

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-173553

(P2005-173553A)

(43) 公開日 平成17年6月30日(2005.6.30)

(51) Int. Cl. ⁷	F I	テーマコード (参考)
G09G 5/00	G09G 5/00 555D	5C082
G06F 3/00	G06F 3/00 Z	5K030
H04L 12/56	H04L 12/56 230Z	

審査請求 未請求 請求項の数 13 O L 外国語出願 (全 72 頁)

(21) 出願番号 特願2004-271144 (P2004-271144)
 (22) 出願日 平成16年9月17日 (2004. 9. 17)
 (31) 優先権主張番号 60/504060
 (32) 優先日 平成15年9月18日 (2003. 9. 18)
 (33) 優先権主張国 米国 (US)
 (31) 優先権主張番号 60/552352
 (32) 優先日 平成16年3月10日 (2004. 3. 10)
 (33) 優先権主張国 米国 (US)
 (31) 優先権主張番号 10/909085
 (32) 優先日 平成16年7月29日 (2004. 7. 29)
 (33) 優先権主張国 米国 (US)

(71) 出願人 502359574
 ジェネシス・マイクロチップ・インコーポ
 レーテッド
 GENESIS MICROCHIP, I
 NC.
 アメリカ合衆国 95002 カリフォル
 ニア、アルビソ、ゴールド・ストリート
 2150
 2150 Gold Street, Al
 viso, CA 95002 U. S. A
 .
 (74) 代理人 110000028
 特許業務法人明成国際特許事務所

最終頁に続く

(54) 【発明の名称】 パケット式ストリームトランスポートスケジューラ及びその使用方法

(57) 【要約】 (修正有)

【課題】 ラスタスキャン転送プロトコルの効率と、等時性パケット転送プロトコルの柔軟性とを有するデータストリームトランスポートプロトコルを提供。

【解決手段】 トランスミッタユニットが接続されたソースデバイスと、レシーバユニットが接続されたシンクデバイスと、トランスミッタユニットにより、ネイティブストリームレートに従ったソースデータストリームを受領し、リンクユニットを経由してトランスミッタユニットとレシーバユニットとを結合し、多数のマルチメディアデータパケットから成るマルチメディアデータパケットストリームを形成、トランスミッタユニットとレシーバユニットとの間のリンクレートに従ってマルチメディアデータパケットストリームを転送するためのトランスポートスケジュールを生成する事により、マルチメディアソースデバイスをマルチメディアシンクデバイスに結合する方法。

【選択図】 図 1

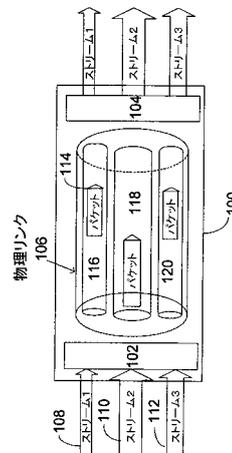


Fig. 1

【特許請求の範囲】**【請求項 1】**

マルチメディアソースデバイスをマルチメディアシンクデバイスに結合するように構成されたパケット式ディスプレイインタフェースであって、

ソースデバイスに接続され、ネイティブストリームレートに従ってソースパケットデータストリームを受領するように構成されたトランスミッタユニットと、

前記シンクデバイスに接続されたレシーバユニットと、

前記トランスミッタユニットと前記レシーバユニットを接続し、前記トランスミッタユニットと前記レシーバユニットとの間で前記ネイティブストリームレートから独立したリンクレートに従い前記ソースパケットデータストリームに基づいて多数のマルチメディアデータパケットから形成されるマルチメディアデータパケットストリームを転送するように構成されたリンクユニットと、

前記リンクユニットに接続され、前記データストリームのうちの選択されたものについて前記リンクユニットを介して転送するためにデータパケットをスケジュールするように構成されたデータパケットスケジューラと、
を備え、

各データストリームのデータパケットサイズは、関連する固定サイズを有し、前記固定サイズはデータストリームビットレートとリンクビットレートとの間の比に依存する、ディスプレイインタフェース。

