 프로세스들의 타이밍은 이들 중 하나 또는 이 둘 모두가 호스트의 라인 수신기들에 의해 보여지는 바와 같이 턴 어라운드 1을 통해 논리 0 레벨로 MDDI_Data 신호들을 드라이브하도록 한다. MDDI_Stb 신호는 전체 턴 어라운드 1 주기 동안 MDDI_Data0이 논리 0 레벨인 것처럼 행동한다. 턴 어라운드 1의 세팅에 대한 보다 상세한 내용은 전술한 바와 같다.

역방향 데이터 패킷 필드는 클라이언트로부터 호스트로 전송되는 일련의 데이터 패킷들을 포함한다. 클라이언트가 호스트로 전송할 데이터가 없는 경우, 클라이언트는 채움(filler) 패킷을 전송하고, MDDI_Data 라인들을 논리 0 상태 또는 레벨로 구동할 수 있다. 이러한 실시예에서, MDDI_Data 라인들이 0으로 구동되면, 호스트는 이를 0 길이(유효 길이가 아님)를 갖는 패킷으로 해석하고, 호스트는 현재 역방향 링크 인캡슐레이션 패킷의 듀레이션 동안 클라이언트로부터 어떠한 추가적인 패킷들로 수용하지 않을 것이다.

턴 어라운드 2 길이 필드(1 바이트)는 제2 턴 어라운드 주기를 설정하기 위해서 턴 어라운드 2에 대해 할당되는 바이트들의 총 수를 규정한다. 턴 어라운드 길이 파라미터에 의해 규정되는 바이트들의 수는 클라이언트의 라인 드라이버들이 디스에이블되기 전에 호스트의 MDDI_Data 라인 드라이버들이 인에이블되도록 하기 위해서 할당된다. 호스트는 턴 어라운드 2의 제1 바이트의 비트 0 동안 그 MDDI_Data 라인 드라이버들을 인에이블하고, 클라이언트는 턴 어라운드 2의 최종 비트에 앞서 클라이언트의 출력들이 완전히 디스에이블되도록 하기 위해서 그 출력들을 디스에이블한다. 클라이언트 드라이버의 인에이블링 및 호스트 드라이버 인에이블링 처리들은 호스트의 라인 수신기들에 의해 보여지는 바와 같이, 턴 어라운드 2 전체 주기 동안 이들 중 하나 또는 이 둘 모두가 MDDI_Data 신호들을 논리 0 레벨로 구동하도록 한다. MDDI_Stb 신호는 실질적으로 전체 턴 어라운드 2 주기 동안 MDDI_Data0은 논리 0 레벨인 것처럼 행동한다. 턴 어라운드 2의 세팅에 대한 설명은 전술한 바와 같다.

역방향 데이터 패킷 필드들은 클라이언트로부터 호스트로 전송되는 일련의 데이터 패킷들을 포함한다. 전술한 바와 같이, 채움 패킷들은 다른 패킷 타입들에 의해 사용되는 않는 잔존 공간을 채우기 위해서 전송된다.

올 제로 2 필드는 비트들을 논리 0 레벨로 설정함으로써 그 값이 0으로 설정되는 한 그룹의 바이트들(본 실시예에서 8)을 포함하고, 다음 턴 어라운드 2 필드 후에 호스트의 라인 드라이버들을 인에이블한 후에, MDDI_Data0 및 MDDI_Stb 모두를 사용하여 클라이언트가 복원 클록을 개시할 수 있도록 하기 위해서 충분한 시간 동안 모든 MDDI_Data 신호들이 논리 0 레벨이 되는 것을 보장하기 위해서 사용된다.

F. 클라이언트 기능 패킷들에 대해서

일 실시예에서 제시된 바와 같이, 프로토콜 버전 필드는 클라이언트에 의해 사용되는 프로토콜 버전을 규정하기 위해서 2 바이트를 사용한다. 초기 버전은 현재 1로 설정되고, 시간이 흘러 새로운 버전이 생성되면 변경되고, 최소 프로토콜 버전 필드는 클라이언트가 사용 또는 해석하는 최소 프로토콜 버전을 규정하기 위해서 2 바이트를 사용한다. 이러한 경우, 0 값은 또한 유효 값이다. 데이터 레이트 기능 필드(2 바이트)는 인터페이스의 순방향 링크 상에서 각 데이터 쌍에 대해 클라이언트가 수신할 수 있는 최대 데이터 레이트를 규정하며, 초당 매가비트(Mbps) 형태로 규정된다. 인터페이스 타입 기능 필드(1 바이트)는 순방향 및 역방향 링크들 상에서 지원되는 인터페이스 타입들을 규정한다. "1"로 설정된 비트는 규정된 인터페이스 타입이 지원됨을 의미하고, "0"으로 설정된 비트는 규정된 타입이 지원되지 않음을 의미한다. 호스트 및 클라이언트들은 순방향 및 역방향 라인들 상에서 적어도 타입 1을 지원하여야만 한다. 인터페이스 타입들의 인접한 범위를 지원하기 위한 요구조건은 존재하지 않는다. 예를 들어, 인터페이스에서 단지 타입 1 및 타입 3만을 지원하고, 타입 3 및 타입 4는 지원하지 않는 것이 유효할 수 있다. 순방향 및 역방향 링크들이 동일한 인터페이스 타입에서 동작할 필요도 없다. 그러나, 링크가 동면 상태에서부터 벗어나면, 순방향 및 역방향 링크들 모두는, 다른 모드들이 협상되고, 선택되고, 또는 호스트 및 클라이언트 모두에 의해 사용이 승인될 때까지, 타입 1에서 동작을 개시하여야 한다.

일 실시예에서, 지원되는 인터페이스들은 순방향 링크에서, 타입 2(2 비트), 타입 3(4 비트), 또는 타입 4(8 비트) 모드 중 하나를 선택하기 위해서, 비트 0, 1, 또는 2를 선택하고; 역방향 링크에서 타입 2, 타입 3, 또는 타입 4를 선택하기 위해서 비트 3, 4, 또는 5를 선택함으로써 표시되고; 여기서 비트 6 및 7은 예비되어 이 경우 0으로 설정된다. 비트맵 폭 및 높이 필드들(여기서 각각 2 바이트임)은 화소들로 각각 비트맵의 폭 및 높이를 규정한다.

단색 기능 필드(1 바이트)는 단색 형태로 디스플레이될 수 있는 해상도의 비트들 수를 규정하기 위해서 사용된다. 디스플레이가 단색 포맷을 사용할 수 없으면, 이 값은 0으로 설정된다. 비트 7 내지 4는 차후 사용을 위해 예비되고, 따라서 0으로 설정된다. 비트 3 내지 0은 각 화소에 대해 존재할 수 있는 그레이스케일의 최대 비트들 수를 정의한다. 이러한 4개의 비트들은 각 화소에 대해 1 내지 15까지의 값들을 규정할 수 있게 한다. 이 값이 0이면, 단색 포맷은 디스플레이에 의해 지원되지 않는다.

베이어 기능 필드는 해상도, 화소 그룹, 베이어 포맷에서 전달될 수 있는 화소 차수(order)의 비트들 수를 규정하기 위해서 1 바이트를 사용한다. 클라이언트가 베이어 포맷을 사용할 수 없으면, 이 값은 0이다. 베이어 기능 필드는 다음 값들로 구성된다: 비트 3 내지 0은 각 화소에서 존재하는 강도 비트들의 최대 수를 정의하고, 비트 5 내지 4는 요구되는 화소 그룹 패턴을 정의하며, 비트 8 내지 6은 요구되는 화소 차수를 정의하며; 비트 14 내지 9는 차후 사용을 위해 예비되어 그 중간에 0으로 설정된다. 논리 1 레벨로 설정될 때, 비트 15는 클라이언트가 패킹되거나 패킹되지 않은 포맷으로 베이어 화소 데이터를 수용할 수 있음을 표시한다. 비트 15가 0으로 설정되면, 이는 클라이언트가 단지 패킹되지 않은 포맷으로만 베이어 화소 데이터를 수용할 수 있음을 표시한다.

컬러 맵 기능 필드(3 바이트)는 디스플레이의 컬러 맵 테이블에서 존재하는 테이블 아이템들의 최대 수를 규정한다. 디스플레이가 컬러 맵 포맷을 사용할 수 없으면, 이 값은 0으로 설정된다.

RGB 기능 필드(2 바이트)는 RGB 포맷에서 디스플레이될 수 있는 해상도의 비트들 수를 규정한다. 디스플레이가 RGB 포맷을 사용할 수 없으면, 이 값은 0이다. RGB 기능 필드는 3개의 개별적인 부호를 갖지 않는 값들로 구성되고; 여기서 비트 3 내지 0은 청색 비트들의 최대 수를 정의하고, 비트 7 내지 4는 녹색 비트들의 최대 수를 정의하며, 비트 11 내지 8은 각 화소에서 적색 비트들의 최대 수를 정의한다. 현재, 비트 14 내지 12는 차후 사용을 위해 예비되고, 일반적으로 0으로 설정된다. 비트 14 내지 12는 차후 사용을 위해 예비되고, 일반적으로 0으로 설정된다. 논리 1 레벨로 설정되는 경우, 비트 15는 클라이언트가 패킹된 또는 패킹되지 않은 포맷에서 RGB 화소 데이터를 수용할 수 있다는 것을 나타낸다. 비트 15가 논리 0 레벨로 설정되면, 이는 클라이언트가 단지 패킹되지 않은 포맷으로 RGB 화소 데이터를 수용할 수 있음을 나타낸다.

Y Cr Cb 기능 필드(2 바이트)는 Y Cr Cb 포맷으로 디스플레이될 수 있는 해상도의 비트들의 수를 규정한다. 디스플레이가 Y Cr Cb 포맷을 사용할 수 없으면, 이 값은 0으로 설정된다. Y Cr Cb 기능 필드는 3개의 개별적인 부호를 갖지 않는 값들로 구성되고; 여기서 비트 3 내지 0은 Cb 샘플에서 비트들의 최대 수를 정의하고, 비트 7 내지 4는 Cr 샘플에서 최대 비트들의 수를 정의하며, 비트 11 내지 8은 Y 샘플에서 최대 비트들의 수를 정의하며, 비트 15 내지 12는 차후 사용을 위해 현재 예비되며, 0으로 설정된다.

클라이언트 특징 기능 필드는 지원되는 클라이언트에서 특정 특징을 표시하는 한 세트의 플래그들을 포함하는 4 바이트를 사용한다. 논리 1로 설정된 비트는 그 기능 필드가 지원됨을 표시하고, 논리 0 레벨로 설정된 비트는 그 기능 필드가 지원되지 않음을 표시한다. 일 실시예에서, 비트 0에 대한 값은 비트맵 블록 전달 패킷(패킷 타입 71)이 지원되는지 여부를 표시한다. 비트 1, 2 및 3에 대한 값은 비트맵 채움 패킷(패킷 타입 72), 비트맵 패턴 채움 패킷(패킷 타입 73), 또는 통신 링크 데이터 채널 패킷(패킷 타입 74) 각각이 지원되는지 여부를 표시한다. 비트 4에 대한 값은 투명한 컬러 인에이블 패킷을 사용하여, 투명한 하나의 컬러를 만드는 기능을 클라이언트가 가지는지 여부를 나타내며, 비트 5 및 6에 대한 값들은 클라이언트가 패킹된 포맷으로 비디오 데이터 또는 오디오 데이터를 각각 수용할 수 있는지 여부를 나타내며, 비트 7에 대한 값은 클라이언트가 카메라로부터 역방향 링크 비디오 스트림을 전송할 수 있는지 여부를 나타낸다. 비트 8에 대한 값은 클라이언트가 화소 데이터의 완전한 라인을 수신하고, 비디오 스트림 패킷의 화소 데이터 속성 필드의 비트 5에 의해 규정되는 디스플레이 어드레싱을 무시할 수 있는 능력을 갖는지를 표시하며, 클라이언트는 화소 데이터 속성 필드의 비트 15를 사용하여 비디오 프레임 데이터의 중단부 또는 프레임 동기를 검출할 수 있다.

비트 11 및 12에 대한 값은 클라이언트가 포인팅 장치와 통신하여 포인팅 장치 데이터 패킷을 송신 및 수신할 수 있을 때, 또는 키보드와 통신하여 키보드 데이터 패킷들을 송신 및 수신할 수 있는 때를 각각 표시한다. 비트 13에 대한 값은 VCP 특징 패킷들(요청 VCP 특징 패킷, VCP 특징 응답 패킷, 셋 VCP 특징 패킷, 요청 유효 파라미터 패킷, 및 유효 파라미터 응답 패킷)을 지원함으로써 하나 이상의 오디오 또는 비디오 파라미터들을 설정할 능력을 클라이언트가 갖는지 여부를 표시

한다. 비트 14에 대한 값은 화소 데이터를 오프라인 디스플레이 프레임 버퍼에 기록할 수 있는 능력을 클라이언트가 갖는지 여부를 표시한다. 이러한 비트가 논리 1 레벨로 설정되면, 디스플레이 업데이트 비트(비디오 스트림 패킷의 화소 데이터 속성 필드의 비트 7 및 6)들은 "01" 값으로 설정된다.

비트 15에 대한 값은 디스플레이 이미지를 리프레쉬하기 위해서 현재 사용되는 디스플레이 프레임 버퍼에만 화소 데이터를 기록하는 능력을 클라이언트가 가질 때를 표시한다. 이러한 비트가 1로 설정되면, 디스플레이 업데이트 비트(비디오 스트림 패킷의 화소 데이터 속성 필드의 비트 7 및 6)들은 "00" 값으로 설정된다. 비트 16에 대한 값은 모든 디스플레이 버퍼들로 하나의 비디오 스트림 패킷으로부터의 화소 데이터를 기록할 수 있는 능력을 클라이언트가 가지는 때를 표시한다. 이러한 비트가 1로 설정되면, 디스플레이 업데이트 비트들(비디오 스트림 패킷의 화소 데이터 속성 필드의 비트 7 및 6)는 "11" 값으로 설정된다.

비트 17에 대한 값은 요청 특정 상태 패킷에 응답하는 능력을 클라이언트가 가지는 때를 표시하고, 비트 18에 대한 값은 왕복 지연 측정 패킷에 응답하는 능력을 클라이언트가 가지는 때를 표시하며, 비트 19에 대한 값은 순방향 링크 스큐 캘리브레이션 패킷에 대한 능력을 클라이언트가 가지는 때를 표시한다.

비트 21에 대한 값은 클라이언트가 언제 요청 특정 상태 패킷을 해석할 능력을 가지고 있고 유효 상태 응답 리스트 패킷을 통해 응답하는지를 표시한다. 클라이언트는 설명된 바와 같이 유효 상태 응답 리스트 패킷의 유효 파라미터 응답 리스트 필드에 있는 추가적인 상태를 리턴시키기 위한 능력을 표시한다.

비트 22에 대한 값은 클라이언트가 등록 액세스 패킷에 응답할 능력을 가지고 있는지 여부를 표시한다. 비트 9, 10, 20, 23 내지 31은 나중의 사용을 위해 또는 시스템 설계자들에게 유용한 대안적인 설계들을 위해 현재 예비되어 있으며 일반적으로 0으로 설정되어 있다.

디스플레이 비디오 프레임 레이트 기능 필드(1 바이트)는 초당 프레임에서 디스플레이의 최대 비디오 프레임 갱신 기능을 특정한다. 호스트는 이 필드에서 특정된 값보다 낮은 레이트에서 이미지를 갱신하도록 선택할 수 있다.

오디오 버퍼 깊이(depth) 필드(2 바이트)는 각각의 오디오 스트림들로 지정된 디스플레이의 유연한 버퍼의 깊이를 특정한다.

오디오 채널 기능 필드(2 바이트)는 어떤 오디오 채널들이 클라이언트 또는 클라이언트에 연결된 장치에 의해 지원되는지를 표시하는 플래그들의 그룹을 포함한다. 1로 설정된 비트는 채널이 지원되는 것을 표시하며, 0으로 설정된 비트는 채널이 지원되지 않는 것을 표시한다. 비트 위치들은, 예를 들어, 일 실시예에서 비트 위치들 0, 1, 2, 3, 4, 5, 6 및 7에 대한 상이한 채널들로 지정되며, 좌측 전방, 우측 전방, 좌측 후방, 우측 후방, 전방 중심, 서브-우퍼, 서라운드 좌측 및 서라운드 우측을 나타낸다. 비트들 8 내지 14는 나중의 사용을 위해 현재 예비되어 있으며, 일반적으로 0으로 설정되어 있다. 일 실시예에서, 비트 15는 클라이언트가 순방향 오디오 채널 인에이블 패킷에 대한 지원을 제공하는지 여부를 표시하기 위해 사용된다. 이러한 경우에, 비트 15는 로직-1 레벨로 설정된다. 그러나, 클라이언트가 순방향 오디오 채널 인에이블 패킷의 결과로서 오디오 채널들을 디스에이블링할 수 없거나 또는 클라이언트가 어떠한 오디오 기능도 지원하지 않는 경우에는, 상기 비트는 로직-0 레벨 또는 값으로 설정된다.

순방향 링크에 대한 2바이트 오디오 샘플 레이트 기능 필드는 클라이언트 장치의 오디오 샘플 레이트 기능을 나타내기 위해 플래그의 세트를 포함한다. 비트 위치는 결과적으로 상이한 레이트로 할당되는데, 예를 들어, 비트 0, 1, 2, 3, 4, 5, 6, 7 및 8은 각각 8,000, 16,000, 24,000, 32,000, 40,000, 48,000, 11,025, 22,025 및 44,100 초당 샘플(SPS)로 할당되며, 비트 9 내지 15는 미래를 위해 또는 원할 경우 택일적인 레이트 사용을 위해 보존되므로, 이들은 일반적으로 '0'으로 설정된다. 이러한 비트 중 하나에 대한 비트 값을 '1'로 설정하는 것은 특정 샘플 레이트가 지원되는 것을 나타내며, 비트를 '0'으로 설정하는 것은 샘플 레이트가 지원되지 않는 것을 나타낸다.

최소 서브 프레임 레이트 필드(2바이트)는 초당 프레임에서 최소 서브 프레임 레이트를 특정한다. 최소 서브 프레임 레이트는 디스플레이 상태를 디스플레이에서 소정의 센서 또는 포인팅 장치를 판독하기에 충분한 업데이트 레이트로 유지한다.

역방향 링크에 대한 2바이트 Mic 샘플 레이트 기능 필드는 클라이언트 장치의 마이크로폰의 오디오 샘플 레이트를 나타내는 플래그의 세트를 포함한다. MDDI를 위해, 클라이언트 장치는 마이크로폰은 적어도 초당 8,000 샘플을 최소한도로 지원하도록 구성된다. 이러한 필드에 대한 비트 위치는 상이한 레이트로 할당되는데, 예를 들어, 위치 0, 1, 2, 3, 4, 5, 6, 7 및 8은 8,000, 16,000, 24,000, 32,000, 40,000, 48,000, 11,025, 22,025 및 44,100 초당 샘플을 각각 나타내기 위해 사

용되며, 비트 9 내지 15는 미래의 사용 또는 원할 경우 택일적 레이트 용도를 위해 보존되므로, 일반적으로 '0'으로 설정된다. 이러한 비트들 중 하나에 대한 비트 값을 '1'로 설정하는 것은 특정 샘플 레이트가 지원되는 것을 나타내며, 비트를 '0'으로 설정하는 것은 샘플 레이트가 지원되지 않는 것을 나타낸다. 만일 어떠한 마이크론도 연결되지 않는다면, Mic 샘플 레이트 기능 비트 각각은 0으로 설정된다.

키보드 데이터 포맷 필드(여기서 1 바이트)는 키보드가 클라이언트 시스템에 연결되는지 여부와 연결된 키보드 타입을 특정한다. 일 실시예에서, 비트들 6 내지 0에 의해 설정된 값은 연결된 키보드의 타입을 정의하기 위해 사용된다. 값이 제로(0)이면, 키보드 타입은 알려지지 않은 것으로 간주된다. 값이 1이면, 키보드 데이터 포맷은 표준 PS-2 스타일인 것으로 간주된다. 현재 2 내지 125의 범위에 있는 값들은 사용되지 않으며, MDD 인터페이스 및 대응하는 클라이언트들 또는 호스트들과 함께 사용하기 위한 특정한 키보드 또는 입력 장치들을 정의하도록 시스템 설계자들, 인터페이스 통합자들 또는 제품 개발자들의 사용을 위해 예비된다. 값 126은 키보드 데이터 포맷이 사용자-정의됨을 나타내기 위해 사용되며, 값 127은 키보드가 이러한 클라이언트에 연결될 수 없다는 것을 나타내기 위해 사용된다. 또한, 비트 7은 키보드가 클라이언트와 통신할 수 있는지 여부를 나타내기 위해 사용될 수 있다. 이러한 비트의 의도된 사용은 키보드가 언제 무선 링크를 이용하는 클라이언트와 통신할 수 있는지를 나타내어야 한다. 비트들 6 내지 0이 키보드가 클라이언트와 연결될 수 없다는 것을 나타내면, 비트 7은 0 레벨로 설정될 것이다. 그러므로, 일 실시예에서, 비트 7의 값이 0이면, 키보드와 클라이언트는 통신할 수 없으며, 비트 7의 값이 1이면, 키보드와 클라이언트는 그들이 서로와 통신할 수 있다는 것을 승인한다.