【請求項 2】

前記マルチメディアデータパケットストリームは、前記ネイティブストリームレートから独立した関連する調整可能なデータストリームリンクレートをそれぞれ有する多数のマルチメディアデータパケットストリームの一つである、請求項 1 記載のディスプレイインタフェース。

【請求項 3】

前記リンクユニットは、更に、

前記トランスミッタユニットから前記レシーバユニットへ前記マルチメディアデータパケットを伝送するように構成された単方向メインリンクと、

前記トランスミッタユニットと前記レシーバユニットとの間を双方向に情報を転送するように構成された双方向補助リンクと、を備える、請求項 1 記載のディスプレイインタフェース。

【請求項 4】

前記双方向補助チャネルは、前記シンクデバイスから前記ソースデバイスへ情報を伝送するように構成された単方向バックチャネルと、前記バックチャネルと協調して前記ソースデバイスから前記シンクデバイスへ情報を伝送するメインチャネルの一部として含まれる単方向フォワードチャネルと、により構成される、請求項 3 記載のディスプレイインタフェース。

【請求項 5】

前記メインリンクユニットは、更に、

前記マルチメディアデータパケットストリームのうちの特定のの一つとそれぞれ関連付けられた多数の仮想リンクを備え、

前記仮想リンクのそれぞれは、関連する仮想リンク帯域幅と仮想リンクレートとを有する、請求項 2 記載のディスプレイインタフェース。

【請求項 6】

メインリンク帯域幅が、前記仮想リンク帯域幅の合計に少なくとも等しい、請求項 5 記載のディスプレイインタフェース。

【請求項 7】

マルチメディアソースデバイスをマルチメディアシンクデバイスに接続する方法であって、

トランスミッタユニットが接続されたソースデバイスを提供するステップと、

10

20

30

40

50

レシーバユニットが接続されたシンクデバイスを提供するステップと、
前記トランスミッタユニットにより、ネイティブストリームレートに従ったソースデータストリームを受領するステップと、
リンクユニットを介して前記トランスミッタユニットと前記レシーバユニットとを接続するステップと、

多数のマルチメディアデータパケットから成るマルチメディアデータパケットストリームを形成するステップと、

前記トランスミッタユニットと前記レシーバユニットとの間のリンクレートに従って前記マルチメディアデータパケットストリームを転送するためのトランスポートスケジュールを生成するステップと、

を備え、

前記マルチメディアデータパケットのそれぞれは、前記リンクレートとデータストリームビットレートとに基づいた固定サイズである、方法。

【請求項 8】

前記トランスミッタユニットから前記レシーバユニットへ前記マルチメディアデータパケットを伝送するように構成された単方向メインリンクを提供するステップと、

前記トランスミッタユニットと前記レシーバユニットとの間を双方向に情報を転送するように構成された双方向補助リンクを提供するステップと、を更に含む、請求項 7 記載の方法。

【請求項 9】

前記双方向補助チャネルは、前記シンクデバイスから前記ソースデバイスへ情報を伝送するように構成された単方向バックチャネルと、前記バックチャネルと協調して前記ソースデバイスから前記シンクデバイスへ情報を伝送するメインチャネルの一部として含まれる単方向フォワードチャネルと、により構成される、請求項 8 記載の方法。

【請求項 10】

前記メインリンクユニットは、更に、

前記マルチメディアデータパケットストリームのうちの特定の一つにそれぞれ関連付けられた多数の仮想リンクを備え、

前記仮想リンクのそれぞれは、関連する仮想リンク帯域幅と仮想リンクレートとを有する、請求項 9 記載の方法。

【請求項 11】

メインリンク帯域幅が、前記仮想リンク帯域幅の合計に少なくとも等しい、請求項 10 記載の方法。

【請求項 12】

データリンクを経由してデータソースとデータシンクとの間での多数のデータパケットの転送をスケジュールする方法であって、

前記データパケットソースから前記データパケットシンクへデータパケット属性を送信するステップと、

前記ソースから前記シンクへ送信されるべき多数のデータストリームのそれぞれについて、ストリームビットレートをデータリンクビットレートと比較するステップと、

前記比較に基づいて、前記データストリームのそれぞれのパケットサイズを固定パケットサイズに設定するステップと、

各データパケットの少なくとも一つを組み合わせるステップと、

組み合わせたデータパケットを前記ソースから前記シンクへ転送するステップと、

を含む方法。

【請求項 13】

データリンクを経由してデータソースとデータシンクとの間での多数のデータパケットの転送をスケジュールするコンピュータプログラム製品であって、

前記データパケットソースから前記データパケットシンクへデータパケット属性を送信するコンピュータコードと、

10

20

30

40

50

前記ソースから前記シンクへ送信されるべき多数のデータストリームのそれぞれについて、ストリームビットレートをデータリンクビットレートと比較するコンピュータコードと、

前記比較に基づいて、前記データストリームのそれぞれのパケットサイズを固定パケットサイズに設定するコンピュータコードと、

各データパケットの少なくとも一つを組み合わせるコンピュータコードと、

組み合わせたデータパケットを前記ソースから前記シンクへ転送するコンピュータコードと、

前記コードを格納するコンピュータ読取可能な媒体と、

を備えるコンピュータプログラム製品。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、マルチメディアデバイスに関する。特に、本発明は、データパケットストリームスケジューラ及びその使用方法について記載する。

【背景技術】

【0002】

ラスタスキャンビデオトランスポートプロトコルは、当初、1つ又は複数の電子銃を使用し、表示画像を一度に一本のラインで物理的に「ペイント」という事実を考慮する必要があるブラウン管(CRT)に基づくディスプレイシステムでの使用のために開発された。例えば、標準解像度(VGA)ビデオ画像は、それぞれ640ピクセルで形成される480本のアクティブ表示ラインを通常含むアクティブ領域で形成される(即ち、解像度640×480)。しかしながら、アクティブ領域に加え、表示されないにもかかわらずビデオ信号に含まれるブランキング領域が含まれ、これはブランキング領域が水平及び垂直両方の帰線に必要な時間量を表すためである。例えば、VGA画像の各フレーム(即ち、それぞれ480ライン×640ピクセルである完全な一フレーム)は、水平帰線のために一ライン当たり約160ピクセルクロックを必要とし、垂直帰線のために約45ライン期間に等しい期間を必要とする。これにより(一ピクセルクロック当たり一ピクセルと仮定すると)、VGA画像を表示するために必要なビデオデータを転送するために必要なビデオ信号は、800ピクセルクロック(640アクティブピクセルクロック+160ブランキングピクセルクロック)程度にする必要がある。そのため、転送効率(合計データストリーム帯域幅上での表示可能データの帯域幅として定義される)は、80%程度となる(即ち、640/800)。

20

30

【0003】

更に最近では、HDTV及びその他のハイエンドグラフィック用途に対応するためにCRTの解像度が増加するにしたがって、ラスタスキャンビデオトランスポートプロトコルの効率は、水平帰線を160ピクセルクロックまでに制限する(これにより関連するブランキング期間を低減する)ように定めることで、約90%までに増加している。例えば、UVGA画像(即ち、1600×1200)について考えると、転送効率は、水平帰線を160ピクセルクロックに維持することで約90%となる(1600/(1600+160))。ラスタスキャンビデオ転送プロトコルは効率的であり(90%程度)、大きなバッファを必要としないが、しかしながら、基本的に、与えられた通りのデータを表示することのみが可能であるという点において柔軟性がない。

40

【0004】

ラスタスキャンビデオプロトコルに加えて、デジタルビデオに基づくシステムの出現が、デジタルビデオトランスポートプロトコルの必要性を形成している。I.E.E.E.1394又はFireWire™と呼ばれるこうしたデジタルビデオトランスポートプロトコルの一つは、均一なビットレートを保証し、(ビデオストリームとオーディオストリームの形態の関連するサウンドトラックとのような)多数のデータストリーム間の同期を維持するために、大きなバッファ(60Kb程度)に依存する等時性パケットトランスポ

50

ートに基づいている。等時性パケット転送プロトコルは（パケットに基づく性質のため）生得的に柔軟だが、大きなバッファという要件は、非常に高いコストを要する可能性がある。