포인팅 장치 데이터 포맷 필드(여기서, 1 바이트)는 포인팅 장치가 클라이언트 시스템에 연결되는지 여부와 연결된 포인팅 장치의 타입을 특정한다. 일 실시예에서, 비트들 6 내지 0에 의해 설정된 값은 연결된 포인팅 장치의 타입을 정의하기 위해 사용된다. 값이 제로(0)이면, 포인팅 장치 타입은 알려지지 않은 것으로 간주된다. 값이 1이면, 포인팅 장치 데이터 포맷은 표준 PS-2 스타일인 것으로 간주된다. 현재 2 내지 125의 범위에 있는 값들은 사용되지 않으며, MDD 인터페이스 및 대응하는 클라이언트들 또는 호스트들과 함께 사용하기 위한 포인팅 장치 또는 입력 장치들을 정의하도록 시스템 설계자들, 인터페이스 통합자들 또는 제품 개발자들의 사용을 위해 예비된다. 값 126은 포인팅 장치 데이터 포맷이 사용자-정의됨을 나타내기 위해 사용되며, 값 127은 포인팅 장치가 이러한 클라이언트에 연결될 수 없다는 것을 나타내기 위해 사용된다. 또한, 비트 7은 포인팅 장치가 클라이언트와 통신할 수 있는지 여부를 나타내기 위해 사용될 수 있다. 이러한 비트의 의도된 사용은 포인팅 장치가 언제 무선 링크를 이용하는 클라이언트와 통신할 수 있는지를 나타내어야 한다. 비트들 6 내지 0이 포인팅 장치가 클라이언트와 연결될 수 없다는 것을 나타내면, 비트 7은 0 레벨로 설정될 것이다. 그러므로, 일 실시예에서, 비트 7의 값이 0이면, 포인팅 장치와 클라이언트는 통신할 수 없으며, 비트 7의 값이 1이면, 포인팅 장치와 클라이언트는 그들이 서로와 통신할 수 있다는 것을 승인한다.

컨텐츠 보호 타입 필드(2 바이트)는 디스플레이에 의해 지원되는 디지털 컨텐츠 보호 타입을 표시하는 플래그들의 세트를 포함한다. 현재, 비트 위치 0은 DTCP가 언제 지원되는지를 나타내기 위해 사용되고, 비트 위치 1은 HDCP가 언제 지원되는지를 나타내기 위해 사용되고, 비트 위치들 2 내지 15는 원하거나 또는 이용가능한 다른 보호 방식들에서 사용하기 위해 예비되며, 이들은 현재 0으로 설정되어 있다.

제조사 이름 필드(여기서, 2 바이트)는 VESA EDID 사양과 동일한 방식으로 세 개의 5-비트 캐릭터들로 패키징된 제조자의 EISA 3-캐릭터 ID를 포함한다. 캐릭터 'A'는 이진수 00001로 표현되고, 캐릭터 'Z'는 이진수 11010으로 표현되고, 'A'와 'Z' 사이에 있는 모든 캐릭터들은 'A'와 'Z' 사이의 알파벳 순서에 대응하는 순차적인 이진값들로서 표현된다. 제조자 이름 필드의 최상위 비트는 사용되지 않으며 나중에 사용될 때까지 일반적으로 로직-0으로 설정된다. 예시: 스트림 "XYZ"에 의해 표현된 제조자는 0x633a의 제조자 이름값을 가질 것이다. 이러한 필드가 클라이언트에 의해 지원되지 않으면 필드는 일반적으로 0으로 설정된다. 제품 코드 필드는 디스플레이 제조자에 의해 지정된 제품 코드를 포함하기 위해 2 바이트를 사용한다. 이러한 필드가 클라이언트에 의해 지원되지 않으면 필드는 일반적으로 0으로 설정된다.

예비 1, 예비 2 및 예비 3 필드들(여기서 각각 2 바이트)은 정보 전달을 위한 나중의 사용을 위해 예비된다. 이러한 필드들에 있는 모든 비트들은 일반적으로 로직-0 레벨로 설정된다. 이러한 필드들의 목적은 현재 모든 다음 2 바이트 필드들이 16-비트 워드 주소로 지정되도록 하고 4-바이트 필드들이 32-비트 워드 주소로 지정되도록 하는 것이다.

일련 번호 필드는 숫자 형태로 디스플레이의 일련 번호를 특정하기 위해 이러한 실시예에서 4 바이트를 사용한다. 이러한 필드가 클라이언트에 의해 지원되지 않으면 필드는 일반적으로 0으로 설정된다. 제조한 주(week) 필드는 디스플레이의 제조한 주를 정의하기 위해 1 바이트를 사용한다. 상기 필드가 클라이언트에 의해 지원되면 이러한 값은 전형적으로 1부터 53까지의 범위에 있다. 이러한 필드가 클라이언트에 의해 지원되지 않으면 필드는 0으로 설정된다. 제조년도 필드는 디스플레이의 제조년도를 정의하기 위해 1 바이트를 사용한다. 이러한 값은 1990년도부터 오프셋되어 있다. 1991년부터 2245년까지의 연도들이 이러한 필드에 의해 표현될 수 있다. 예시: 2003년도는 13의 제조년도 값에 대응한다. 이러한 필드가 클라이언트에 의해 지원되지 않으면 필드는 0으로 설정된다.

CRC 필드(여기서 2 바이트)는 패킷 길이를 포함하는 패킷들에 있는 모든 바이트의 16-비트 CRC를 포함한다.

G. 클라이언트 요청 및 상태 패킷에 대해

역방향 링크 요청 필드(3바이트)는 정보를 호스트로 전송하기 위해 다음의 서브 프레임에서 역방향 링크에서 클라이언트가 필요로 하는 바이트의 수를 특정한다.

CRC 에러 카운트 필드(1바이트)는 얼마나 많은 CRC 에러가 미디어 프레임의 시작 이후에 발생하였는지를 나타낸다. CRC 카운트는 0의 서브 프레임 카운트를 갖는 서브 프레임 헤더 패킷이 전송될 경우 리셋된다. 만일 실제 CRC 에러의 번호가 255를 초과하면, 이러한 값은 일반적으로 255로 포화된다.

기능 변화 필드는 1 바이트를 사용하여 디스플레이의 기능에서의 변화를 나타낸다. 이는 사용자가 마이크론, 키보드, 또는 디스플레이와 같은 주변 장치와 접속할 경우 또는 몇몇 다른 이유로 발생한다. 비트[7:0]가 0과 같다면, 기능은 최종 클라이언트 기능 패킷이 전송되기 때문에 변화되지 않는다. 그러나 비트[7:0]가 1 내지 255와 동일하다면, 기능은 변화한다. 클라이언트 기능 패킷은 새로운 디스플레이 특성을 결정하기 위해 조사된다.

클라이언트 비지(busy) 플래그 필드는 클라이언트가 특정한 기능을 수행하고 있다는 것을 나타내기 위해 2 바이트를 사용한다. 로직-1 레벨 또는 값으로 설정된 비트는 특정 기능이 현재 클라이언트에 의해 수행되며 클라이언트에서 관련된 기능이 바쁜 상태에 있다는 것을 나타낸다. 클라이언트에 있는 관련된 기능이 준비되면, 비트는 로직-0으로 설정된다. 클라이언트는 자신에 의해 지원되지 않는 모든 기능들에 대하여 바쁜 상태(1로 설정된 비트)를 리턴시켜야 한다.

일 실시예에서 이러한 바이트들은 다음과 같은 관계에 따라 해석된다: 비트 0이 '1'이면 비트맵 블록 전달 기능이 바쁜 상태에 있으며, 비트 1이 '1'이면 비트맵 영역 필(fill) 기능이 바쁜 상태에 있으며, 비트 2가 '1'이면 비트맵 패턴 필 기능이 바쁜 상태에 있다. 현재, 비트들 3 내지 15는 나중의 사용을 위해 예비되어 있으며 이러한 비트들이 나중에 지정되면 바쁜 상태를 표시하기 위해 로직-1 레벨 또는 상태로 설정된다.

H. 비트 블록 전달 패킷에 대해

윈도우 상부 좌측 좌표 X 값 및 Y 값 필드는 이동될 윈도우의 상부 좌측 모서리의 좌표의 X 및 Y 값을 특정하기 위해 각각 2 바이트를 사용한다. 윈도우 폭 및 높이 필드는 이동될 윈도우의 폭 및 높이를 특정하기 위해 각각 2 바이트를 사용한다. 윈도우 X 이동 및 Y 이동 필드는 윈도우가 각각 수평 및 수직으로 이동될 픽셀의 수를 특정하기 위해 각각 2 바이트를 사용한다. 일반적으로, 이러한 좌표는 X에 양의 값이 윈도우가 우측으로 이동하게 하고, 음의 값은 좌측으로 이동하게 하는 반면, Y에 대한 양의 값은 윈도우가 아래로 이동하게 하고, 음의 값은 위쪽으로 이동하게 하도록 구성된다.

I. 비트맵 영역 채움 패킷에 대해

윈도우 상부 좌측 좌표 X 값 및 Y 값 필드는 채워질 윈도우의 상부 좌측 모서리의 좌표의 X 및 Y 값을 특정하기 위해 각각 2 바이트를 사용한다. 윈도우 폭 및 높이 필드(각각 2 바이트)는 채워질 윈도우의 폭 및 높이를 특정한다. 비디오 데이터 포맷 기술자 필드(2바이트)는 픽셀 영역 채움 값의 포맷을 특정한다. 포맷은 비디오 스트림 패킷에서 동일 필드와 동일하다. 픽셀 영역 채움 값 필드(4바이트)는 전송한 필드에 의해 특정된 윈도우로 채워질 픽셀 값을 포함한다. 이러한 픽셀의 포맷은 비디오 데이터 포맷 기술자 필드에서 특정된다.

J. 비트맵 패턴 채움 패킷에 대해

윈도우 상부 좌측 좌표 X 값 및 Y 값 필드는 채워질 윈도우의 상부 좌측 모서리의 좌표의 X 및 Y 값을 특정하기 위해 각각 2 바이트를 사용한다. 윈도우 폭 및 높이 필드(각각 2 바이트)는 채워질 윈도우의 폭 및 높이를 특정한다. 패턴 폭 및 패턴 높이 필드들(각각 2 바이트)은 각각 필 패턴의 폭 및 높이를 특정한다. 수형 패턴 오프셋 필드(2 바이트)는 채워질 특정된 윈도우의 좌측 모서리로부터의 픽셀 데이터 패턴의 수형 오프셋을 특정한다. 특정되는 값은 패턴 폭 필드의 값보다 작아야 한다. 수직 패턴 오프셋 필드(2 바이트)는 채워질 특정된 윈도우의 상부 모서리로부터의 픽셀 데이터 패턴의 수직 오프셋을 특정한다. 특정되는 값은 패턴 높이 필드의 값보다 작아야 한다.

2-바이트 비디오 데이터 포맷 서술자 필드는 픽셀 데이터 필 값의 포맷을 특정한다. 도 11은 비디오 데이터 포맷 서술자가 어떻게 코딩되는지를 나타낸다. 상기 포맷은 비디오 스트림 패킷에 있는 동일한 필드와 같다.

파라미터 CRC 필드(2바이트)는 패킷 길이로부터 비디오 포맷 기술자까지의 모든 바이트의 CRC를 포함한다. 만일 이러한 CRC가 체크에 실패하면 전체 패킷이 폐기된다. 패턴 픽셀 데이터 필드는 비디오 데이터 포맷 기술자에 의해 특정된 포맷으로 채움 패턴을 특정하는 원 비디오 정보를 포함한다. 데이터는 바이트로 패킹되며, 각각의 행의 제1 픽셀은 바이트 정렬되어야 한다. 채움 패턴 데이터는 한번에 행 단위로 전송된다. 패턴 픽셀 데이터 CRC 필드(2바이트)는 패턴 픽셀 데이터만의 CRC를 포함한다. 만일 이러한 CRC가 체크에 실패하면 패턴 픽셀 데이터가 여전히 사용될 수 있지만 CRC 에러 카운트는 증가한다.

K. 통신 링크 데이터 채널 패킷

파라미터 CRC 필드(2바이트)는 패킷 길이로부터 패킷 타입까지의 모든 바이트의 16비트 CRC를 포함한다. 만일 이러한 CRC가 체크에 실패하면 전체 패킷이 폐기된다.

통신 링크 데이터 필드는 통신 채널로부터 원 데이터를 포함한다. 이러한 데이터는 디스플레이에서 계산 장치로 간단하게 통과된다.

통신 링크 데이터 CRC 필드(2바이트)는 통신 링크 데이터만의 16비트 CRC를 포함한다. 만일 이러한 CRC가 체크에 실패하면 통신 링크 데이터는 여전히 사용되거나 사용가능하지만, CRC 에러 카운트는 증가한다.

L. 인터페이스 타입 핸드오프 요청 패킷

인터페이스 타입 필드(1바이트)는 사용할 새로운 인터페이스 타입을 특정한다. 이러한 필드에서의 값은 이하의 방법으로 인터페이스 타입을 특정한다. 만일 비트 7에서의 값이 '0'과 동일하면 타입 핸드오프 요청은 순방향 링크에 대한 것이며, 만일 '1'과 같다면, 타입 핸드오프 요청은 역방향 링크에 대한 것이다. 비트 6 내지 3은 미래의 사용을 위해 보존되며 일반적으로 0으로 설정된다. 비트 2 내지 0은 사용될 인터페이스 타입을 한정하는데, 값 1은 타입 1 모드로의 핸드오프를 의미하며, 값 2는 타입 2 모드로의 핸드오프를 의미하며, 값 3은 타입 3 모드로의 핸드오프를 의미하며, 값 4는 타입 4 모드로의 핸드오프를 의미한다. '0' 및 5 내지 7의 값은 택일적 모드 또는 모드의 조합의 미래의 설계를 위해 보존된다.

M. 인터페이스 타입 확인 응답 패킷

인터페이스 타입 필드(1바이트)는 사용할 새로운 인터페이스 타입을 확인하는 값을 갖는다. 이러한 필드에서의 값은 이하의 방법으로 인터페이스 타입을 특정한다. 만일 비트 7이 '0'과 같다면 타입 핸드오프 요청은 순방향 링크에 대한 것이며, 만일 '1'과 같다면 타입 핸드오프 요청은 역방향 링크에 대한 것이다. 비트 위치 6 내지 3은 원하는 대로 다른 핸드오프 타입을 설계하는데 사용하기 위해 일반적으로 보존되며, 통상 0으로 설정된다. 그러나 비트 위치 2 내지 0은 음의 승인을 나타내는 '0'의 값으로 사용될 인터페이스 타입을 한정하거나, 요청된 핸드오프가 실행되지 않음을 한정하는데 사용되며, '1', '2', '3' 및 '4'는 각각 타입 1, 타입 2, 타입 3 및 타입 4로의 핸드오프를 나타낸다. 5 내지 7의 값은 원할 경우 모드의 택일적 설계로 사용하기 위해 보존된다.

N. 실행 타입 핸드오프 패킷

1바이트 인터페이스 타입 필드는 사용할 새로운 인터페이스 타입을 나타낸다. 이러한 필드에 존재하는 값은 타입 핸드오프가 순방향 또는 역방향 링크인지를 결정하기 위해 비트 7의 값을 우선 사용함으로써 인터페이스 타입을 특정한다. '0'의 값은 타입 핸드오프 요청이 순방향 링크에 대한 것이며, '1'의 값은 역방향 링크에 대한 것이다. 비트 6 내지 3은 미래의 사용을 위해 보존되며, 일반적으로 0의 값으로 설정된다. 그러나 비트 2 내지 0은 사용될 인터페이스 타입을 한정하는데 사용되며, 값 1, 2, 3 및 4는 핸드오프의 사용을 타입 1, 타입 2, 타입 3 및 타입 4 모드로 각각 특정한다. 이러한 비트에 대한 값 0 및 5 내지 7의 사용은 미래의 사용을 위해 보존된다.

O. 순방향 오디오 채널 인에이블 패킷에 대해

오디오 채널 인에이블 마스크 필드(1바이트)는 오디오 채널이 클라이언트에서 인에이블되는 것을 나타내는 플래그의 그룹을 포함한다. 1로 설정된 비트는 대응하는 채널을 인에이블시키고, 0으로 설정된 비트는 대응하는 채널을 디세이블 시키며, 비트 0 내지 5는 채널 0 내지 5를 지정하는데, 이들은 좌측 정면, 우측 정면, 좌측 후면, 우측 후면, 정면 중앙 및 서브 우퍼 채널을 각각 어드레싱한다. 비트 6 및 7은 미래의 사용을 위해 보존되며, 한편으로 일반적으로 0으로 설정된다.

P. 역방향 오디오 샘플 레이트 패킷

오디오 샘플 레이트 필드(1 바이트)는 디지털 오디오 샘플 레이트를 특정한다. 이러한 필드에 대한 값은 상이한 레이트로 할당되는데, 0, 1, 2, 3, 4, 5, 6, 7 및 8은 각각 8,000, 16,000, 24,000, 32,000, 40,000, 48,000, 11,025, 22,025 및 44,100 초당 샘플(SPS)로 할당되며, 비트 9 내지 254는 원하는 택일적 레이트로 사용하기 위해 보존되며, 이들은 일반적으로 '0'으로 설정된다. 값 255는 역방향 링크 오디오 스트림을 디스에이블하는데 사용된다.

샘플 포맷 필드(1바이트)는 디지털 오디오 샘플의 포맷을 특정한다. 비트[1:0]가 '0'과 같다면, 디지털 오디오 샘플은 선형 포맷이며, '1'과 같다면, 디지털 오디오 샘플은 μ -Law 포맷이며, 2와 같다면 디지털 오디오 샘플은 A-Law 포맷이다. 비트 [7:2]는 원할 경우 오디오 포맷을 지정하는데 선택적인 사용을 위해 보존되며, 일반적으로 0으로 설정된다.

Q. 디지털 콘텐츠 보호 오버헤드 패킷에 대해

콘텐츠 보호 타입 필드(1바이트)는 사용된 디지털 콘텐츠 보호 방법을 특정한다. '0'의 값은 디지털 송신 콘텐츠 보호 (DTCP)를 나타내며, 1의 값은 고대역폭 디지털 콘텐츠 보호 시스템(HDCP)을 나타낸다. 2 내지 255 범위의 값은 일반적으로 특정되지 않지만, 원할 경우 택일적 보호 방식으로 사용을 위해 보존된다. 콘텐츠 보호 오버헤드 메시지 필드는 호스트와 클라이언트 사이에 전송된 콘텐츠 보호 메시지를 포함하는 가변 길이 필드이다.

R. 투명 컬러 인에이블 패킷에 대해

투명 컬러 인에이블 필드(1 바이트)는 투명 컬러 모드가 인에이블 또는 디스에이블될 때를 특정한다. 비트 0이 0과 같다면 투명 컬러 모드는 디스에이블되고, 1과 같다면 투명 컬러 모드는 인에이블되며 투명 컬러는 이하의 두 파라미터에 의해 특정된다. 이러한 바이트의 비트 1 내지 7은 미래의 사용을 위해 보존되며 일반적으로 0으로 설정된다.

비디오 데이터 포맷 기술자 필드(2 바이트)는 픽셀 영역 채움 값의 포맷을 특정한다. 도 11은 비디오 데이터 포맷 기술자가 어떻게 코딩되는 지를 나타낸다. 포맷은 일반적으로 비디오 스트림 패킷의 동일 필드와 동일하다.

픽셀 영역 채움 값 필드는 앞서 특정된 윈도우로 채워질 픽셀 값에 대해 할당된 4 바이트를 사용한다. 픽셀의 포맷은 비디오 데이터 포맷 기술자 필드에서 특정된다.

S. 왕복(round trip) 지연 측정 패킷에 대해

2 바이트 패킷 길이 필드는 패킷 길이 필드를 포함하지 않는 패킷에서 바이트의 전체 수를 특정하고, 일 실시예에서 159의 고정된 길이를 갖도록 선택된다. 2 바이트 패킷 타입 필드는 패킷을 왕복 지연 측정 패킷으로 식별하는 82의 값으로 이러한 패킷 타입을 식별한다. hClient ID 필드는 전술한 바와 같이 Client ID로서의 미래의 사용을 위해 보존되며, 일반적으로 0으로 설정된다.

일 실시예에서, 파라미터 CRC 필드(2 바이트)는 패킷 길이로부터 패킷 타입까지의 모든 바이트의 16비트 CRC를 포함한다. 만일 이러한 CRC가 체크에 실패하면 전체 패킷이 폐기된다.

보호 시간 1 필드(여기서 64 바이트)는 호스트의 라인 드라이버가 디스에이블되기 전에 클라이언트의 MDDI_Data 라인 드라이버가 인에이블되도록 사용된다. 클라이언트는 보호 시간 1의 비트 0 동안 자신의 MDDI_Data 라인 드라이버를 인에이블하며, 호스트는 보호 시간 1의 최종 비트에 앞서 완전히 디스에이블되도록 자신의 라인 드라이버를 디스에이블시킨다. 호스트 및 클라이언트는 모두 이들이 디스에이블되지 않은 경우 보호 시간 1 동안 로직 0 레벨을 구동시킨다. 이러한 필드의 다른 목적은 모든 MDDI_Data 신호가, 클라이언트가 호스트의 라인 드라이버를 디스에이블하기 전에 MDDI_Stb만을 사용하여 클럭 또는 클럭 신호를 복구하기 시작하도록, 충분한 시간 동안 로직 0 레벨에 있는 것을 보장하는 것이다.