【0005】

そのため、望ましいものは、ラスタスキャン転送プロトコルの（転送効率とメモリリソース利用率との両方の観点での）効率と、等時性パケット転送プロトコルの柔軟性とを有するデータストリームトランスポートプロトコルである。

【発明の開示】

【0006】

トランスミッタユニットが接続されたソースデバイスを提供するステップと、レシーバユニットが接続されたシンクデバイスを提供するステップと、トランスミッタユニットにより、ネイティブストリームレートに従ったソースデータストリームを受領するステップと、リンクユニットを経由しトランスミッタユニットとレシーバユニットとを接続するステップと、多数のマルチメディアデータパケットから成るマルチメディアデータパケットストリームを形成するステップと、トランスミッタユニットとレシーバユニットとの間のリンクレートに従ってマルチメディアデータパケットストリームを転送するためのトランスポートスケジュールを生成し、これにおいてマルチメディアデータパケットが、それぞれ、リンクレートとデータストリームビットレートとに基づいた固定サイズになるステップとにより、マルチメディアソースデバイスをマルチメディアシンクデバイスに結合する方法。

【0007】

データパケットソースからデータパケットシンクへデータパケット属性を送信するステップと、ソースからシンクへ送信されるべき多数のデータストリームのそれぞれについて、ストリームビットレートをデータリンクビットレートと比較するステップと、この比較に基づいてデータストリームのそれぞれのパケットサイズを固定パケットサイズに設定するステップと、各データパケットの少なくとも一つを組み合わせるステップと、組み合わせたデータパケットをソースからシンクへトランスポートするステップとにより、データリンクを経由してデータソースとデータシンクとの間での多数のデータパケットのトランスポートをスケジュールする方法。

【0008】

更に別の実施形態においては、データパケットソースからデータパケットシンクへデータパケット属性を送信するコンピュータコードと、ソースからシンクへ送信されるべき多数のデータストリームのそれぞれについて、ストリームビットレートをデータリンクビットレートと比較するコンピュータコードと、この比較に基づいてデータストリームのそれぞれのパケットサイズを固定パケットサイズに設定するコンピュータコードと、各データパケットの少なくとも一つを組み合わせるコンピュータコードと、組み合わせたデータパケットをソースからシンクへトランスポートするコンピュータコードと、これらのコードを格納するコンピュータ読取可能な媒体とを含む、データリンクを経由してデータソースとデータシンクとの間での多数のデータパケットのトランスポートをスケジュールするコンピュータプログラム製品。

【発明を実施するための最良の形態】

【0009】

次に、本発明の特定の実施形態を詳細に参照し、本発明の例は、添付の図面に例示される。本発明については、特定の実施形態に関連させて説明するが、説明する実施形態に本発明を限定することは意図されないと理解されるであろう。反対に、代替、修正、及び等価物を、付記する請求項によって定義されるような本発明の趣旨及び範囲内に含まれ得るものとして網羅することが意図される。

【0010】

本発明は、パケットベースのデジタルインタフェースを経由して、ビデオシンク又はレシーバに結合されたビデオソースを有するビデオディスプレイシステムの観点から説明さ

10

20

30

40

50

れる。ソースデバイスに結合されるトランスミッタユニットは、それぞれが関連ストリーム属性を有する任意の数のパケット化ビデオストリームを受領する。説明するビデオシステムに関して、こうした属性は、ビデオ形式と、色深度と、その他を含む。レシーバユニットは、データリンク又はメインリンクと、データパケットの伝送前にメインリンクを経由してソースからレシーバへストリーム属性データを転送するために部分的に使用される関連補助リンクとを経由して、ソースに結合される。これにより、パケットヘッダは、データパケットがどのデータストリームに関連するかを最初に特定するために使用され、したがって、ストリームID又はその他のこのような識別子のみを含むことができる。これにより、パケットオーバーヘッドは実質的に低減され、パケットオーバーヘッドは、ビデオデータ及びオーディオデータのようなマルチメディアコンテンツのためのメインリンク帯域幅が失われないようにし、効率的なパケットトランスポートメカニズムを提供する。メインリンクでのデータの伝送を調整するために、トランスポートストリームスケジューラは、ソースからシンクへストリーム属性データを送信する補助チャンネルが更に提供され得るデータリンク上での任意の数のパケット化データストリームをスケジューリングする、柔軟で効率的なシステム、方法、及び装置を提供する。

【0011】

各データストリームは、多数の関連データパケットで形成されており、関連データパケットのサイズは、特定のデータストリームに必要なリンク帯域幅の相対部分によって決まる。特定のデータストリーム i について、対応するパケットサイズ PS_i は、以下の関係を通じて、最大パケットサイズ MPS と、リンクビットレート LBR と、ストリームビットレート SBR_i とに関連付けられている。

$$PS_i = MPS * SBR_i / (LBR_i + 1)$$

【0012】

これにより、パケットサイズは、データリンク帯域幅と比較したデータストリームの相対帯域幅に基づいて、各データストリームについて決定される。例えば、最大パケットサイズが64データシンボルで、リンクビットレート LBR が一レーン当たり 2.5 Gbps である場合において、表1は、選択されたストリームビットレートに対応する代表的なパケットサイズを示す。

【表1】

リンクビットレート (LBR)	ストリームビットレート (SBR)	パケットサイズ (PS)
2.5 Gbps / レーン	1.0 Gbps / レーン	32 リンクシンボル
2.5 Gbps / レーン	0.3125 Gbps / レーン	10 リンクシンボル
2.5 Gbps / レーン	0.25 Gbps / レーン	8 リンクシンボル
レーン間アライメント (ILA)		2 リンクシンボル

【0013】

スケジューラは、多数のストリームのパケットを対応するリンクデータストリームへと(トランスミッタにおいて)時分割多重化及び(レシーバにおいて)逆多重化する。望ましい実施形態において、トランスミッタは、例えば、最大データリンクレート、利用可能バッファサイズ、及び時間基準回復(タイムベースリカバリ。TBR)ユニットの数に関して、レシーバの能力を読み出す。この知識により、トランスミッタは、最も効率的なト

ランスポート構成と、レシーバが後続のデータストリームに対応できるかを、レシーバに対する追加の問い合わせを全く行う必要なく判断できる。データストリームランスポートを開始する前に、トランスミッタは、レシーバに対して、ビデオデータ、色形式及び深度、ジオメトリに関するもの及び各データストリームに関連するパケットサイズといったストリーム属性を通知する。この属性データを通信することで、パケットヘッダのサイズは、ストリームIDのみが必要となる程度まで実質的に低減できる。これにより、大きなサイズのパケットヘッダにより実質的に大きなオーバーヘッドを必要とする従来のパケットランスポートプロトコルに比べ、転送効率は大幅に増加する。

【0014】

本発明のストリームスケジューラを説明するための基盤を提供するために、本発明の実施に十分に適した代表的なデジタルビデオシステムについて説明する。 10

【0015】

図1は、本発明の任意の数の実施形態を実施するのに十分に適したパケット式デジタルビデオディスプレイインタフェースの一般化表現を図示している。インタフェース100は、物理リンク106（パイプとも呼ばれる）を経由して、トランスミッタ102をレシーバに接続する。説明する実施形態では、多数のデータストリーム108～112がトランスミッタ102に受領され、トランスミッタ102は、必要であれば、それぞれを対応する数のデータパケット114にパケット化する。こうしたデータパケットは、その後、対応するデータストリームの形態にされ、それぞれのデータストリームは、関連する仮想パイプ116～120を経由してレシーバ104に渡される。各仮想リンクのリンクレート（即ち、データパケット転送レート）は、特定のデータストリームのために最適化可能であり、結果として、（特定のデータストリームに応じて、それぞれ互いに異なる可能性のある）関連リンクレートをそれぞれが有するデータストリームを伝送する物理リンク106が生じることに留意されたい。データストリーム110～114は、ビデオ、グラフィック、オーディオ、及びその他といった、任意の数の形態をとることができる。 20