측정 주기 필드는 순방향 링크에 사용된 절반의 데이터 전송률로 2 바이트의 0xff, 및 30 바이트의 0x0으로 클라이언트가 응답하게 하도록 사용된 64 바이트의 윈도우이다. 이러한 데이터 전송률은 1의 역방향 링크 레이트 제수에 대응한다. 클라이언트는 측정기간의 시작임을 감지한 때 이러한 응답을 즉시 반환시킨다. 클라이언트로부터의 이러한 응답은 호스트에서 측정기간의 첫 번째 비트의 시작 이후 클라이언트에서의 논리 지연에 더하여 링크의 정확하게 왕복 지연에 호스트에서 수신될 것이다.

모든 제로 1 필드(2 바이트)는 호스트 및 클라이언트에서 MDDI_Data 라인 드라이버가 MDDI_Data가 언제나 구동되도록 오버랩하게 하는 제로들을 포함한다. 호스트는 MDDI_Data 라인 드라이버를 보호 시간 2의 비트 0 동안 인에이블시키고, 클라이언트는 또한 측정기간의 종료에서 행한 것과 같이 로직 0 레벨로 신호를 구동시킨다.

보호 시간 2 필드(64 바이트)에서의 값은 왕복 지연이 측정 주기에서 측정될 수 있는 최대 양에 있는 경우, 클라이언트에 의해 구동된 측정 주기의 오버랩을 가능하게 한다. 클라이언트는 보호 시간 2의 0 비트 동안 자신의 라인 드라이버를 디스에이블시키며, 호스트는 보호 시간 2의 최종 비트 후 자신의 라인 드라이버를 즉시 인에이블시킨다. 호스트 및 클라이언트 모두는 이들이 디스에이블된 경우 보호 시간 2 동안 로직 0 레벨을 구동시킨다. 이러한 필드의 다른 목적은 호스트에 대한 라인 드라이버를 인에이블시킨 후 클라이언트가 MDDI_Data 0 및 MDDI_Stb를 사용하여 클럭 신호를 복구하기 시작하도록 하는 충분한 시간 동안 모든 MDDI_Data 신호가 로직 0 레벨에 있는 것을 보장하는 것이다.

T. 순방향 링크 스큐(skew) 캘리브레이션 패킷에 대해

일 실시예에서, 파라미터 CRC 필드(2 바이트)는 패킷 길이로부터 패킷 타입까지의 모든 바이트의 16비트 CRC를 포함한다. 만일 이러한 CRC가 체크에 실패하면 전체 패킷은 폐기된다.

모두 제로 1 필드는 파라미터 CRC 필드의 끝부분에서 MDDI_Stb 상에 트랜지션이 존재하도록 보장하기 위해 1 바이트를 사용한다.

캘리브레이션 데이터 시퀀스 필드는 각각의 데이터 주기에서 MDDI_Data 신호들이 토글링(toggle)하도록 하는 데이터 시퀀스를 포함한다. 캘리브레이션 데이터 시퀀스 필드의 길이는 순방향 링크를 통해 사용되는 인터페이스에 의해 결정된다. 캘리브레이션 데이터 시퀀스의 프로세싱 동안에, MDDI 호스트 제어기는 모든 MDDI_Data 신호들을 스트로브(strobe) 신호와 동일하게 설정한다. 클라이언트 클럭 복원 회로는 데이터 클럭을 복원하기 위해 MDDI_Stb Xor MDDI_Data0보다는 오직 MDDI_Stb만을 사용하여야 하며, 캘리브레이션 데이터 시퀀스 필드는 클라이언트 디스플레이에 의해 수신된다. 캘리브레이션 데이터 시퀀스 필드의 시작부에서 MDDI_Stb 신호의 정확한 위상에 의존하여, 캘리브레이션 데이터 시퀀스는 일반적으로 이러한 패킷이 전송될 때 사용되는 인터페이스 타입에 기반하여 다음 중 하나가 될 것이다:

타입 I-(64 바이트 데이터 시퀀스) 0xaa, 0xaa ... or 0x55, 0x55...

타입 II-(128 바이트 데이터 시퀀스) 0xcc, 0xcc ... or 0x33, 0x33...

타입 III-(256 바이트 데이터 시퀀스) 0xf0, 0xf0 ... or 0x0f, 0x0f...

타입 IV-(512 바이트 데이터 시퀀스) 0xff, 0x00, 0xff, 0x00 ... or 0x00, 0xff, 0x00, 0xff...

타입 1 및 타입 2 인터페이스에 대한 가능한 MDDI_Data 및 MDDI_Stb 파형의 예는 각각 도 62a 및 62b에 도시된다.

XVII. 결론

다양한 실시예가 기술되었지만, 이는 단지 예일 뿐 본 발명을 한정하지 않는다는 것을 이해해야 한다. 따라서 본 발명의 사상은 실시예에 의해 한정되지 않으며 이하의 첨부된 청구항에 의해서만 한정된다.

(57) 청구의 범위

청구항 1.

통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 디지털 프리젠테이션 데이터를 고속으로 전송하기 위한 디지털 데이터 인터페이스로서,

함께 연결되어, 상기 통신 경로를 통해 호스트와 클라이언트 사이에 미리 선택된 디지털 제어 및 프리젠테이션 데이터 세트를 전달하기 위한 통신 프로토콜을 형성하는 다수의 패킷 구조; 및

상기 통신 경로를 통해 상기 클라이언트에 연결되는 상기 호스트 장치에 존재하며, 상기 통신 프로토콜을 형성하는 패킷들을 생성, 전송 및 수신하며, 하나 이상의 데이터 패킷 타입으로 디지털 프리젠테이션 데이터를 형성하도록 구성된 적어도 하나의 링크 제어를 포함하는, 디지털 데이터 인터페이스.

청구항 2.

제 1 항에 있어서,

상기 호스트와 클라이언트 사이에 전달되는 미디어 프레임 내에서 함께 그룹화되는 패킷들을 더 포함하며, 미리 결정된 수의 상기 패킷에 의해 미리 정해진 고정 길이를 갖는 상기 미디어 프레임들은 상이한 가변 길이를 갖는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 3.

제 1 항에 있어서,

상기 호스트로부터의 패킷 전송의 최초에 배치되는 서브 프레임 헤더 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 4.

제 1 항에 있어서,

상기 링크 제어기는 호스트 링크 제어기이고,

상기 통신 경로를 통해 상기 호스트에 연결되는 상기 클라이언트 장치에 존재하며, 상기 통신 프로토콜을 형성하는 패킷들을 생성, 전송 및 수신하며, 하나 이상의 데이터 패킷 타입으로 디지털 프리젠테이션 데이터를 형성하도록 구성된 적어도 하나의 클라이언트 링크 제어를 더 포함하는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 5.

제 1 항에 있어서,

클라이언트 사용자에게 프리젠테이션하기 위해 순방향 링크를 통해 상기 호스트로부터 상기 클라이언트로 데이터를 전송하기 위한 하나 이상의 비디오 타입 데이터용 비디오 스트림 패킷 및 오디오 타입 데이터용 오디오 스트림 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 6.

제 2 항에 있어서,

각각 다른 최대 비트 수의 데이터를 소정 시간대에 병렬로 전송할 수 있게 하며, 각각 상기 호스트와 클라이언트 링크 드라이버 사이의 협의에 의해 선택 가능한 다수의 전송 모드를 더 포함하는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 7.

제 1 항에 있어서,

컬러맵, 비트 블록 전송, 비트맵 영역 채움, 비트맵 패턴 채움 및 투명색 인에이블 타입 패킷으로 이루어진 그룹으로부터 선택된 비디오 정보를 전송하는데 유용한 다수의 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 8.

제 1 항에 있어서,

상기 호스트에 의해 생성되어 데이터가 없는 순방향 링크 전송 기간을 차지하는 채움 타입 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 9.

제 1 항에 있어서,

인터페이스-사용자 정의 데이터를 전송하기 위한 사용자 정의 스트림 타입 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 10.

제 1 항에 있어서,

상기 통신 경로를 통한 어느 한 방향으로의 데이터 전송을 종료하기 위해 상기 호스트에 의해 상기 클라이언트에 전송되는 링크 셋다운 타입 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 11.

제 1 항에 있어서,

상기 클라이언트가 절전 모드 상태에서부터 상기 호스트를 웨이크업하는 수단을 더 포함하는 것을 특징으로 하는 디지털 데이터 인터페이스.

청구항 12.

사용자에게 프리젠테이션 하기 위해 통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 고속으로 디지털 데이터를 전송하는 방법으로서,

미리 정의된 다수의 패킷 구조 중 적어도 하나를 생성하고 함께 연결하여 미리 정의된 통신 프로토콜을 형성하는 단계;

상기 통신 프로토콜을 이용하여 상기 통신 경로를 통해 상기 호스트와 상기 클라이언트 장치 사이에 미리 선택된 디지털 제어 및 프리젠테이션 데이터 세트를 전달하는 단계;

상기 호스트 장치에 존재하며, 상기 통신 프로토콜을 형성하는 패킷들을 생성, 전송 및 수신하고, 하나 이상의 데이터 패킷 타입으로 디지털 프리젠테이션 데이터를 형성하도록 구성된 적어도 하나의 호스트 링크 제어기를 상기 통신 경로를 통해 상기 클라이언트에 연결하는 단계; 및

상기 링크 제어를 이용하여 상기 통신 경로를 통해 데이터를 패킷 형태로 전송하는 단계를 포함하는, 디지털 데이터 전송 방법.

청구항 13.

제 12 항에 있어서,

상기 호스트와 클라이언트 사이의 통신용 미디어 프레임 내에서 상기 패킷들을 함께 그룹화하는 단계를 더 포함하며, 미리 결정된 수의 상기 패킷에 의해 미리 정해진 고정 길이를 갖는 상기 미디어 프레임들은 상이한 가변 길이를 갖는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 14.

제 12 항에 있어서,

서브 프레임 헤더 타입 패킷에 의해 상기 호스트로부터의 패킷 전송을 시작하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 15.

제 12 항에 있어서,

상기 통신 링크를 통해 상기 호스트와 클라이언트 사이에 양방향으로 정보를 전송하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 16.

제 12 항에 있어서,

상기 클라이언트 장치에 존재하며, 상기 통신 프로토콜을 형성하는 패킷들을 생성, 전송 및 수신하고, 하나 이상의 데이터 패킷 타입으로 디지털 프리젠테이션 데이터를 형성하도록 구성된 적어도 하나의 클라이언트 링크 제어를 상기 통신 경로를 통해 상기 호스트 장치에 연결하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 17.

제 16 항에 있어서,

상기 호스트 링크 제어기는 하나 이상의 차동 라인 드라이버를 포함하고,

상기 클라이언트 링크 제어기는 상기 통신 경로에 연결된 하나 이상의 차동 라인 수신기를 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 18.

제 12 항에 있어서,

상기 클라이언트가 상기 인터페이스를 통해 제공할 수 있는 데이터 타입 및 데이터 전송률을 결정하기 위해 상기 호스트 링크 제어기에 의해 상기 클라이언트로부터의 디스플레이 성능 정보를 요청하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 19.

제 12 항에 있어서,

상기 통신 경로의 일부로서 상기 링크 제어기들 각각에 의해 USB 데이터 인터페이스를 작동시키는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 20.

제 12 항에 있어서,

상기 패킷들은 각각 패킷 길이 필드, 하나 이상의 패킷 데이터 필드 및 순환 중복 검사 필드를 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 21.

제 13 항에 있어서,

각각 다른 최대 비트 수의 데이터를 소정 시간대에 병렬로 전송할 수 있게 하는 각 방향으로의 다수의 전송 모드 중 하나의 사용을 상기 호스트와 클라이언트 링크 드라이버 사이에 협의하는 단계; 및

데이터 전송시 상기 전송 모드들 사이를 동적으로 조정하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 22.

제 12 항에 있어서,

다수의 패킷 중 하나 이상을 이용하여, 컬러맵, 비트 블록 전송, 비트맵 영역 채움, 비트맵 패턴 채움 및 투명색 인에이블 타입 패킷으로 이루어진 그룹으로부터 선택된 비디오 정보를 전송하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 23.

제 12 항에 있어서,

상기 호스트에 의해 데이터가 없는 순방향 링크 전송 기간을 차지하는 채움 타입 패킷을 생성하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 24.

제 12 항에 있어서,

상기 호스트에 의해 상기 클라이언트에 전송되는 링크 셋다운 타입 패킷을 이용하여 상기 통신 경로를 통한 어느 한 방향으로의 데이터 전송을 종료하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 25.

제 12 항에 있어서,

상기 클라이언트와의 통신에 의해 절전 모드 상태에서부터 상기 호스트를 웨이크업하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 26.

사용자에게 프리젠테이션 하기 위해 통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 고속으로 디지털 데이터를 전송하기 위한 장치로서,

미리 정의된 다수의 패킷 구조 중 하나 이상을 생성하고 함께 연결하여 미리 정의된 통신 프로토콜을 형성하고, 상기 통신 프로토콜을 이용하여 상기 통신 경로를 통해 상기 호스트와 상기 클라이언트 장치 사이에 미리 선택된 디지털 제어 및 프리젠테이션 데이터 세트를 전달하기 위해 상기 호스트 장치에 배치된 적어도 하나의 호스트 링크 제어기;

상기 클라이언트 장치에 배치되며, 상기 통신 경로를 통해 상기 호스트 링크 제어기에 연결되는 적어도 하나의 클라이언트 제어기; 및

상기 통신 프로토콜을 형성하는 패킷들을 생성, 전송 및 수신하고, 하나 이상의 데이터 패킷 타입으로 디지털 프리젠테이션 데이터를 형성하도록 구성된 각각의 링크 제어기를 포함하는, 디지털 데이터 전송 장치.

청구항 27.

제 26 항에 있어서,

상기 호스트 제어기는 상태 머신인 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 28.

제 26 항에 있어서,

상기 호스트 제어기는 범용 신호 프로세서인 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 29.

제 26 항에 있어서,

상기 호스트로부터의 패킷 전송의 시작에 서브 프레임 헤더 타입 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 30.

제 26 항에 있어서,

상기 링크 제어기는 상기 통신 링크를 통해 상기 호스트와 클라이언트 장치 사이에 양방향으로 정보를 전송하도록 구성되는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 31.

제 30 항에 있어서,

상기 호스트 제어기는 하나 이상의 차동 라인 드라이버를 포함하고,

상기 클라이언트 수신기는 상기 통신 경로에 연결된 하나 이상의 차동 라인 수신기를 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 32.

제 26 항에 있어서,

클라이언트 사용자에게 프리젠테이션하기 위해 상기 호스트로부터 상기 클라이언트로 데이터를 전송할 때 비디오 타입 데이터용 비디오 스트림 타입 패킷 및 오디오 타입용 오디오 스트림 타입 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 33.

제 26 항에 있어서,

상기 클라이언트로부터 상기 호스트로 데이터를 전송하기 위한 하나 이상의 역방향 링크 암호화 타입 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 34.

제 33 항에 있어서,

클라이언트 링크 제어기로부터 상기 호스트 링크 제어기로 디스플레이 또는 프리젠테이션 용량을 전달하기 위한 적어도 하나의 디스플레이 용량 타입 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 35.

제 26 항에 있어서,

상기 패킷들은 각각 패킷 길이 필드, 하나 이상의 패킷 데이터 필드 및 순환 중복 검사 필드를 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 36.

제 26 항에 있어서,

상기 호스트 및 클라이언트 링크 제어기는 각각 다른 최대 비트 수의 데이터를 소정 시간대에 병렬로 전송할 수 있게 하는 각 방향으로의 다수의 전송 모드 중 하나를 사용하도록 구성되며, 데이터 전송시 상기 전송 모드들 사이를 동적으로 조정할 수 있는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 37.

제 26 항에 있어서,

컬러맵, 비트 블록 전송, 비트맵 영역 채움, 비트맵 패턴 채움 및 투명색 인에이블 타입 패킷으로 이루어진 그룹으로부터 선택된 비디오 정보를 전송하기 위한 다수의 패킷 중 하나 이상을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 38.

제 26 항에 있어서,

데이터가 없는 순방향 링크 전송 기간을 차지하는 채움 타입 패킷을 상기 호스트에 의해 전송하기 위한 채움 타입 패킷을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 39.

제 26 항에 있어서,

상기 호스트 제어기는 상기 통신 경로를 통한 어느 한 방향으로의 데이터 전송을 종료하기 위한 링크 셧다운 타입 패킷을 상기 클라이언트에 전송하도록 구성되는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 40.

사용자에게 프리젠테이션 하기 위해 통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 고속으로 디지털 데이터를 전송하기 위한 전자 시스템용 컴퓨터 프로그램 제품으로서,

애플리케이션 프로그램이 컴퓨터 시스템에서 실행되게 하는 컴퓨터 판독 가능 프로그램 코드 수단이 구현된 컴퓨터 사용 가능 매체를 포함하며, 상기 컴퓨터 판독 가능 프로그램 코드 수단은,

상기 컴퓨터 시스템이 미리 정의된 다수의 패킷 구조 중 하나 이상을 생성하고 함께 연결하여 미리 정의된 통신 프로토콜을 형성하게 하는 컴퓨터 판독 가능 제 1 프로그램 코드 수단;

상기 컴퓨터 시스템이 상기 통신 프로토콜을 이용하여 상기 통신 경로를 통해 상기 호스트와 상기 클라이언트 장치 사이에 미리 선택된 디지털 제어 및 프리젠테이션 데이터 세트를 전달하게 하는 컴퓨터 판독 가능 제 2 프로그램 코드 수단;

상기 컴퓨터 시스템이, 상기 호스트 장치에 존재하는 적어도 하나의 호스트 링크 제어기를 상기 통신 경로를 통해 상기 클라이언트 장치에 배치된 적어도 하나의 클라이언트 제어기에 연결하게 하는 컴퓨터 판독 가능 제 3 프로그램 코드 수단 - 상기 링크 제어기는 상기 통신 프로토콜을 형성하는 패킷들을 생성, 전송 및 수신하고, 하나 이상의 데이터 패킷 타입으로 디지털 프리젠테이션 데이터를 형성하도록 구성됨 -; 및

상기 컴퓨터 시스템이 상기 링크 제어기를 이용하여 상기 통신 경로를 통해 데이터를 패킷 형태로 전송하게 하는 컴퓨터 판독 가능 제 4 프로그램 수단을 포함하는, 컴퓨터 프로그램 제품.

청구항 41.

사용자에게 프리젠테이션 하기 위해 통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 고속으로 디지털 데이터를 전송하는 장치로서,

미리 정의된 다수의 패킷 구조 중 하나 이상을 생성하고 함께 연결하여 미리 정의된 통신 프로토콜을 형성하는 수단;

상기 통신 프로토콜을 이용하여 상기 통신 경로를 통해 상기 호스트와 상기 클라이언트 장치 사이에 미리 선택된 디지털 제어 및 프리젠테이션 데이터 세트를 전달하는 수단;

상기 통신 프로토콜을 형성하는 패킷들을 생성, 전송 및 수신하고, 하나 이상의 데이터 패킷 타입으로 디지털 프리젠테이션 데이터를 형성하도록 각각 구성된 상기 호스트 및 클라이언트 각각의 적어도 2개의 링크 제어기를 상기 통신 경로를 통해 서로 연결하는 수단; 및

상기 링크 제어기를 이용하여 상기 통신 경로를 통해 데이터를 패킷 형태로 전송하는 수단을 포함하는, 디지털 데이터 전송 장치.

청구항 42.

제 41 항에 있어서,

서브 프레임 헤더 타입 패킷에 의해 상기 호스트로부터의 패킷 전송을 시작하는 수단을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 43.

제 41 항에 있어서,

상기 통신 링크를 통해 상기 호스트와 클라이언트 사이에 양방향으로 정보를 전송하는 수단을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 44.

제 41 항에 있어서,

상기 클라이언트가 상기 인터페이스를 통해 제공할 수 있는 데이터 타입 및 데이터 전송률을 결정하기 위해 상기 호스트 링크 제어기에 의해 상기 클라이언트로부터의 디스플레이 성능 정보를 요청하는 수단을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 45.

제 44 항에 있어서,

적어도 하나의 디스플레이 용량 타입 패킷을 이용하여 클라이언트 링크 제어기로부터 상기 호스트 링크 제어기로 디스플레이 또는 프리젠테이션 용량을 전달하는 수단을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 46.

제 42 항에 있어서,

각각 다른 최대 비트 수의 데이터를 소정 시간대에 병렬로 전송할 수 있게 하는 각 방향으로의 다수의 전송 모드 중 하나의 사용을 상기 호스트와 클라이언트 링크 드라이버 사이에 협의하는 수단; 및

데이터 전송시 상기 전송 모드들 사이를 동적으로 조정하는 수단을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 47.

제 41 항에 있어서,

다수의 패킷 중 하나 이상을 이용하여, 컬러맵, 비트 블록 전송, 비트맵 영역 채움, 비트맵 패턴 채움 및 투명색 인에이블 타입 패킷으로 이루어진 그룹으로부터 선택된 비디오 정보를 전송하는 수단을 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 장치.

청구항 48.

통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 고속으로 디지털 데이터를 전송하기 위한 전자 시스템용 프로세서로서, 상기 프로세서는,

미리 정의된 다수의 패킷 구조 중 적어도 하나를 생성하고 함께 연결하여 미리 정의된 통신 프로토콜을 형성하고;

하나 이상의 데이터 패킷 타입으로 디지털 프리젠테이션 데이터를 형성하고;

상기 통신 프로토콜을 이용하여 상기 통신 경로를 통해 상기 호스트와 상기 클라이언트 장치 사이에 미리 선택된 디지털 제어 및 프리젠테이션 데이터 세트를 전달하고;

상기 통신 경로를 통해 데이터를 패킷 형태로 전송하도록 구성되는, 프로세서.