【0016】

通常、ソースがビデオソースである時、データストリーム110～114は、コンポジットビデオ、シリアルデジタル、パラレルデジタル、RGB、又は家庭用デジタルビデオといった、任意の数及びタイプの周知の形式を有する可能性がある様々なビデオ信号を含む。アナログテレビ、スチルカメラ、アナログVCR、DVDプレイヤー、ビデオカメラ、レーザディスクプレイヤー、TVチューナ、セットトップボックス（及び衛星DSS又はケーブル信号）、及びその他といった何らかの形態のアナログビデオソースをソース102が含む場合、ビデオ信号はアナログビデオ信号にできる。ソース102は、更に、例えば、デジタルテレビ（DTV）、デジタルスチルカメラ、及びその他といったデジタル画像ソースを含むことができる。デジタルビデオ信号は、SMPTE 274M-1995（解像度1920×1080、プログレッシブ又はインタレーススキャン）、SMPTE 296M-1997（解像度1280×720、プログレッシブスキャン）、及び標準の480プログレッシブスキャンビデオといった任意の数及びタイプの広知のデジタル形式にできる。 30

【0017】

ソース102がアナログ画像信号を提供する場合、アナログ-デジタルコンバータ（A/D）は、アナログ電圧又は電流信号を別個の一連のデジタルエンコードされた数（信号）に変換し、このプロセスにおいて、デジタル処理に相応しい適切なデジタル画像データワードが形成される。広範なA/Dコンバータのいずれかを使用できる。一例として、その他のA/Dコンバータは、例えば、Philips、Texas Instrument、Analog Device、Brooktree、及びその他が製造したものを含む。 40

【0018】

例えば、データストリーム110がアナログタイプの信号である場合、トランスミッタに含まれる、又はトランスミッタに結合されたアナログ-デジタルコンバータ（図示なし）は、アナログデータをデジタル化し、その後、デジタル化データストリーム110を多 50

数のデータパケット114にパケット化するパケタイザがパケット化を行い、それぞれのデータパケット114は、仮想リンク116を経由して、レシーバ104に伝送されることになる。レシーバ104は、次に、データパケット114を再び適切に組み合わせることで、オリジナルの形式にすることで、データストリーム110を再構成する。リンクレートは、ネイティブストリームレートから独立していることに留意されたい。唯一の要件は、物理リンク106のリンク帯域幅が伝送対象の(複数の)データストリームの総帯域幅より高いことである。説明する実施形態において、着信データ(ビデオデータの場合のピクセルデータ等)は、データマップ定義に基づいて、それぞれの仮想リンク上でパックされる。これにより、物理リンク106(又は構成する仮想リンクのいずれか)は、DVIのような従来の相互接続で行われるように、一リンクキャラクタクロック当たり一ピクセルデータを伝送しない。

10

【0019】

これにより、インタフェース100は、ビデオデータ及びグラフィックデータのみではなく、必要に応じて、オーディオ及びその他のアプリケーションデータのトランスポートのためのスケーラブルな媒体を提供する。加えて、本発明は、ホットプラグイベント検出をサポートし、物理リンク(又はパイプ)を最適な伝送レートに自動的に設定する。本発明は、多数のプラットフォームに適した全てのディスプレイのために、低ピン数、純デジタルのディスプレイ相互接続を提供する。こうしたプラットフォームは、ディスプレイのホスト、ラップトップ/オールインワン、及びHDTVその他の家庭用電子機器での応用を含む。

20

【0020】

ビデオデータ及びグラフィックデータを提供することに加え、表示タイミング情報をデジタルストリームに埋め込み、本質的に完全かつ即時的な表示調整を提供し、「自動調整」のような機能の必要性を除去できる。本発明のインタフェースのパケットに基づく性質は、多数のビデオ/グラフィックストリーム及びマルチメディア用途でのオーディオストリームといった多数のデジタルデータストリームをサポートするスケーラビリティを提供する。加えて、周辺機器用のユニバーサルシリアルバス(USB)トランスポートと表示制御とを、追加のケーブルの必要性なしで提供できる。