청구항 49.

통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 고속으로 디지털 데이터를 전송하기 위한 전자 시스템에서 동기 포착에 이용되는 상태 머신으로서, 상기 상태 머신은,

적어도 하나의 비동기 프레임 상태 동기화 상태, 적어도 2개의 동기 포착 상태 동기화 상태 및 적어도 3개의 동기 상태 동기화 상태를 갖도록 구성되는, 상태 머신.

청구항 50.

통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 고속으로 디지털 데이터를 전송하기 위한 전자 시스템에서 동기 포착에 이용되는 상태 머신으로서, 상기 상태 머신은,

적어도 하나의 동기 포착 상태 동기화 상태 및 적어도 2개의 동기 상태 동기화 상태를 갖도록 구성되는, 상태 머신.

청구항 51.

제 50 항에 있어서,

동기 포착 상태와 제 1 동기 상태 사이의 변화를 위한 하나의 조건은 상기 통신 링크에서의 동기화 패턴 존재 검출인 것을 특징으로 하는 상태 머신.

청구항 52.

제 51 항에 있어서,

동기 포착 상태와 제 1 동기 상태 사이의 변화를 위한 제 2 조건은 프레임 경계에서 양호한 CRC 값 또는 서브 프레임 헤더 패턴의 존재 검출인 것을 특징으로 하는 상태 머신.

청구항 53.

제 50 항에 있어서,

제 1 동기 상태와 동기 포착 상태 사이의 변화를 위한 하나의 조건은 서브 프레임 경계에서 잘못된 CRC 값 또는 비동기 패턴의 존재 검출인 것을 특징으로 하는 상태 머신.

청구항 54.

제 50 항에 있어서,

제 1 동기 상태와 제 2 동기 상태 사이의 변화를 위한 하나의 조건은 서브 프레임 경계에서 잘못된 CRC 값 또는 비동기 패턴의 존재 검출인 것을 특징으로 하는 상태 머신.

청구항 55.

제 50 항에 있어서,

동기 포착 상태와 제 1 동기 상태 사이의 변화를 위한 하나의 조건은 상기 통신 링크에서의 동기 패턴 존재 검출 및 양호한 패턴 CRC 값의 존재 검출인 것을 특징으로 하는 상태 머신.

청구항 56.

제 50 항에 있어서,

제 1 동기 상태와 동기 포착 상태 사이의 변화 조건은 패턴에서 잘못된 CRC 값의 존재 검출인 것을 특징으로 하는 상태 머신.

청구항 57.

통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 고속으로 디지털 데이터를 전송하기 위한 전자 시스템에서 동기 포착에 이용되는 상태 머신으로서, 상기 상태 머신은,

적어도 하나의 동기 포착 상태 동기화 상태 및 적어도 2개의 동기 상태 동기화 상태를 갖도록 구성되며, 제 1 동기 상태와 동기 포착 상태 사이의 직접 변경 조건은 임의의 일련의 패킷에서의 잘못된 CRC 값의 존재 검출인, 상태 머신.

청구항 58.

제 57 항에 있어서,

제 1 동기 상태와 동기 포착 상태 사이의 직접 변화 조건은 고유 워드가 도달하기로 예정된 시각에 나오지 않는 경우의 검출인 것을 특징으로 하는 상태 머신.

청구항 59.

제 26 항에 있어서,

적어도 10 클록 사이클 동안 데이터 라인을 하이(high) 상태로 구동함으로써 통신 링크를 웨이크업하고, 상기 호스트에 의해 상기 데이터 라인이 0인 것처럼 스트로브 신호의 전송을 시작하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 60.

제 59 항에 있어서,

상기 호스트가 150 클록 사이클 동안 상기 데이터 라인을 하이로 구동한 후 스트로브 신호를 계속해서 전송하면서, 상기 호스트에 의해 50 클록 사이클 동안 상기 데이터 라인을 로우(low)로 구동하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 61.

제 59 항에 있어서,

상기 호스트에 의해 제 1 서브 프레임 헤더 패킷의 전송을 시작하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 62.

제 60 항에 있어서,

상기 클라이언트에 의해 하이인 데이터 라인의 적어도 150개의 연속한 클록 사이클을 카운트하고, 이어서 로우인 데이터 라인의 적어도 50개의 연속한 클록 사이클을 카운트하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 63.

제 62 항에 있어서,

상기 클라이언트에 의해 상기 제 1 서브 프레임의 고유 워드를 검색하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 64.

제 60 항에 있어서,

상기 클라이언트가 하이인 데이터의 70개의 연속한 클록 사이클을 카운트한 후 상기 클라이언트에 의해 상기 데이터 라인을 하이로 구동하는 것을 중단하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 65.

제 64 항에 있어서,

상기 클라이언트에 의해 하이인 데이터 라인의 150 클록 사이클에 이르도록 하이인 데이터 라인의 다른 80개의 연속한 클록 사이클을 카운트하고, 로우인 데이터 라인의 50 클록 사이클을 검색하여 고유 워드를 찾는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 66.

제 26 항에 있어서,

역방향 타이밍 패킷 동안 상승 및 하강 에지에서 상기 데이터 라인을 샘플링함으로써, 상기 호스트에 의해 1이 샘플링될 때까지 발생하는 클록 사이클 수를 카운트하는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

청구항 67.

통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 디지털 데이터가 CRC 값을 갖는 패킷 형태로 전송되는 통신 시스템에서 에러 코드를 전송하는 방법으로서,

에러 존재를 검출하는 단계;

상기 에러에 대응하는 미리 결정된 에러 코드를 선택하는 단계; 및

CRC 값을 상기 코드에 겹쳐 쓰는 단계를 포함하는, 에러 코드 전송 방법.

청구항 68.

제 67 항에 있어서,

상기 에러가 정정될 때까지 전송되는 연속한 패킷들에 상기 CRC 값을 겹쳐 쓰는 단계를 더 포함하는 것을 특징으로 하는 에러 코드 전송 방법.

청구항 69.

사용자에게 프리젠테이션 하기 위해 통신 경로를 통해 호스트 장치와 클라이언트 장치 사이에 고속으로 디지털 데이터를 전송하는 방법으로서,

각각 적어도 하나의 CRC 필드를 포함하는 미리 정의된 다수의 패킷 구조 중 하나 이상을 생성하고 함께 연결하여 미리 정의된 통신 프로토콜을 형성하는 단계;

상기 통신 프로토콜을 이용하여 상기 통신 경로를 통해 상기 호스트와 상기 클라이언트 장치 사이에 미리 선택된 디지털 제어 및 프리젠테이션 데이터 세트를 전달하는 단계;

상기 호스트 장치에 존재하며, 상기 통신 프로토콜을 형성하는 패킷들을 생성, 전송 및 수신하고, 하나 이상의 데이터 패킷 타입으로 디지털 프리젠테이션 데이터를 형성하도록 구성된 적어도 하나의 호스트 링크 제어기를 상기 통신 경로를 통해 상기 클라이언트에 연결하는 단계;

상기 링크 제어기를 이용하여 상기 통신 경로를 통해 데이터를 패킷 형태로 전송하는 단계;

상기 통신 링크에 대한 에러 존재를 검출하는 단계;

상기 에러에 대응하는 미리 결정된 에러 코드를 선택하는 단계; 및

상기 코드에 CRC 값을 겹쳐 쓰는 단계를 포함하는, 디지털 데이터 전송 방법.

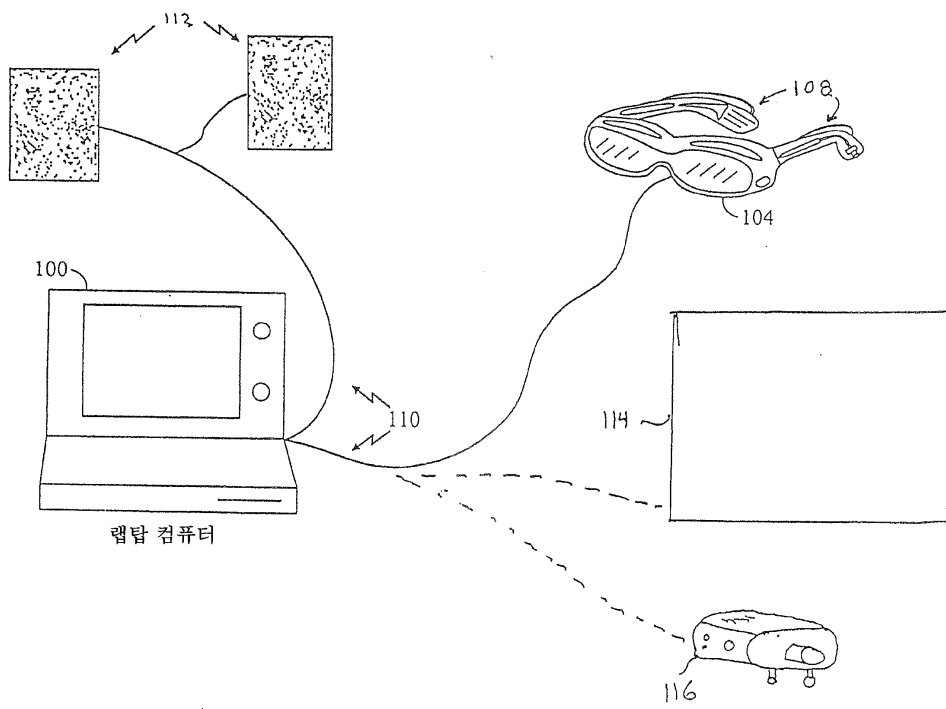
청구항 70.

제 69 항에 있어서,

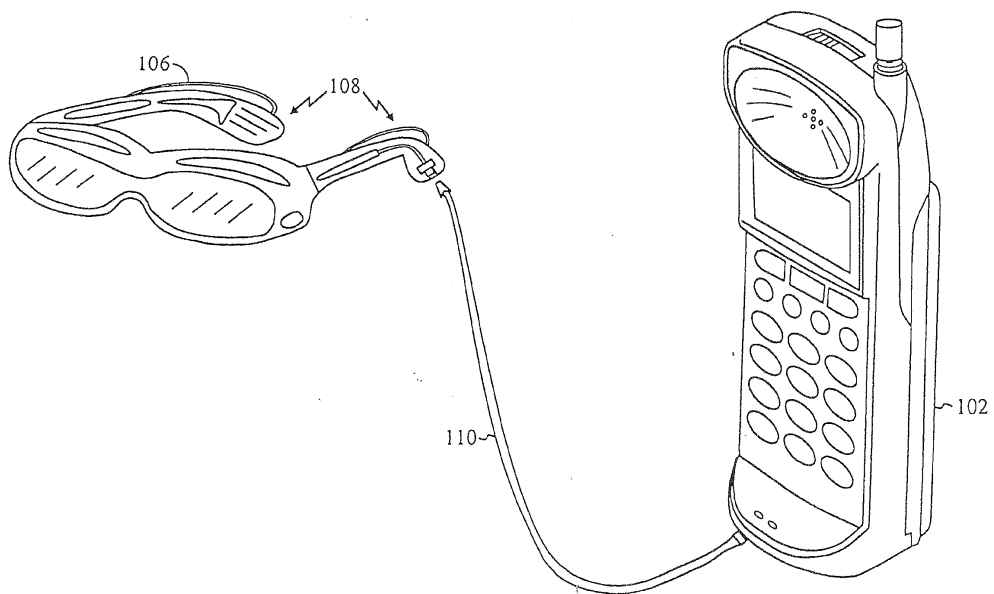
상기 에러가 정정될 때까지 전송되는 연속한 패킷들에 상기 CRC 값을 겹쳐 쓰는 단계를 더 포함하는 것을 특징으로 하는 디지털 데이터 전송 방법.

도면

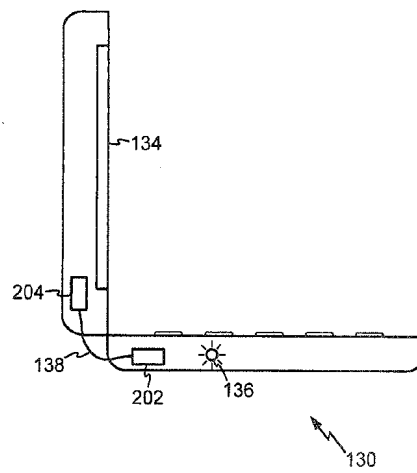
도면1a



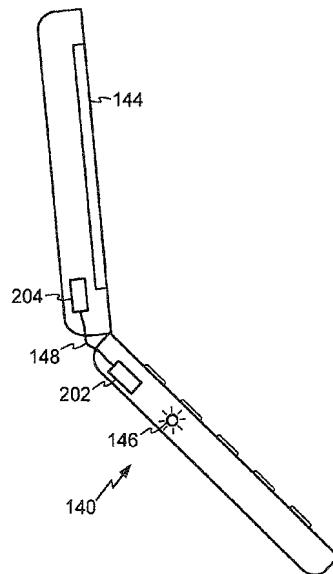
도면1b



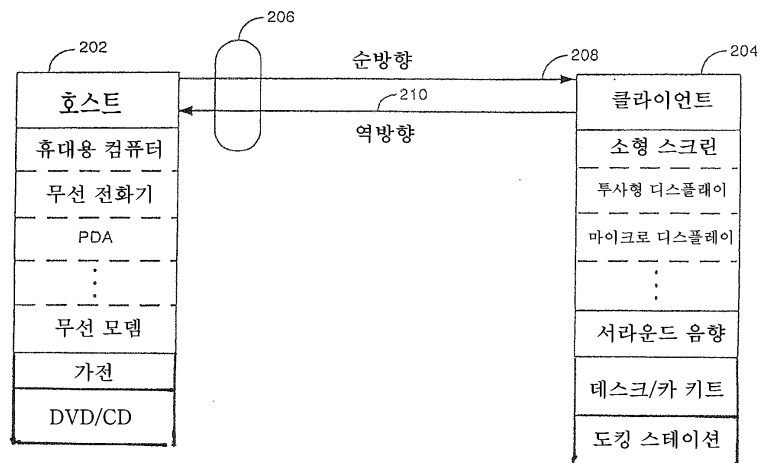
도면1c



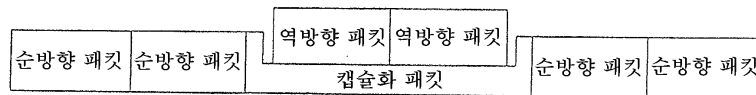
도면1d



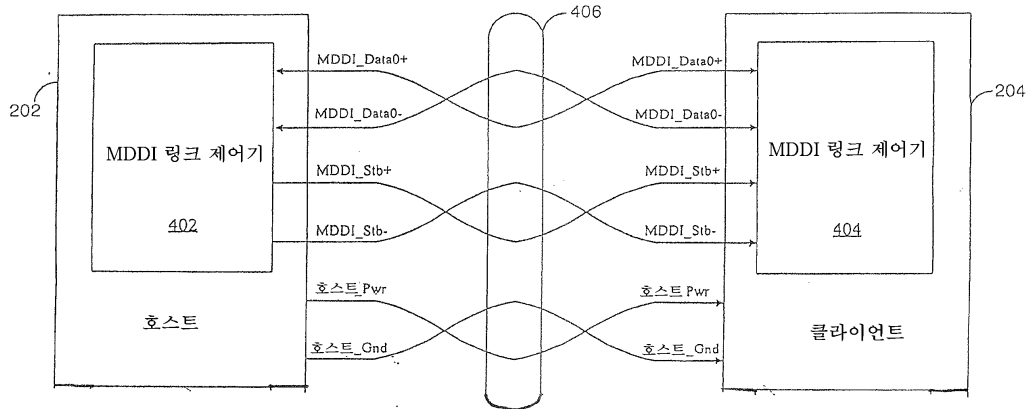
도면2



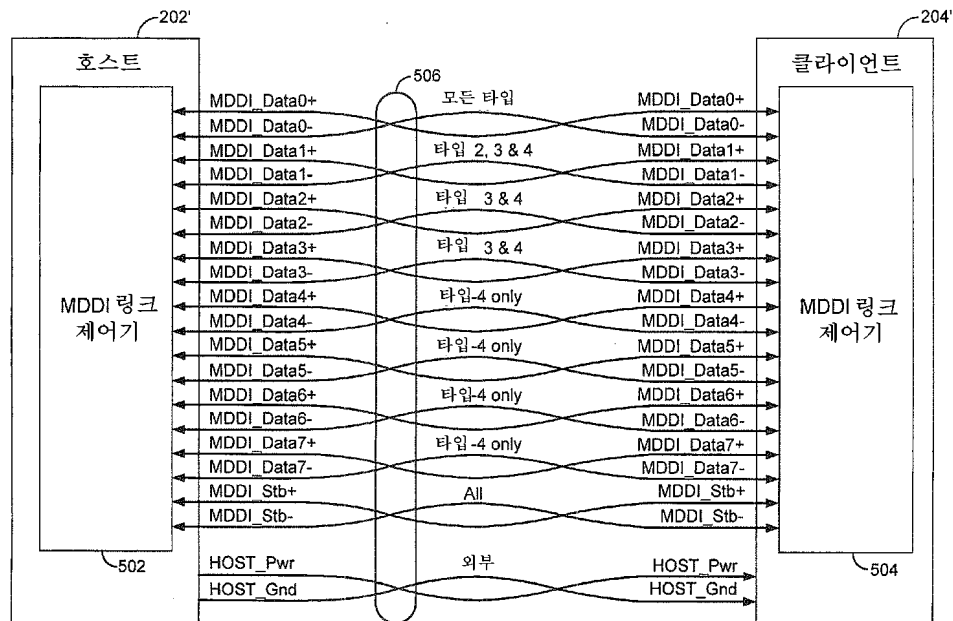
도면3



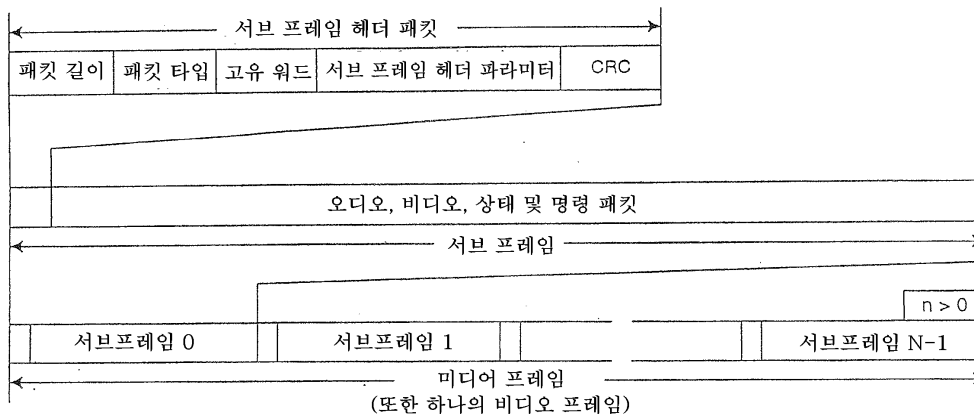
도면4



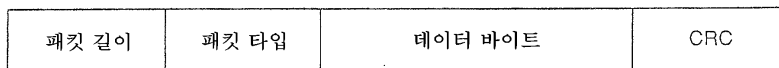
도면5



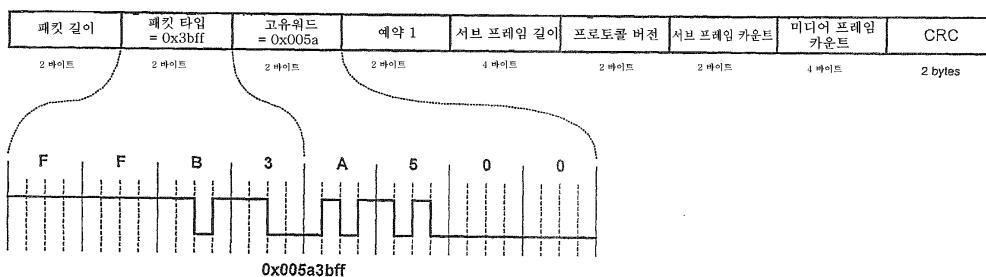
도면6



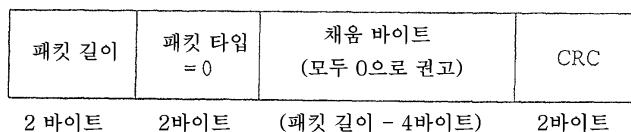
도면7



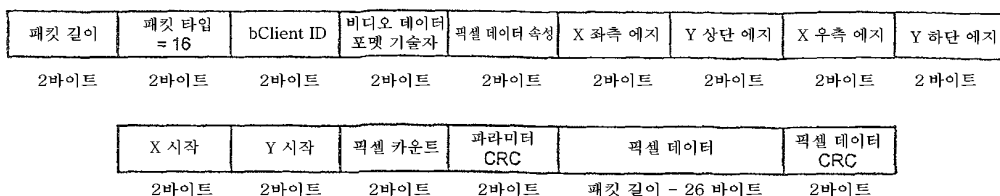
도면8



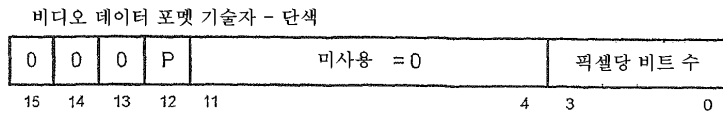
도면9



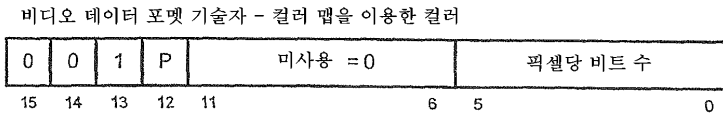
도면10



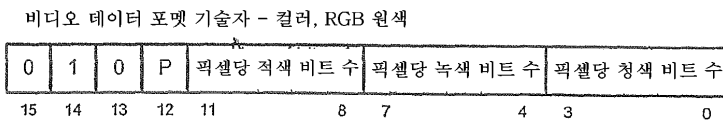
도면11a



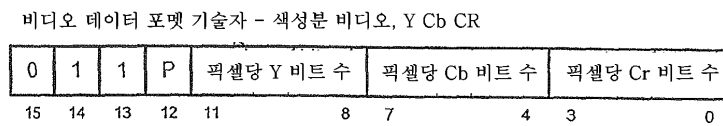
도면11b



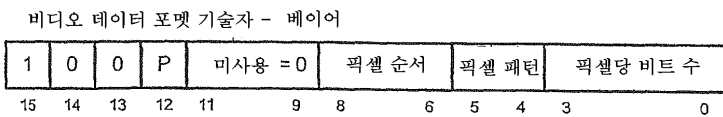
도면11c



도면11d

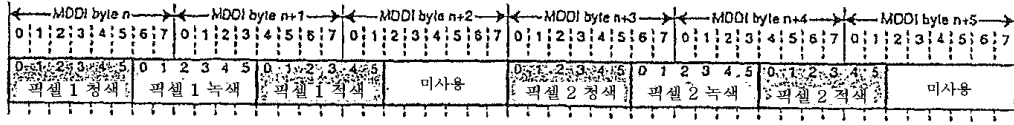


도면11e

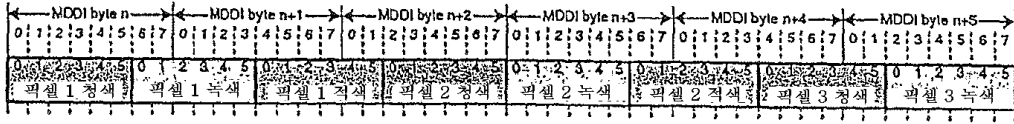


도면12

바이트 정렬된 RGB 픽셀(비압축), P = 0



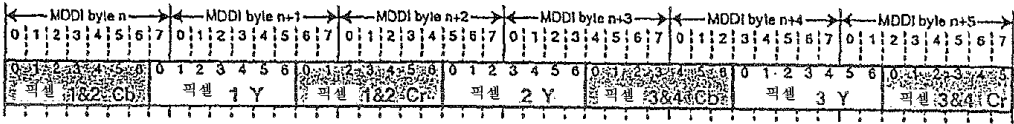
압축된 RGB 픽셀 데이터, P = 1



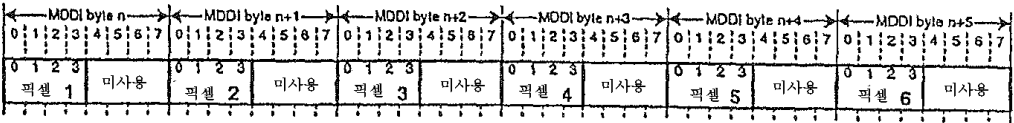
바이트 정렬된 Y Cr Cb 픽셀(비압축), P = 0



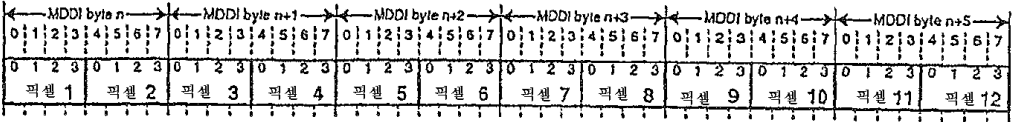
압축된 Y Cr Cb 픽셀 데이터, P = 1



바이트 정렬된 컬러 맵(팔레트) 또는 단색 픽셀(비압축), P = 0



압축된 컬러 맵(팔레트) 또는 단색 픽셀 데이터, P = 1



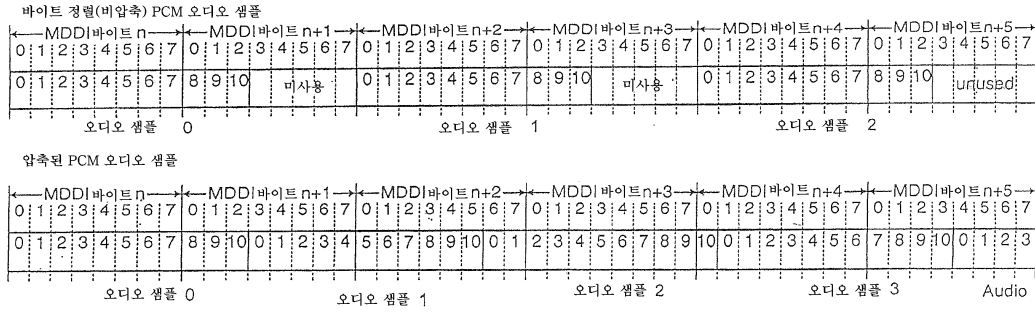
도면13

오디오 스트림 패킷

패킷 길이	패킷 타입 = 32	bClient ID	오디오채널 ID	예약 1	오디오 샘플 카운트
2 바이트	2 바이트	2 바이트	1 바이트	1 바이트	2 바이트

샘플당 비트 및 압축	오디오 샘플 레이트	파라미터 CRC	디지털 오디오 데이터	오디오 데이터 CRC
1 바이트	1 바이트	2 바이트	패킷 길이 - 14 바이트	2 바이트

도면14



도면15

패킷 길이	패킷 타입 = 56~63	bClient ID	스트림 파라미터	파라미터 CRC	스트림 데이터	스트림 데이터 CRC
2 바이트	2 바이트	2 바이트		2 바이트		2 바이트

도면16

패킷 길이	패킷 타입 = 64	hClient ID	필러 램 아이템 카운트	필러 램 오프셋	필러 램 아이템 크기	예약 1	파라미터 CRC	필러 램 데이터	데이터 CRC
2 바이트	2 바이트	2 바이트	2 바이트	4 바이트	1 바이트	1 바이트	2 바이트	(패킷 길이 - 16) 바이트	2 바이트

도면17

역방향 링크 캡슐화 패킷

패킷 길이	패킷 타입 = 65	hClient ID	역방향 링크 플래그	역방향 레이트 계수	전환 1 길이	전환 2 길이
2 바이트	2 바이트	2 바이트	1 바이트	1 바이트	1 바이트	1 바이트

파라미터 CRC	모두 제로 1	전환 1	역방향 데이터 패킷	전환 2	모두 제로 2
2 바이트	8 바이트	X 바이트	(패킷 길이 - X - Y - 26) 바이트	Y 바이트	8 바이트

도면18

클라이언트 용량 패킷

패킷 길이	패킷 타입 = 66	cClient ID	프로토콜 버전	최소 프로토콜 버전	데이터 레이트 용량	인터 페이스 타입 용량
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	1 바이트

Alt 디스플레이 수	예약 1	비트맵 폭	비트맵 높이	디스플레이 윈도우 폭	디스플레이 윈도우 높이	컬러 맵 크기
1 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	4 바이트

컬러 맵 RGB 폭	RGB 용량	단색 용량	예약 2	Y Cb Cr 용량	베이어 용량	알파-커서 이미지 평면
2 바이트	2 바이트	1 바이트	1 바이트	2 바이트	2 바이트	2 바이트

클라이언트 피쳐 용량	최대 비디오 프레임 레이트	최소 비디오 프레임 레이트	최소 서브 프레임 레이트	오디오 버퍼 깊이	오디오 채널 용량	오디오 샘플 레이트 용량
4 바이트	1 바이트	1 바이트	2 바이트	2 바이트	2 바이트	2 바이트

오디오 샘플 해상도	마이크 샘플 해상도	마이크 샘플 레이트 용량	키보드 데이터 포맷	포인팅 장치 데이터 포맷	컨텐츠 보호 타입	Mfr 명칭
1 바이트	1 바이트	2 바이트	1 바이트	1 바이트	2 바이트	2 바이트

제품 코드	예약 3	일련번호	주간 Mfr	연간 Mfr	CRC
2 바이트	2 바이트	4 바이트	1 바이트	1 바이트	2 바이트

도면19

패킷 길이	패킷 타입 = 67	bClient ID	키보드 데이터 포맷	키보드 데이터	CRC
2바이트	2바이트	2바이트	2바이트	(패킷_길이 - 8) 바이트	2바이트

도면20

패킷 길이	패킷 타입 = 68	bClient ID	포인팅 장치 포맷	포인팅 장치 데이터	CRC
2 바이트	2 바이트	2 바이트	2 바이트	(패킷_길이 - 8) 바이트	2 바이트

도면21

패킷 길이	패킷 타입 - 69	CRC	모두 제로
2 바이트	2 바이트	2 바이트	16 바이트

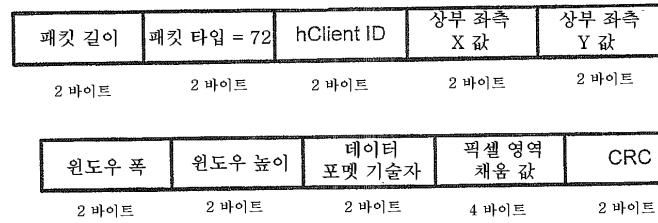
도면22

패킷 길이	패킷 타입 = 70	cClient ID	역방향 링크 요청	CRC 에러 카운트	용량 변경	그래픽 비지블 레그	CRC
2 바이트	2 바이트	2 바이트	2 바이트	1 바이트	1 바이트	2 바이트	2 바이트

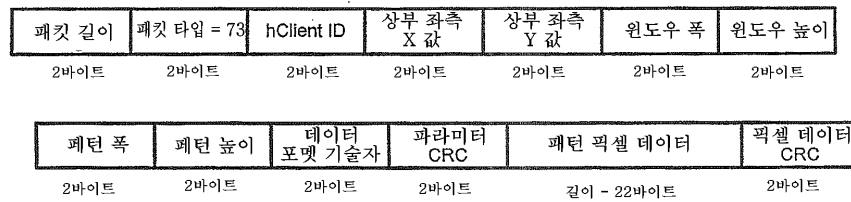
도면23

윈도우 폭	윈도우 높이	윈도우 X 이동	윈도우 Y 이동	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

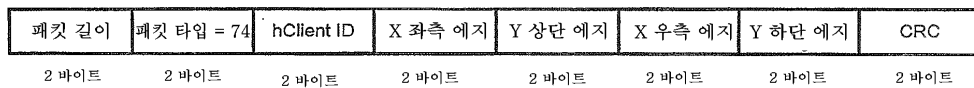
도면24



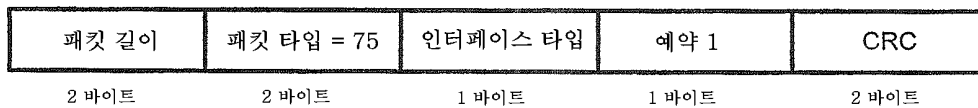
도면25



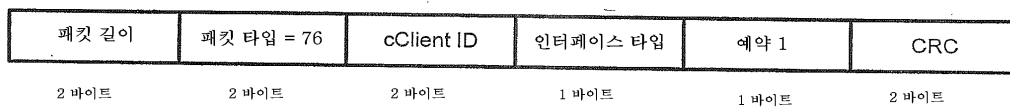
도면26



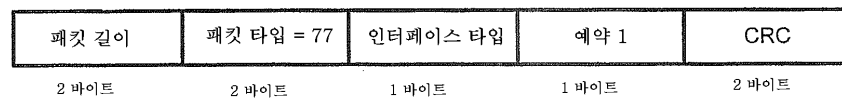
도면27



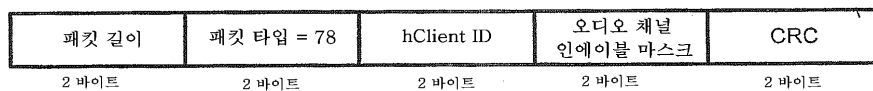
도면28



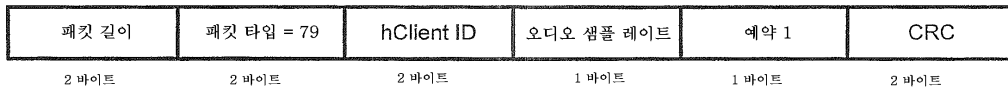
도면29



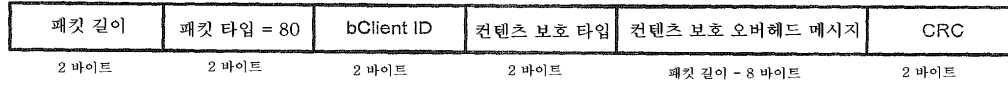
도면30



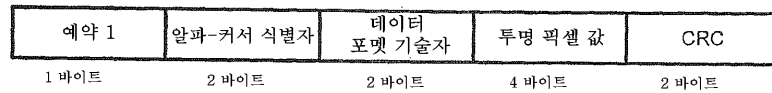
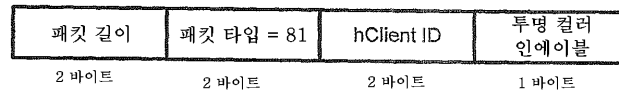
도면31



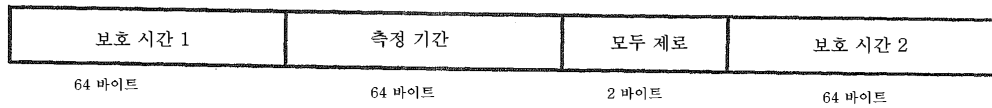
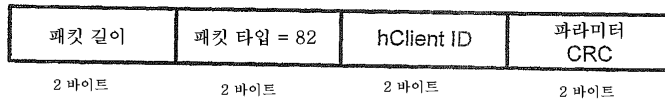
도면32