【0021】

本発明のディスプレイインタフェースのその他の実施形態について、下で説明する。

30

【0022】

図2は、ビデオソース200とビデオディスプレイユニット204とを接続するために使用される、図1に図示したシステム100に基づくシステム200を例示している。例示した実施形態において、ビデオソース202は、デジタル画像(又はデジタルビデオソース)206及びアナログ画像(又はアナログビデオソース)208の一方又は両方を含むことができる。デジタル画像ソース206の場合、デジタルデータストリーム210がトランスミッタ102に提供されるが、アナログビデオソース208の場合には、結合されたA/Dコンバータユニット212が、アナログデータストリーム213を対応するデジタルデータストリーム214に変換する。デジタルデータストリーム214は、その後、トランスミッタ102によって、デジタルデータストリーム210とほぼ同じ形で処理される。ディスプレイユニット204は、アナログタイプのディスプレイ又はデジタルタイプのディスプレイにすることが可能であり、或いは一部のケースでは、提供されたアナログ又はデジタル信号のどちらでも処理できる。いずれの場合においても、ディスプレイユニット204は、レシーバとディスプレイ218、及びアナログタイプのディスプレイの場合にはD/Aコンバータユニット220とのインタフェースをとるディスプレイインタフェース216を含む。説明する実施形態において、ビデオソース202は、任意の数の形態(パーソナルデスクトップコンピュータ、デジタル又はアナログテレビ、セットトップボックス、その他)をとることが可能であり、ビデオディスプレイユニット104は、ビデオディスプレイ(LCD型ディスプレイ、CRT型ディスプレイ、その他)の形態をとることが可能である。

40

50

【 0 0 2 3 】

しかしながら、ビデオソース又はビデオシンクのタイプに関係なく、様々なデータストリームは、物理リンク 1 0 6 上での伝送の前に（必要に応じて）デジタル化及びパケット化され、物理リンク 1 0 6 は、等時性データストリームのための単方向メインリンク 2 2 2 と、リンク設定及びビデオソース 2 0 2 とビデオディスプレイ 2 0 4 との間でのその他のデータトラフィック（様々なリンク管理情報、ユニバーサルシリアルバス（U S B）データ、その他）のための双方向補助チャンネル 2 2 4 とを含む。

【 0 0 2 4 】

メインリンク 2 2 2 は、これにより、多数の等時性データストリーム（多数のビデオ/グラフィックストリーム及びマルチチャンネルオーディオストリーム）を同時に伝送することができる。説明する実施形態において、メインリンク 2 2 2 は、多数の異なる仮想チャンネルを含み、仮想チャンネルのそれぞれは、数ギガビット/秒（G b p s）で等時性データストリーム（非圧縮グラフィック/ビデオ及びオーディオデータ）を転送できる。そのため、論理的な観点では、メインリンク 2 2 2 は、単一の物理パイプとして捉えられ、この単一の物理パイプ内部では、多数の仮想パイプを確立できる。これにより、論理データストリームが物理チャンネルに割り当てられるのではなく、各論理データストリームが自分の論理パイプ（即ち、上記の仮想チャンネル）において伝送される。

10

【 0 0 2 5 】

説明する実施形態において、メインリンク 2 2 2 の速度又は転送レートは、リンク条件を補償するために調整可能である。例えば、一実施において、メインリンク 2 2 2 の速度は、一チャンネル当たり、最も遅い速度である約 1 . 0 G b p s ~ 約 2 . 5 G b p s に近い範囲において、約 0 . 4 G b p s の増分で調整できる（図 3 参照）。一チャンネル当たり 2 . 5 G b p s において、メインリンク 2 2 2 は、単一のチャンネル上で、S X G A 6 0 H z を一ピクセル当たり 1 8 ビットの色深度でサポートできる。チャンネル数の減少により、相互接続のコストが減少するだけでなく、ポータブルデバイス及びその他といった電力に敏感な用途にとって重要な考慮事項（及び望ましい状態）である電力消費量の減少も発生することに留意されたい。しかしながら、チャンネル数を四本に増加させることで、メインリンク 2 2 2 は、6 0 H z の W Q S X G A (3 2 0 0 × 2 0 4 8) を一ピクセル当たり 2 4 ビットの色深度で、或いは 6 0 H z の Q S X G A (2 5 6 0 × 2 0 4 8) を一ピクセル当たり 1 8 ビットの色深度で、データ圧縮することなくサポートできる。最低レートの一チャンネル当たり 1 . 0 G b p s でも、非圧縮 H D T V (即ち、1 0 8 0 i 又は 7 2 0 p) データストリームをサポートするために必要なチャンネルは僅か二本である。