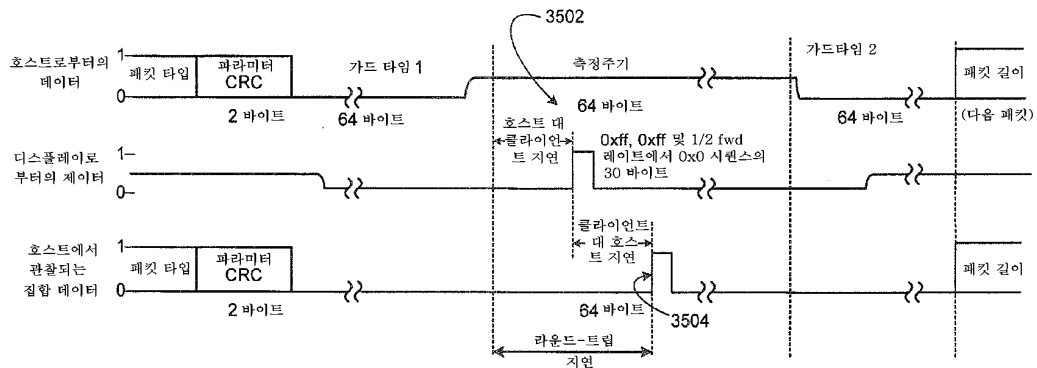
도면33



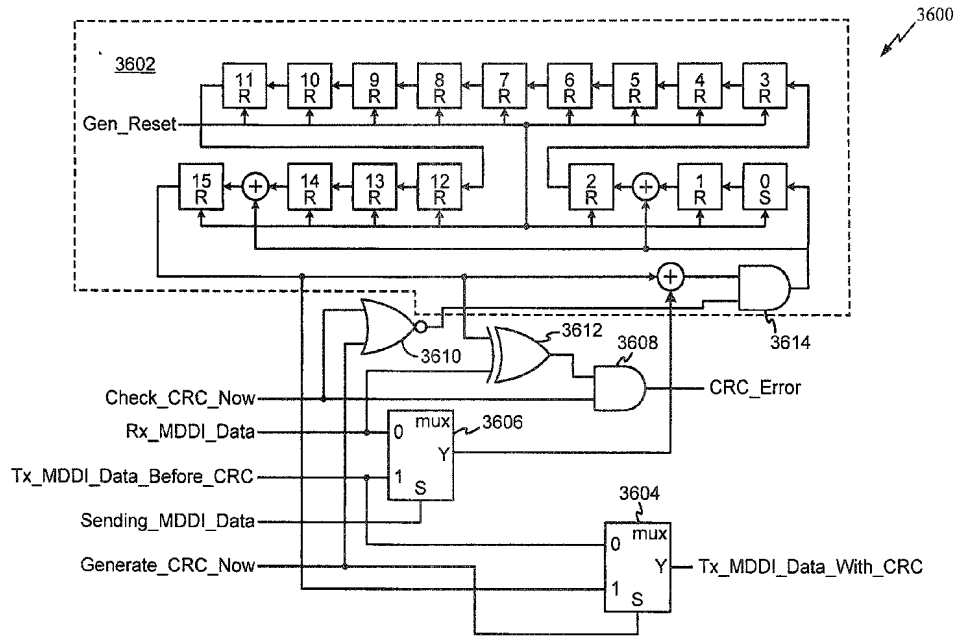
도면34



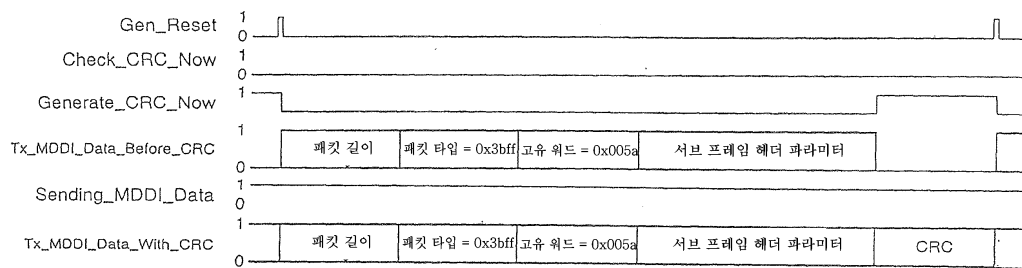
도면35



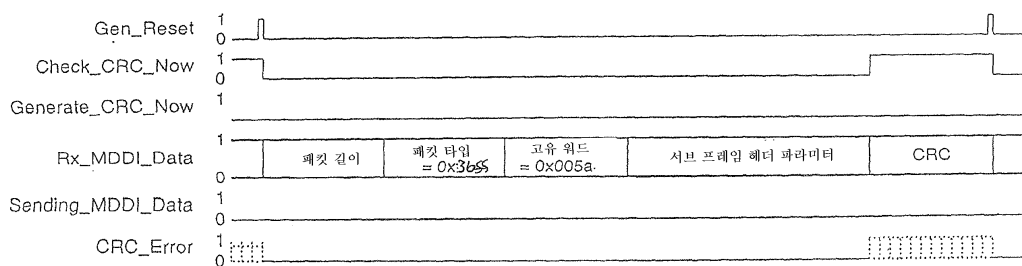
도면36



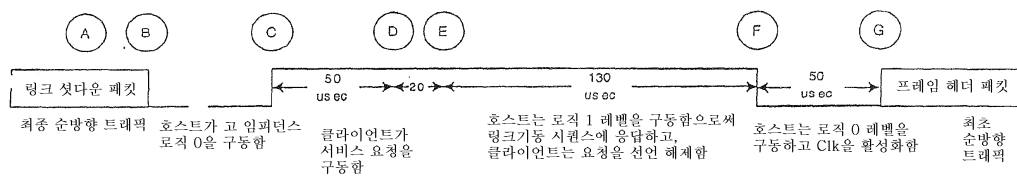
도면37a



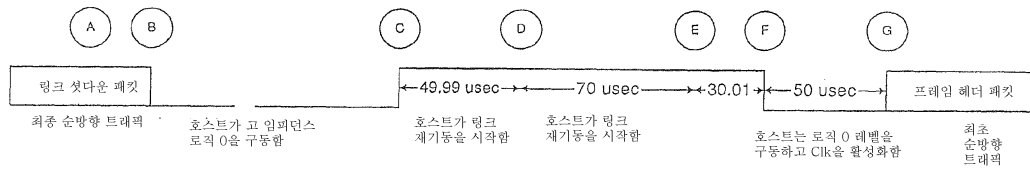
도면37b



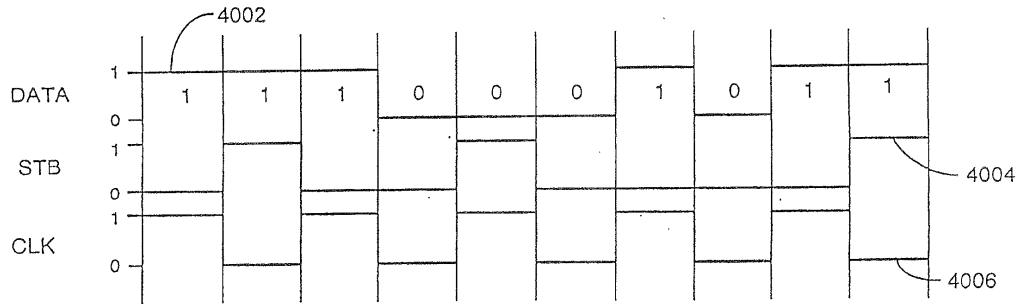
도면38



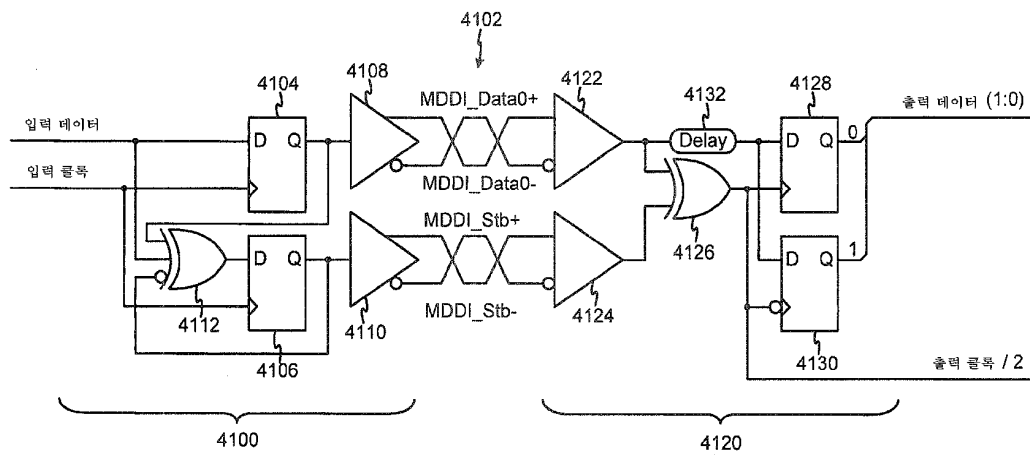
도면39



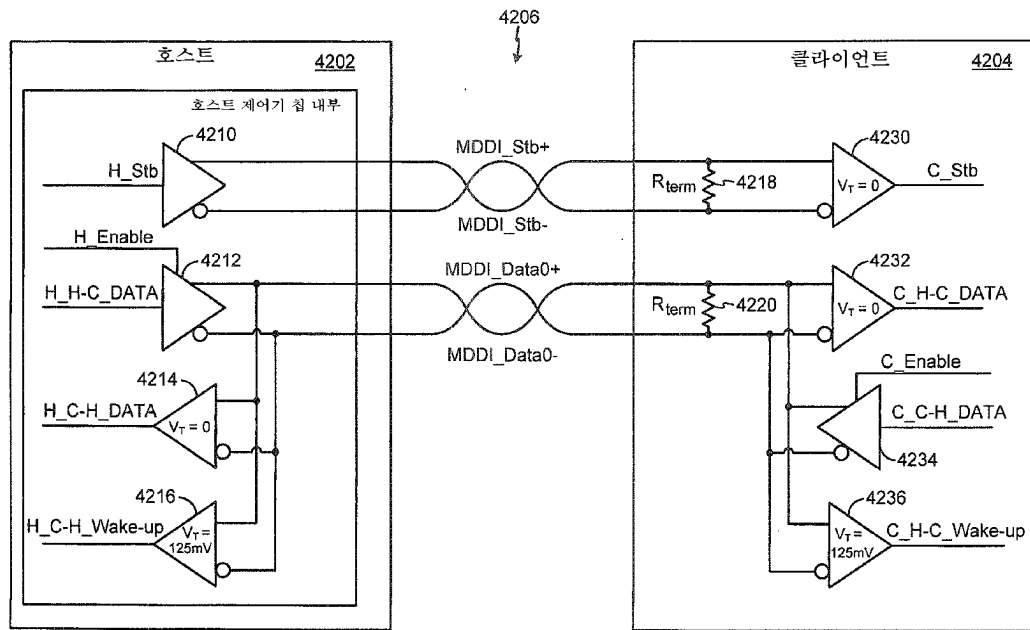
도면40



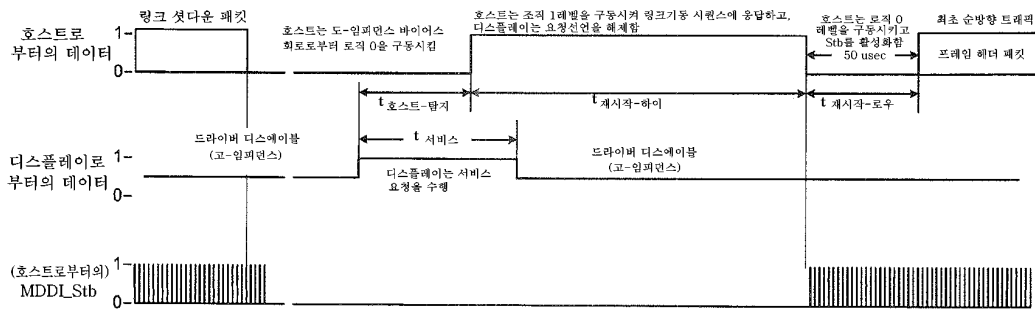
도면41



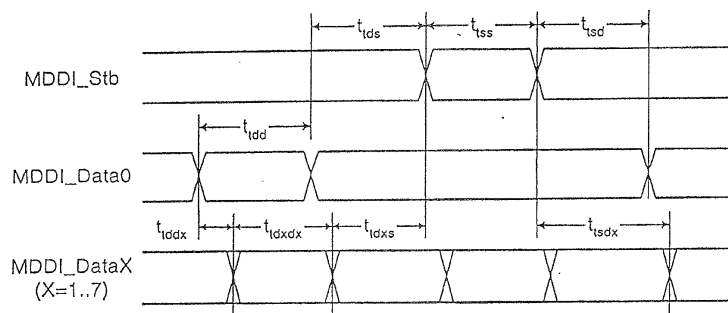
도면42



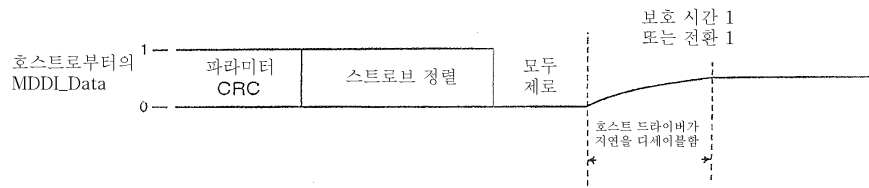
도면43



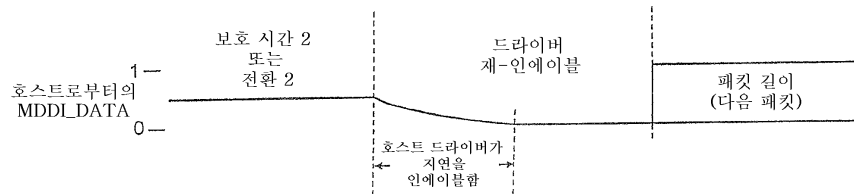
도면44



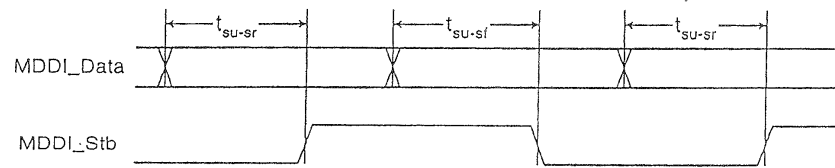
도면45



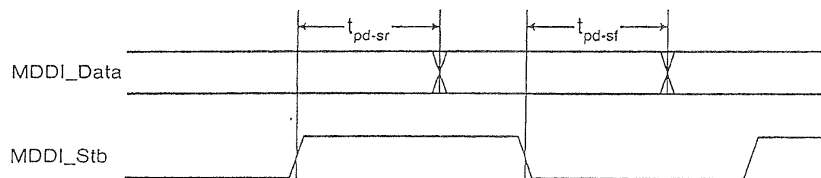
도면46



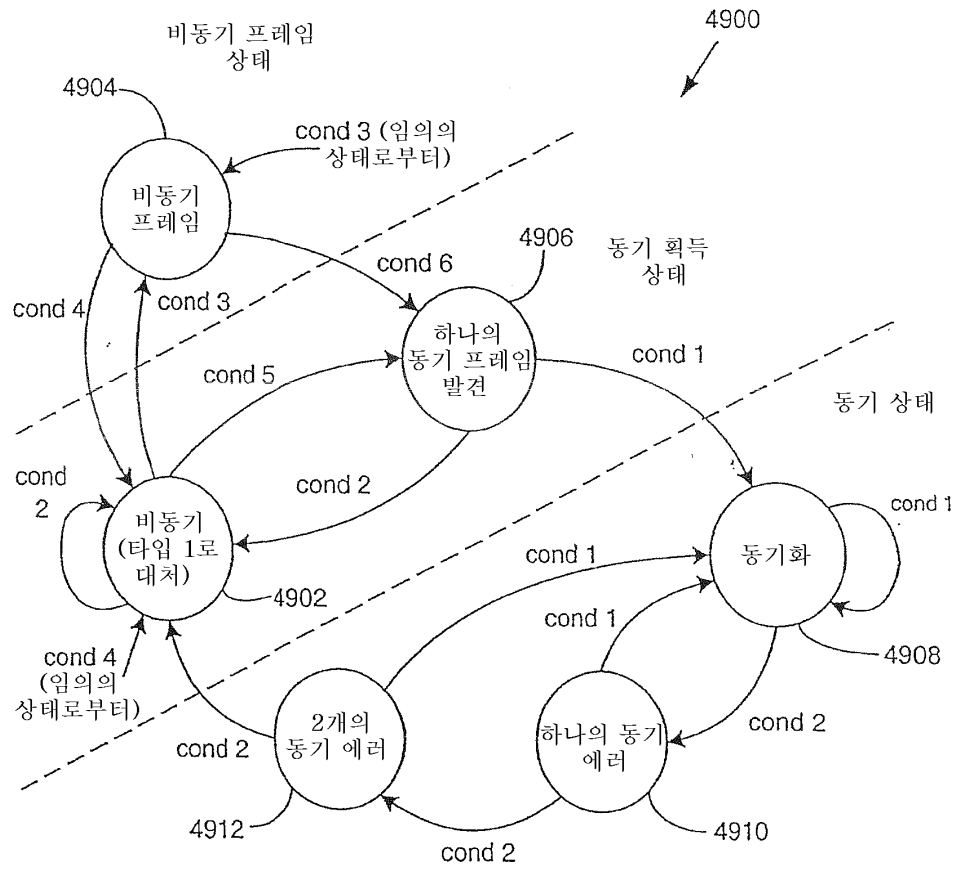
도면47



도면48



도면49



cond 1 = 서브 프레임 헤더 패킷 & 프레임 경계에서 양호한 CRC, 프레임 길이 > 0

cond 2 = 비동기 패턴 또는 프레임 경계에서 잘못된 CRC

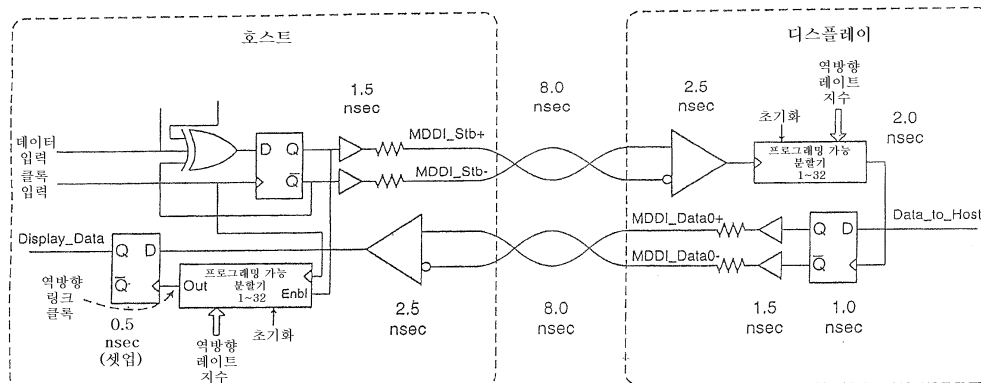
cond 3 = 동기 패턴 발견, 프레임 길이 = 0

cond 4 = 링크 쉷다운 패킷 수신

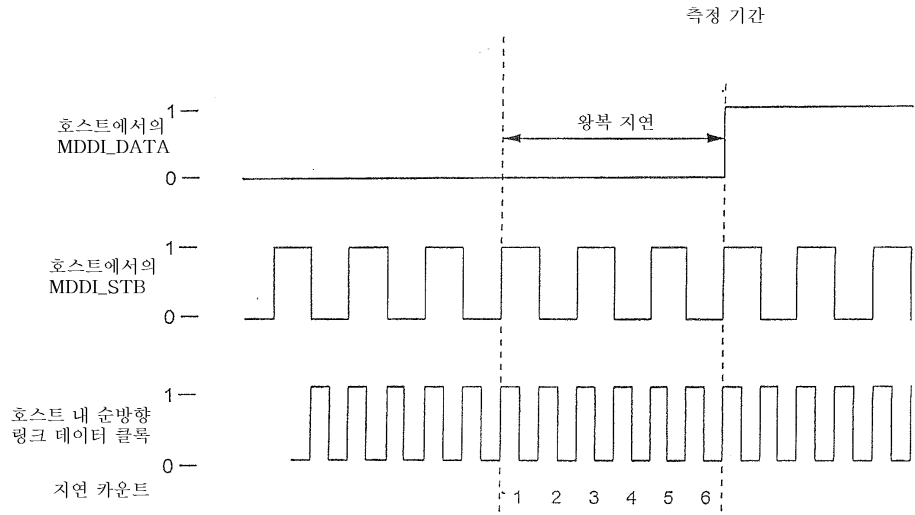
cond 5 = 동기 패턴 발견, 프레임 길이 > 0

cond 6 = 프레임 헤더 패킷 & 양호한 CRC, 프레임 길이 > 0

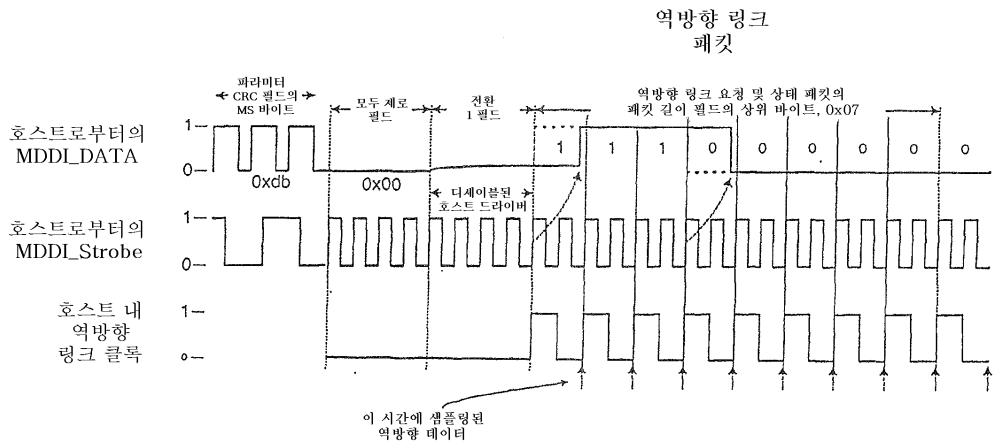
도면50



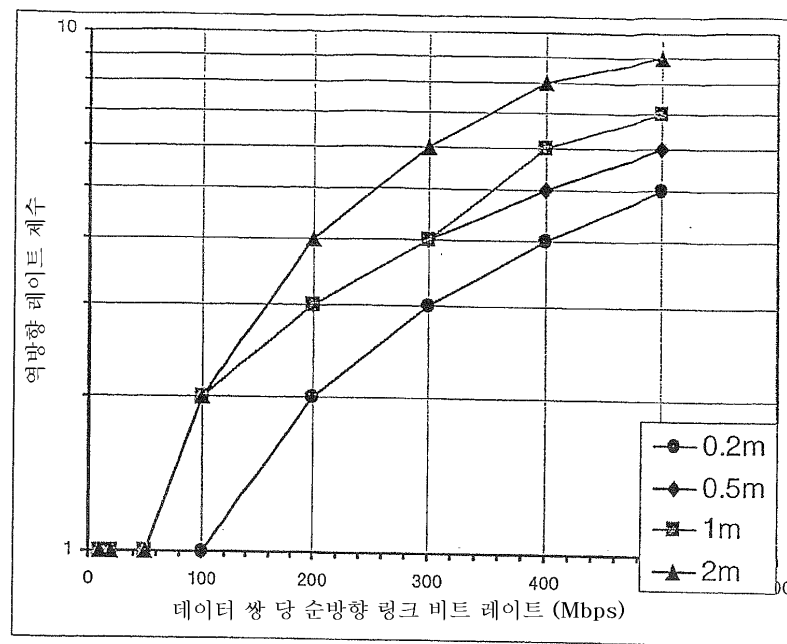
도면51



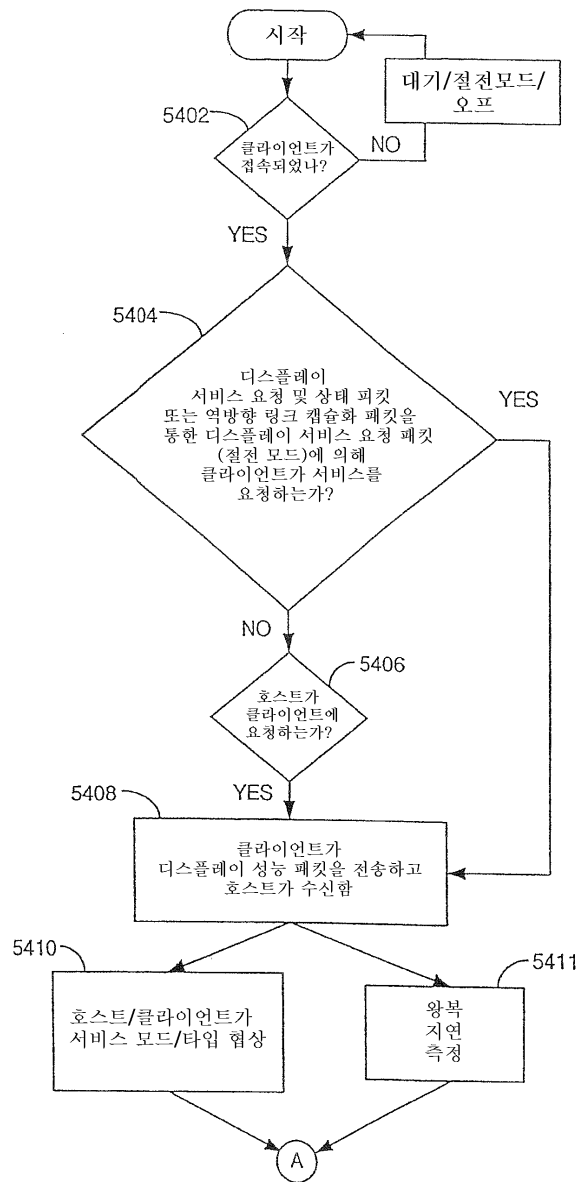
도면52



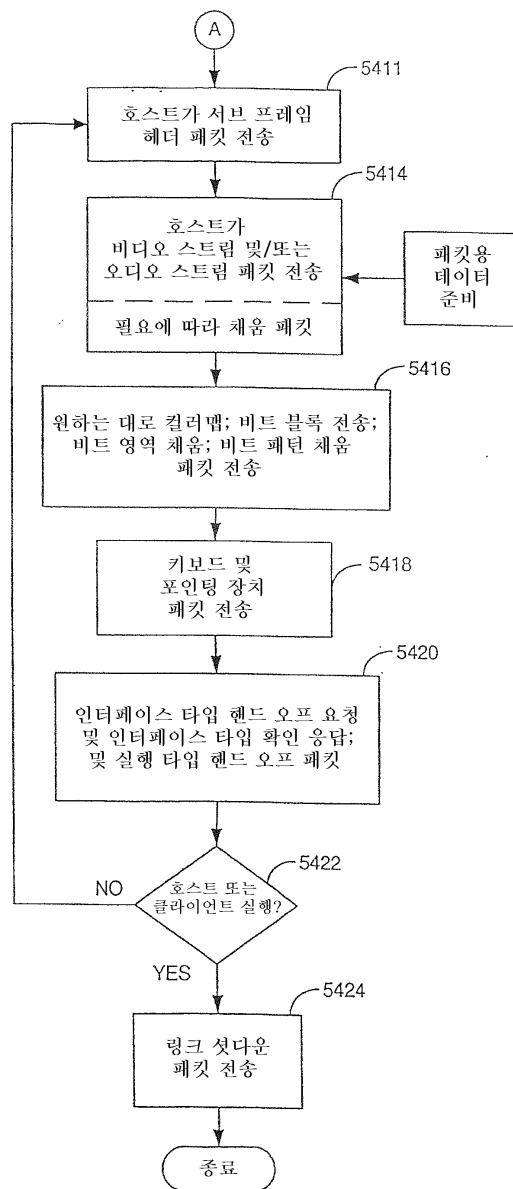
도면53



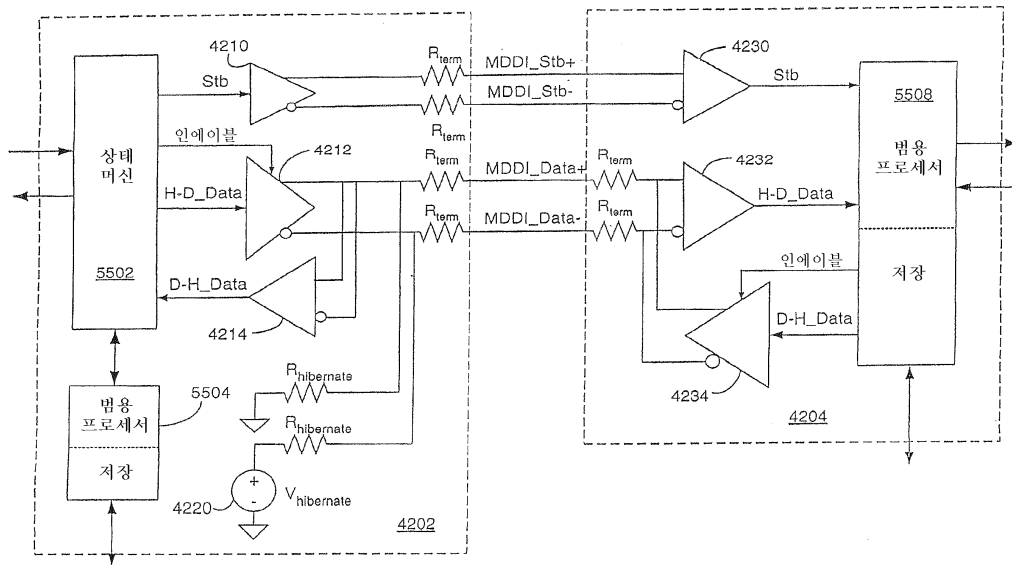
도면54a



도면54b



도면55

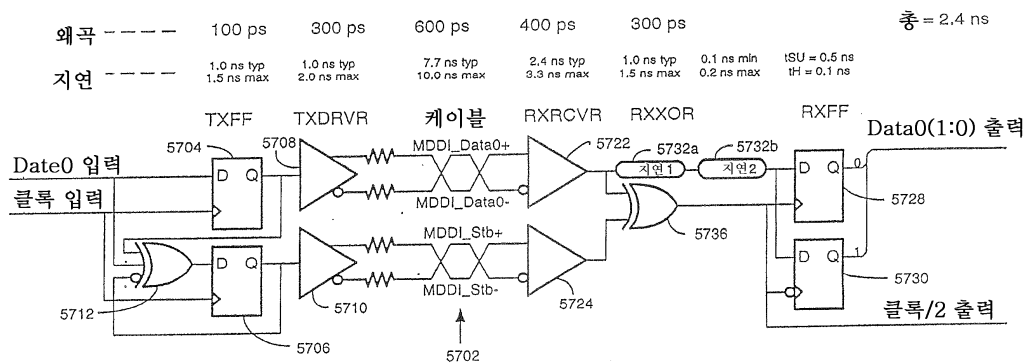


도면56

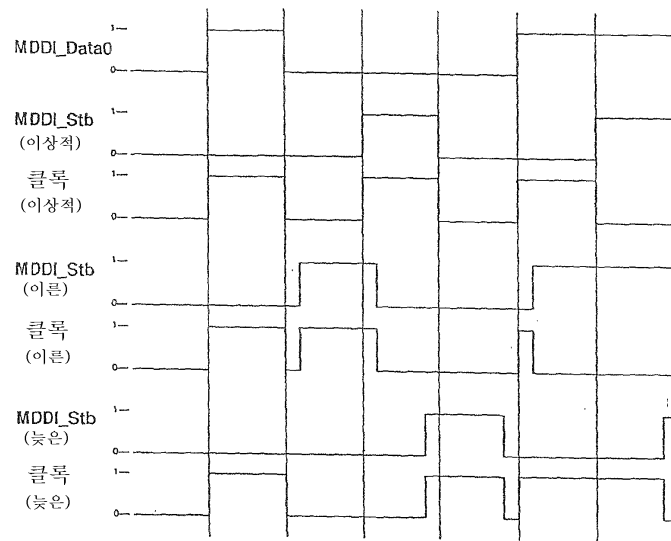
순방향 링크 스루 캘리브레이션 패킷

패킷 길이	패킷 타입 = 83	hClient ID	파라미터 CRC	모두 제로 1	캘리브레이션 시퀀스	모두 제로 2
2 바이트	2 바이트	2 바이트	2 바이트	8 바이트	패킷 길이 - 22 바이트	8 바이트

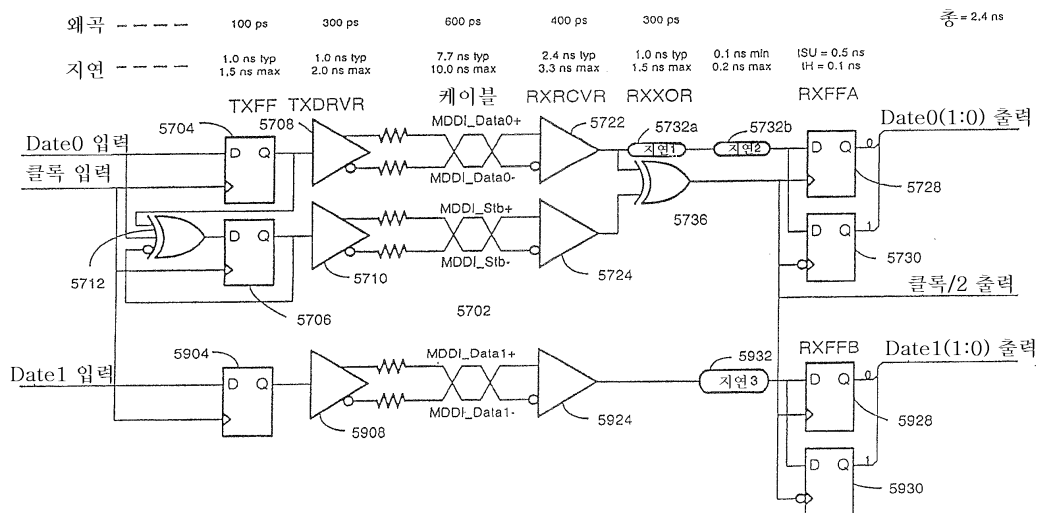
도면57



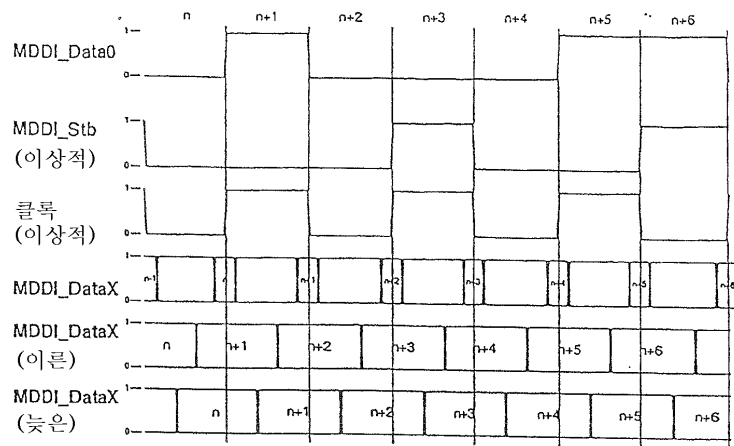
도면58



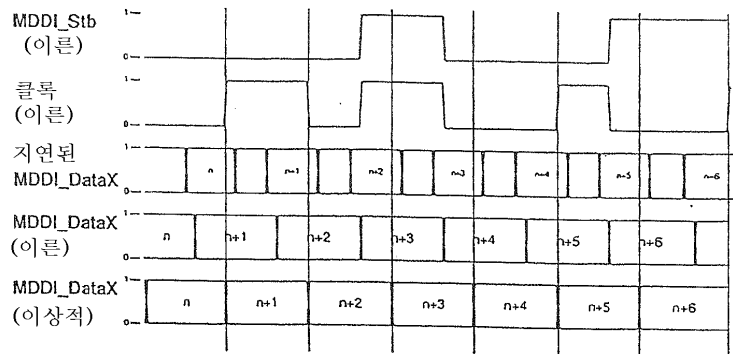
도면59



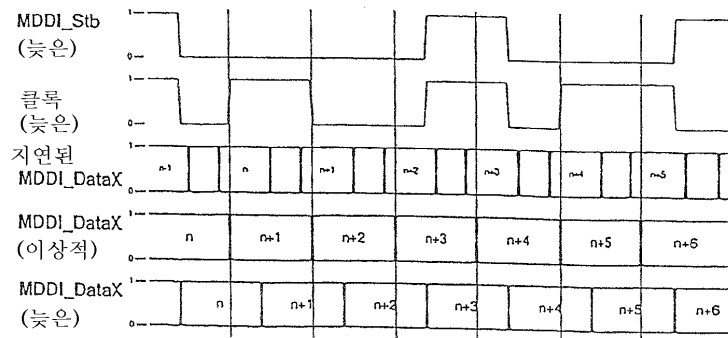
도면60a



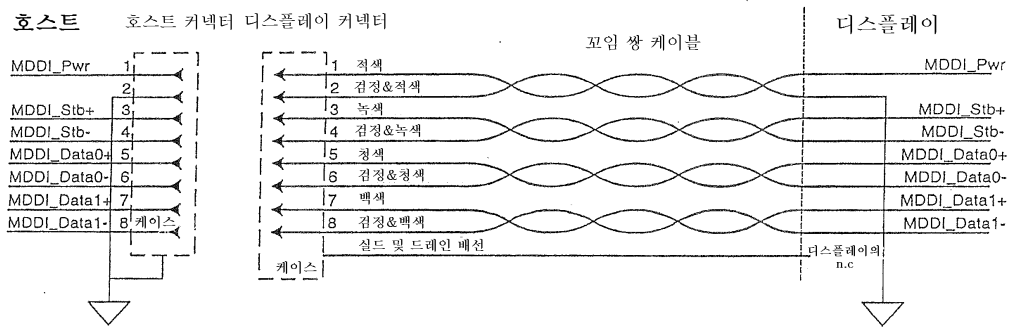
도면60b



도면60c

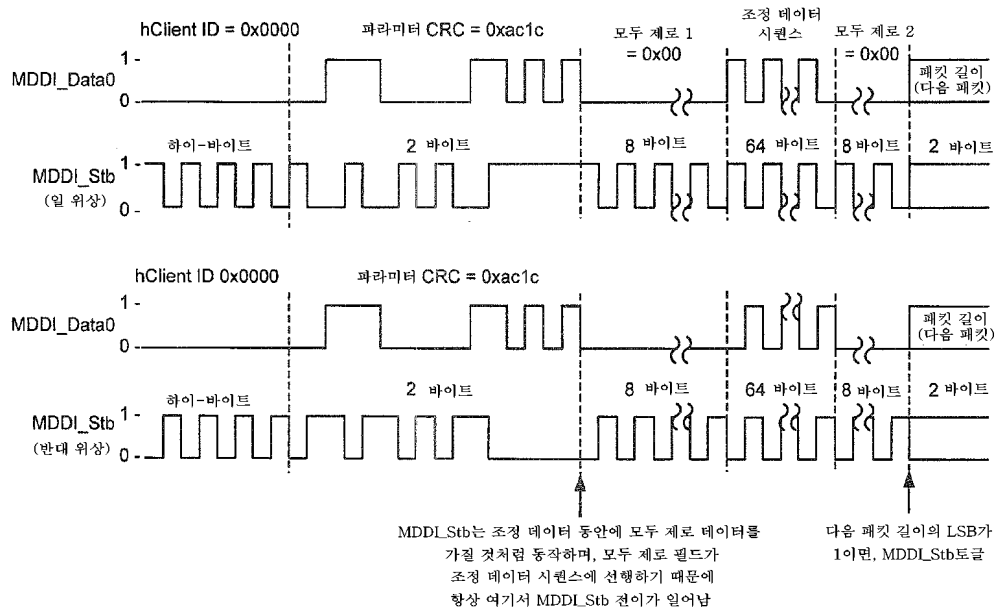


도면61

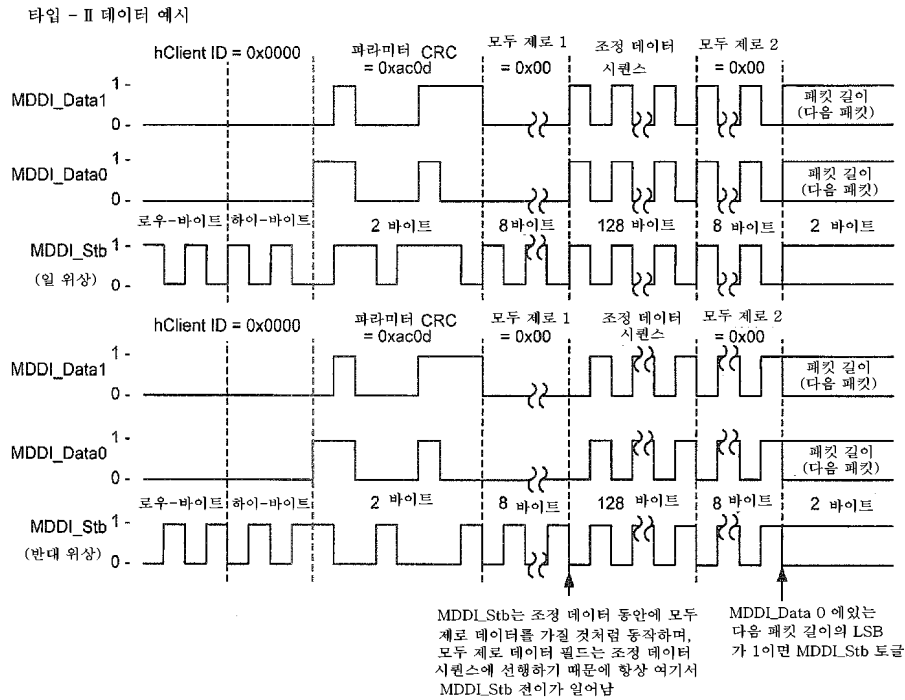


도면62a

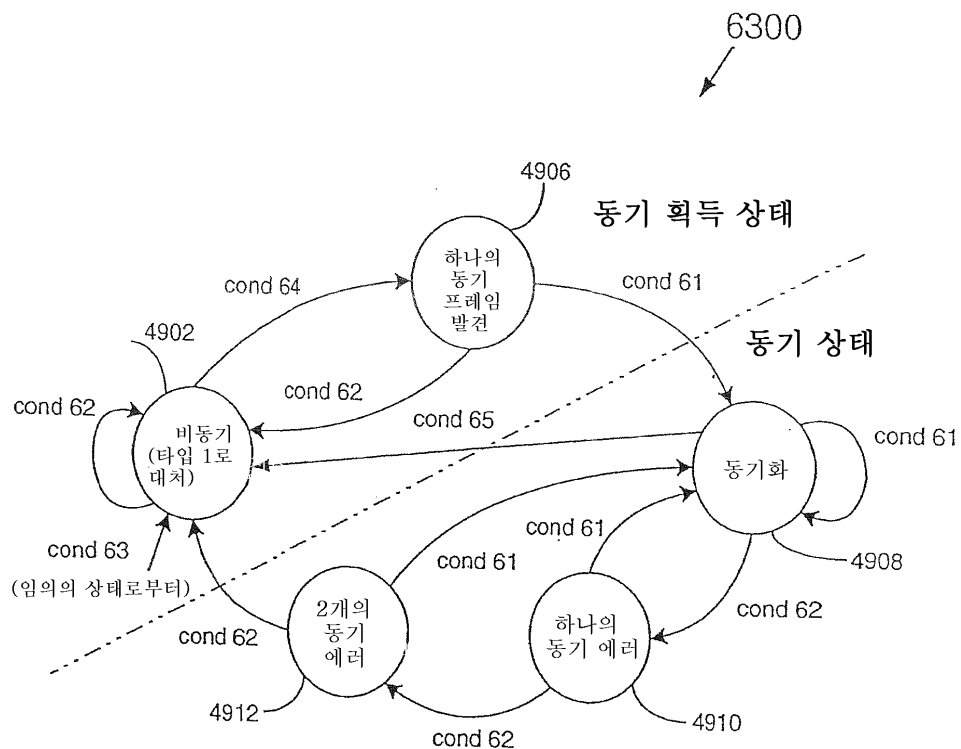
타입 - I 데이터 예시



도면62b

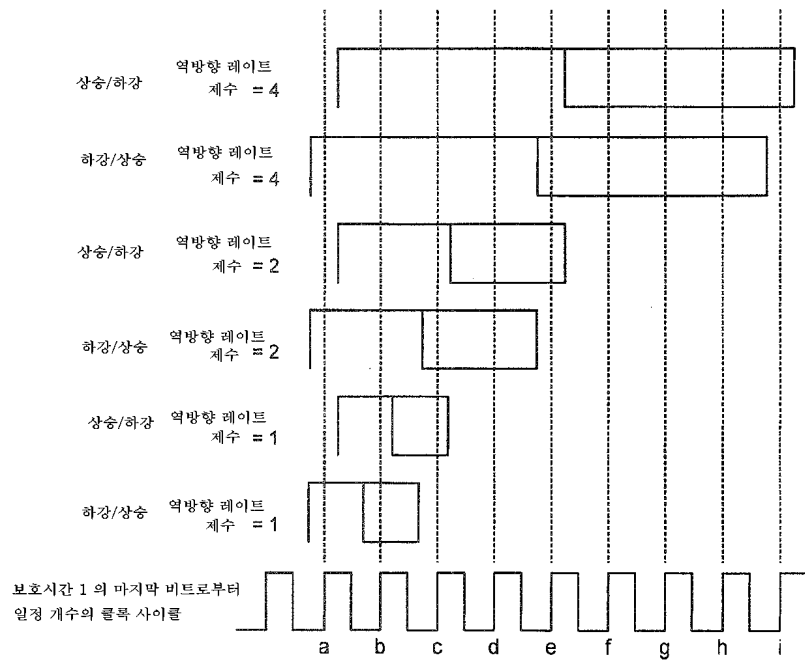


도면63

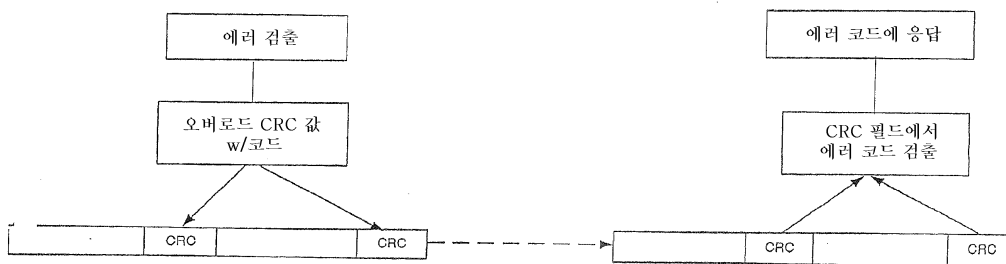


- cond 61 = 서브 프레임 헤더 패킷 & 프레임 경계에서 양호한 CRC, 프레임 길이 > 0
- cond 62 = 비동기 패턴 또는 프레임 경계에서 잘못된 CRC
- cond 63 = 링크 췌다운 패킷 수신
- cond 64 = 동기 패턴 발견
- cond 65 = 고유 워드 부정확

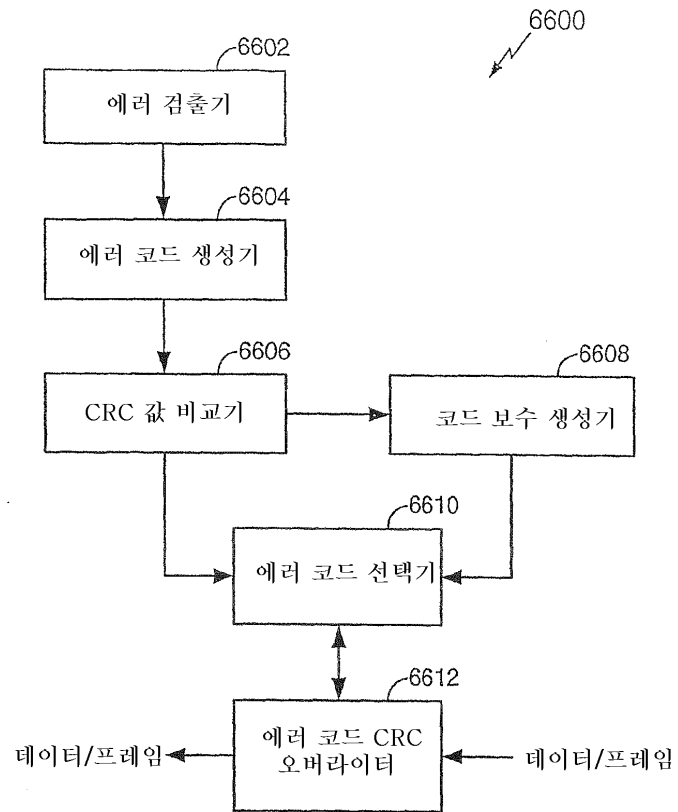
도면64



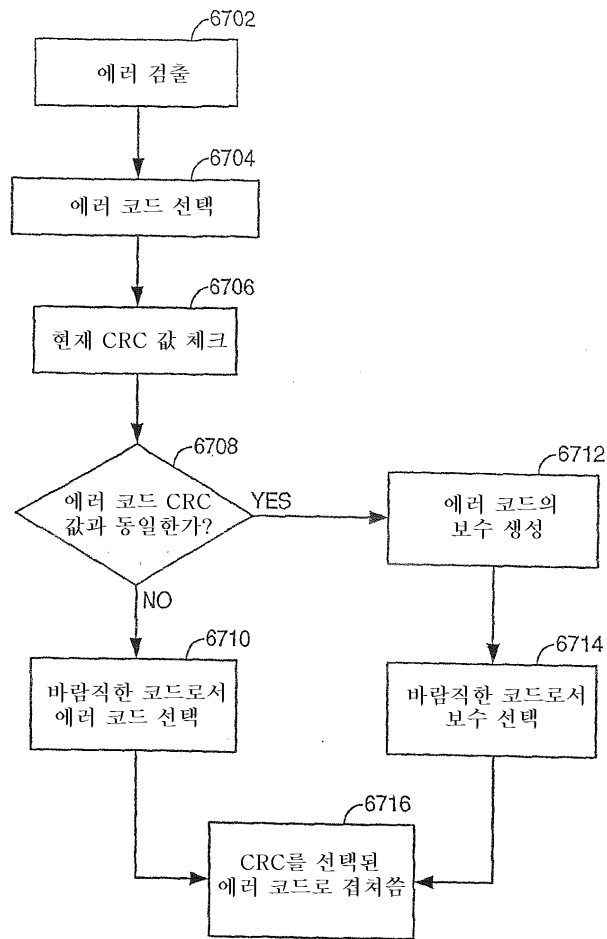
도면65



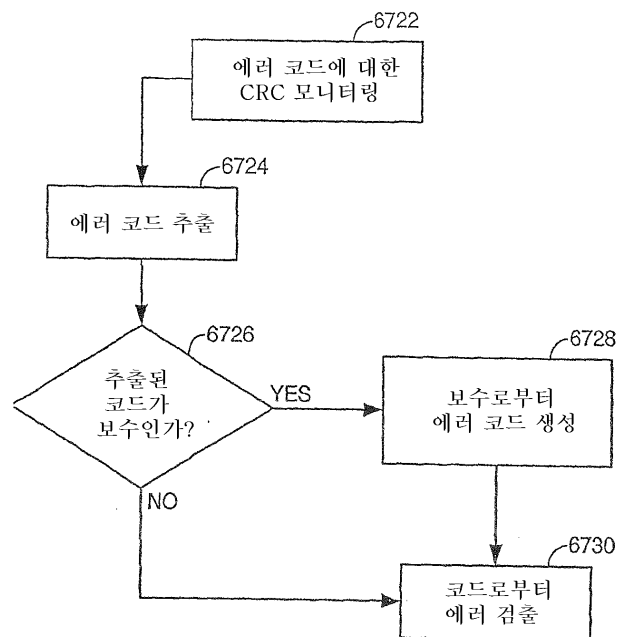
도면66



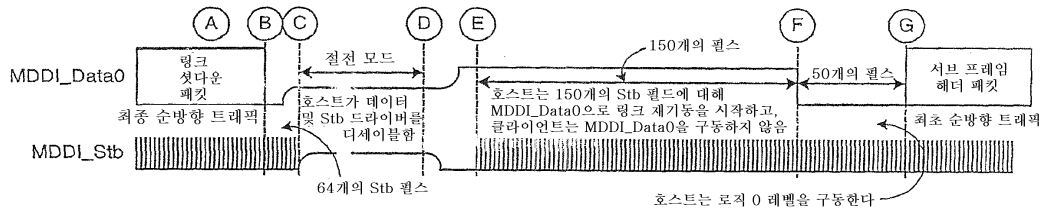
도면67a



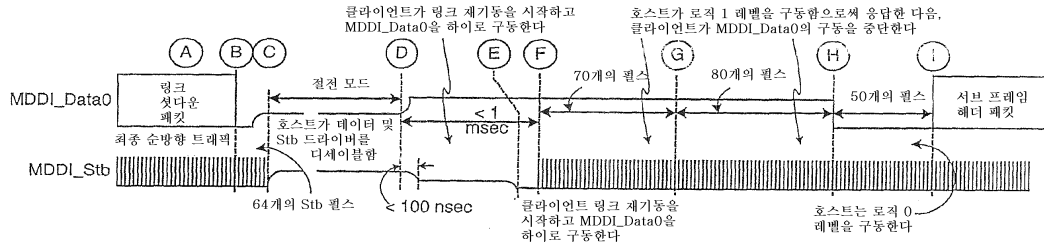
도면67b



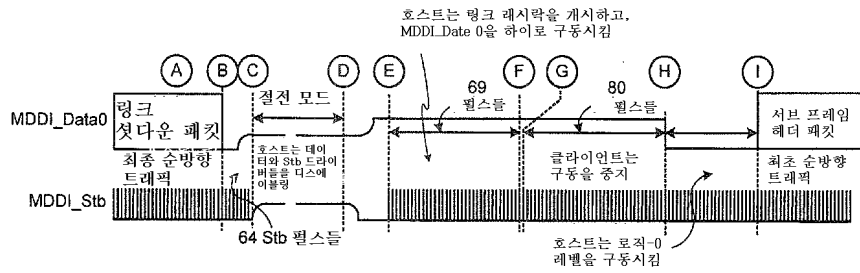
도면68a



도면68b



도면68c



도면69

요청 VCP 피쳐 패킷

패킷 길이	패킷 타입 = 128	hClient ID	MCCS VCP 코드	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

도면70

VCP 피쳐 응답 패킷

패킷 길이	패킷 타입 = 129	cClient ID	MCCS 버전	응답 시퀀스 번호	리스트 내 피쳐 수	VCP 피쳐 응답 리스트	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	(패킷 길이 - 12) 바이트	2 바이트

도면71

VCP 피쳐 응답 리스트 아이템

MCCS VCP 코드	결과 코드	최대값	현재값
2 바이트	2 바이트	4 바이트	4 바이트

도면72

세트 VCP 피치 패킷

패킷 길이	패킷 타입 = 130	hClient ID	MCCS VCP 코드	리스트 내 값들의 수	제어값 리스트	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	(패킷 길이 - 10) 바이트	2 바이트

2

도면73

요청 유효 파라미터 패킷

패킷 길이	패킷 타입 = 131	hClient ID	MCCS VCP 코드	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

도면74

유효 파라미터 응답 패킷

패킷 길이	패킷 타입 = 132	cClient ID	MCCS VCP 코드	응답 코드	응답 서퀀스 번호	리스트 내 값들의 수	VCP 파라미터 응답 리스트	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	(패킷 길이 - 14) 바이트	2 바이트

도면75

알파 커서 이미지 용량 패킷

패킷 길이	패킷 타입 = 133	cClient ID	알파 커서 식별자	알파 커서 비트맵 폭	알파 커서 비트맵 높이
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

RGB 용량	단색 용량	예약 1	Y Cr Cb 용량	투명 맵 해상도	용량 비트	CRC
2 바이트	1 바이트	1 바이트	2 바이트	1 바이트	1 바이트	2 바이트

도면76

알파 커서 투명 맵 패킷

패킷 길이	패킷 타입 = 134	hClient ID	알파 커서 식별자	투명 맵 X 시작	투명 맵 Y 시작
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

투명 맵 해상도	예약 1	파라미터 CRC	투명 맵 데이터	투명 맵 데이터 CRC
1 바이트	1 바이트	2 바이트	(패킷 길이 - 16) 바이트	2 바이트

도면77

알파-커서 이미지 오프셋 패킷

패킷 길이	패킷 타입 = 135	hClient ID	알파-커서 X 오프셋	알파-커서 Y 오프셋	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

도면78

알파 커서 비디오 스트림 패킷

패킷 길이	패킷 타입 = 17	bClient ID	비디오 데이터 포맷 기술자	픽셀 데이터 속성	X 좌측 에지	Y 상단 에지	X 우측 에지	Y 하단 에지
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

X 시작	Y 시작	픽셀 카운트	파라미터 CRC	픽셀 데이터	픽셀 데이터 CRC
2 바이트	2 바이트	2 바이트	2 바이트	(패킷 길이 - 26)바이트	2 바이트

도면79

스케일 비디오 스트림 용량 패킷

패킷 길이	패킷 타입 = 143	cClient ID	스트림 최대 개수	소스 최대 X 크기	소스 최대 Y 크기
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

RGB 용량	단색 용량	예약 1	Y Cr Cb 용량	용량 비트	예약 2	CRC
2 바이트	1 바이트	1 바이트	2 바이트	1 바이트	1 바이트	2 바이트

도면80

스케일 비디오 스트림 셋업 패킷

패킷 길이	패킷 타입 = 136	hClient ID	스트림 ID	비디오 데이터 포맷 기술자	픽셀 데이터 속성
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

X 좌측 에지	Y 상단 에지	X 우측 에지	Y 하단 에지	X 이미지 크기	Y 이미지 크기	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

도면81

스케일 비디오 스트림 확인 응답 패킷

패킷 길이	패킷 타입 = 137	cClient ID	스트림 ID	확인 코드	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

도면82

스케일 비디오 스트림 패킷

패킷 길이	패킷 타입 = 18	hClient ID	스트림 ID	파라미터 CRC	픽셀 카운트	픽셀 데이터	픽셀 데이터 CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	2 바이트	패킷 길이-12 바이트	2 바이트

도면83

요청 지정 상태 패킷

패킷 길이	패킷 타입 = 138	hClient ID	상태 패킷 ID	CRC
2 바이트	2 바이트	2 바이트	2 바이트	2 바이트

도면84

유효 상태 응답 리스트 패킷

패킷 길이	패킷 타입 = 139	cClient ID	리스트 내 값들 개수	유효 파라미터 응답 리스트	CRC
2 바이트	2 바이트	2 바이트	2 바이트	패킷 길이 - 8 바이트	2 바이트

도면85a

패킷 처리 지연 파라미터 패킷

패킷 길이	패킷 타입 = 140	cClient ID	리스트 아이템 수	지연 파라미터 리스트	CRC
2 바이트	2 바이트	2 바이트	2 바이트	패킷 길이 - 8 바이트	2 바이트

도면85b

지연 파라미터 리스트 아이템

지연 패킷 타입	픽셀 지연	수평 픽셀 지연	수직 픽셀 지연	고정 지연
2 바이트	1 바이트	1 바이트	1 바이트	1 바이트

도면86

개인용 디스플레이 용량 패킷

패킷 길이	패킷 타입 = 141	cClient ID	서브 픽셀 레이아웃	픽셀 형태	수평 뷰 필드	수직 뷰 필드	가시 축 교차
2 바이트	2 바이트	2 바이트	1 바이트	1 바이트	1 바이트	1 바이트	1 바이트

좌측/우측 이미지 오버랩	투시	최대 휘도	광 용량	최소 IPD	최대 IPD	필드 곡률 리스트의 포인트 (25개의 2 바이트 값)	CRC
1 바이트	1 바이트	1 바이트	2 바이트	1 바이트	1 바이트	50 바이트	2 바이트

도면87a

클라이언트 에러 보고 패킷

패킷 길이	패킷 타입 = 142	cClient ID	리스트 아이템 수	에러 코드 리스트	CRC
2 바이트	2 바이트	2 바이트	2 바이트	패킷 길이 - 8 바이트	2 바이트

도면87b

에러 보고 리스트 아이템

디스플레이 에러 코드	에러 서브 코드
2 바이트	2 바이트

도면88

클라이언트 식별 패킷

패킷 길이	패킷 타입 = 144	cClient ID	Mfr 주간	Mfr 년도	Mfr 명칭 길이	제품명 길이	일련번호 길이
2 바이트	2 바이트	2 바이트	1 바이트	1 바이트	2 바이트	2 바이트	2 바이트

제조사명 스트림	제품명 스트림	일련번호 스트림	CRC
Mfr 명칭 길이 바이트	Mfr 명칭 길이 바이트	Mfr 명칭 길이 바이트	2 바이트

도면89

대안 디스플레이 용량 패킷

패킷 길이	패킷 타입 = 145	cClient ID	대안 디스플레이 번호	예약 1	비트맵 폭	비트맵 높이	디스플레이 원도우 폭	디스플레이 원도우 높이
2 바이트	2 바이트	2 바이트	1 바이트	1 바이트	2 바이트	2 바이트	2 바이트	2 바이트

컬러맵 RGB 폭	RGB 용량	단색 용량	예약 2	Y Cb Cr 용량	디스플레이 피쳐 용량	예약 3	CRC
2 바이트	2 바이트	1 바이트	1 바이트	2 바이트	1 바이트	1 바이트	2 바이트

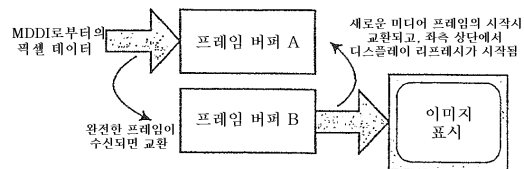
도면90

레지스터 액세스 패킷

패킷 길이	패킷 타입 = 146	bClient ID	읽기/쓰기 플래그	레지스터 어드레스	파라미터 CRC	레지스터 데이터 리스트	레지스터 데이터 CRC
2 바이트	2 바이트	2 바이트	2 바이트	4 바이트	2 바이트	패킷 길이 - 14 바이트	2 바이트

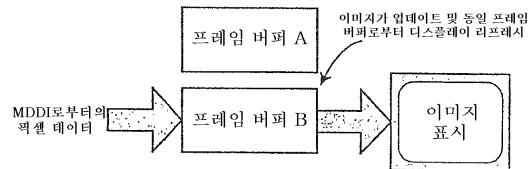
도면91a

디스플레이 업데이트 비트 "01"

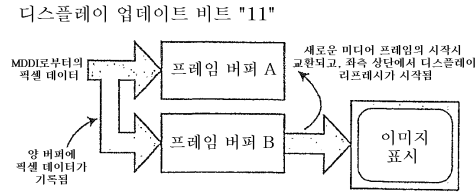


도면91b

디스플레이 업데이트 비트 "00"



도면91c



도면92

이미지 전달 디스플레이 리프레쉬가 이미지 전달보다 빠른 두개의 버퍼들

프레임 1, 버퍼 A 20% 40% 60% 80%	프레임 2, 버퍼 B 20% 40% 60% 80%	프레임 3, 버퍼 A 20% 40% 60% 80%	프레임 4, 버퍼 B 20% 40% 60% 80%	프레임 5, 버퍼 A 20% 40% 60% 80%	프레임 6, 버퍼 B 20% 40% 60% 80%
--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------

디스플레이 리프레쉬

프레임 0, 버퍼 B 20% 40% 60% 80%	프레임 0, 버퍼 B 20% 40% 60% 80%	프레임 1, 버퍼 A 20% 40% 60% 80%	프레임 2, 버퍼 B 20% 40% 60% 80%	프레임 2, 버퍼 B 20% 40% 60% 80%	프레임 3, 버퍼 A 20% 40% 60% 80%	프레임 4, 버퍼 B 20% 40% 60% 80%	프레임 4, 버퍼 B 20% 40% 60% 80%	프레임 5, 버퍼 A 20% 40% 60% 80%
--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------

이미지 전달 디스플레이 리프레쉬가 이미지 전달보다 훨씬 빠른 두개의 버퍼들

프레임 1, 버퍼 A 20% 40% 60% 80%	프레임 2, 버퍼 B 20% 40% 60% 80%	프레임 3, 버퍼 A 20% 40% 60% 80%	프레임 4, 버퍼 B 20% 40% 60% 80%	프레임 5, 버퍼 A 20% 40% 60% 80%	프레임 6, 버퍼 B 20% 40% 60% 80%
--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------

디스플레이 리프레쉬

fn 0, b=B	fn 0, b=B	fn 0, b=B	fn 1, b=A	fn 1, b=A	fn 2, b=B	fn 2, b=B	fn 2, b=B	fn 3, b=A	fn 3, b=A	fn 4, b=B	fn 4, b=B	fn 4, b=B	fn 5, b=A	fn 5, b=A
20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80
% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %

도면93

이미지 전달 디스플레이 리프레쉬가 이미지 전달보다 느린 두개의 버퍼들

프레임 1, 버퍼 A 20% 40% 60% 80%	프레임 2, 버퍼 B 20% 40% 60% 80%	프레임 3, 버퍼 A 20% 40% 60% 80%	프레임 4, 버퍼 B 20% 40% 60% 80%	프레임 5, 버퍼 A 20% 40% 60% 80%	프레임 6, 버퍼 B 20% 40% 60% 80%
--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------

디스플레이 리프레쉬

프레임 0, 버퍼 B 20% 40% 60% 80%	프레임 1&3, 버퍼 A 20% 40% 60% 80%	프레임 3, 버퍼 A 20% 40% 60% 80%	프레임 4&6, 버퍼 B 20% 40% 60% 80%
--------------------------------	----------------------------------	--------------------------------	----------------------------------

도면94

이미지 전달 디스플레이 리프레쉬가 이미지 전달보다 훨씬 빠른 두개의 버퍼들, 작은 비디오 창

프레임 1, 버퍼 A 20% 40% 60% 80%	프레임 2, 버퍼 B 20% 40% 60% 80%	프레임 3, 버퍼 A 20% 40% 60% 80%	프레임 4, 버퍼 B 20% 40% 60% 80%	프레임 5, 버퍼 A 20% 40% 60% 80%	프레임 6, 버퍼 B 20% 40% 60% 80%
--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------

디스플레이 리프레쉬

fn 0, b=B	fn 0, b=B	fn 0, b=B	fn 1, b=A	fn 1, b=A	fn 2, b=B	fn 2, b=B	fn 2, b=B	fn 3, b=A	fn 3, b=A	fn 4, b=B	fn 4, b=B	fn 4, b=B	fn 5, b=A	fn 5, b=A
20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80
% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %

도면95

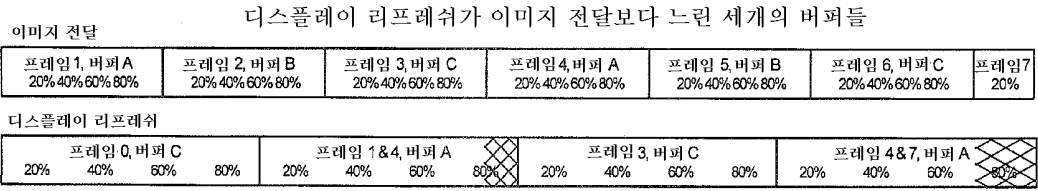
이미지 전달 디스플레이 리프레쉬가 이미지 전달보다 느린 세개의 버퍼들

프레임 1, 버퍼 A 20% 40% 60% 80%	프레임 2, 버퍼 B 20% 40% 60% 80%	프레임 3, 버퍼 C 20% 40% 60% 80%	프레임 4, 버퍼 A 20% 40% 60% 80%	프레임 5, 버퍼 B 20% 40% 60% 80%	프레임 6, 버퍼 C 20% 40% 60% 80%
--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------	--------------------------------

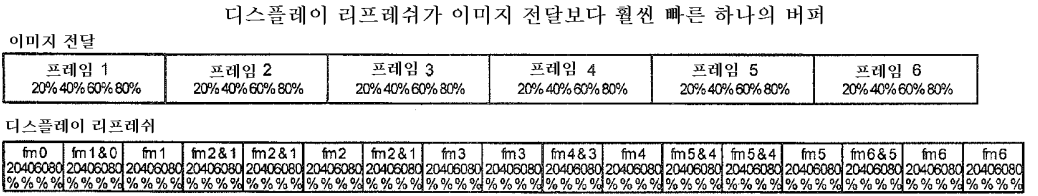
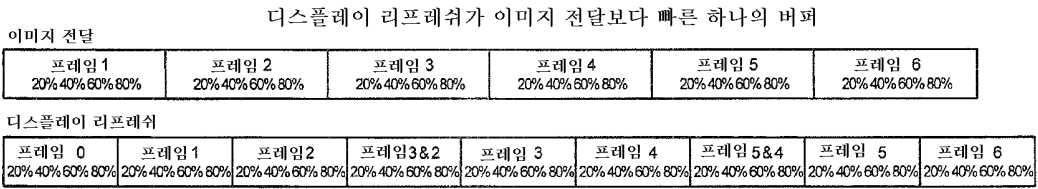
디스플레이 리프레쉬

fn 0, b=C	fn 0, b=C	fn 0, b=C	fn 1, b=A	fn 1, b=A	fn 2, b=B	fn 2, b=B	fn 2, b=B	fn 3, b=C	fn 3, b=C	fn 4, b=A	fn 4, b=A	fn 4, b=A	fn 5, b=B	fn 5, b=B
20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80	20 40 60 80
% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %	% % % %

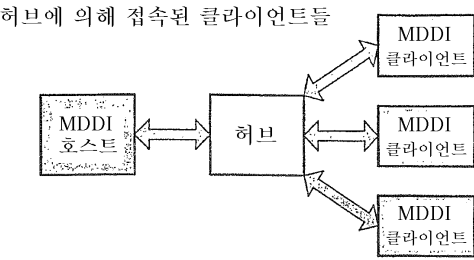
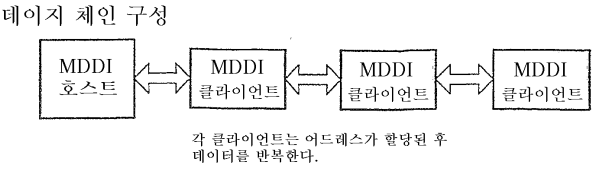
도면96



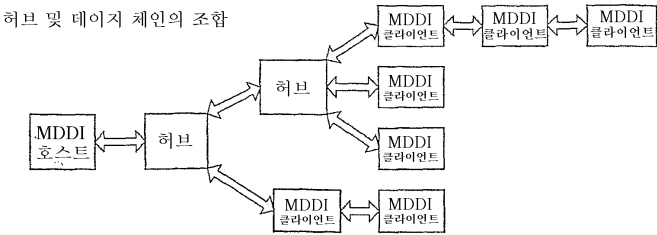
도면97



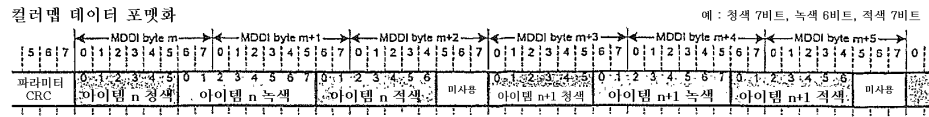
도면98



도면99



도면100



도면101