20

30

【 0 0 2 6 】

説明する実施形態において、メインリンクのデータレートは、その構成要素である仮想リンクの総帯域幅を上回る帯域幅となるものが選択される。インタフェースに送信されたデータは、ネイティブレートでトランスミッタに到達する。レシーバ 1 0 4 内の時間基準回復（T B R）ユニット 2 2 6 は、必要な場合には、メインリンクデータパケットに埋め込まれたタイムスタンプを使用して、ストリームのオリジナルのネイティブレートを再生する。しかしながら、図 2 B に図示した適切に構成されたデジタルディスプレイデバイス 2 3 2 では、ディスプレイデータがリンクキャラクタクロックレートでディスプレイドライバ電子回路に送信されるため、時間基準回復は不要であり、これにより、必要なチャンネル数は、ディスプレイの複雑性及びコストの相応の低減と共に、大幅に減少することに留意されたい。例えば、図 2 C は、アレイ 2 4 0 内の選択された表示素子 2 3 8 を駆動するためにロウドライバ 2 3 6 と組み合わせて使用される様々なカラムドライバ 2 3 4 へ、ディスプレイデータが本質的にパイプラインされるため、時間基準回復が存在しない形で構成された例示的な L C D パネル 2 3 2 を表している。

40

【 0 0 2 7 】

その他の実施形態では、リンクレート及びピクセル/オーディオクロックレートの単純な算出法について説明する。現時点で存在する全ての標準ピクセル/オーディオクロック周波数は、以下のマスタ周波数のサブセットであることが研究及び理解されている。

50

23.76GHz=210ラ33ラ57ラ111Hz

【0028】

これは、ピクセル（又はオーディオ）クロックレートを、四つのパラメータA、B、C、及びDによって、次のように表現できることを意味する。

ピクセルクロックレート = $2^A \times 3^B \times 5^C \times 11^D$

A = 4ビット、B = 2ビット、C = 3ビット、及びD = 1ビット

【0029】

リンクレート（8B/10B文字のような10ビット文字を使用するリンクにおいて、シリアルリンクビットレート/10）がピクセルクロックレートとは異なる可能性があるリンクについても、こうした四つのパラメータA'、B'、C'、及びD'によってリンクレートを定義することには利点がある。この利点は、リンククロックからピクセル/オーディオクロックを再生する際の容易さである。例えば、リンクレートがA' = 6、B' = 3、C' = 7、及びD' = 0に設定され、対応するリンクレートが135MHzであるとする。しかしながら、ピクセルクロックレートがA = 8、B = 3、C = 6、及びD = 0（= 108MHz）に設定されると仮定すると、ピクセルクロックは、ピクセルクロックレートがリンクレート * 22/51に等しくなるように、リンククロックから生成できる。

10

【0030】

時間基準回復を必要とするシステムを再び参照すると、時間基準回復ユニット226は、デジタルクロックシンセサイザとして実施してよい。非圧縮ビデオストリームについて、タイムスタンプは、パケットヘッダに格納され、下で更に詳細に説明するように20ビット値となる。一定のストリームにおいて、20ビットのうちの各四ビットが、各ヘッダにおいて連続的に格納される（TS3-0、TS7-4、TS11-8、TS15-12、TS19-16）。ネイティブストリーム周波数（Freq_native）は、次のように、リンク文字クロック周波数（Freq_link_char）から取得される。

20

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

【0031】

トランスミッタ102は、220サイクルのリンク文字クロック周波数期間においてネイティブストリームクロック数をカウントすることで、このタイムスタンプを生成する。カウンタは、220サイクルのリンク文字クロック毎に値を更新する。これら二つのクロックは互いに非同期であるため、タイムスタンプ値は、時間と共に1ずつ変化することになる。更新と更新との間に、トランスミッタ102は、一定のパケットストリームのヘッダにおいて、同じタイムスタンプを繰り返し送信する。タイムスタンプ値の（1カウントより大きい）急変は、レシーバによって、ストリームソースの不安定状態を示すものと解釈される。

30

【0032】

オーディオストリームでは、タイムスタンプが通信されないことに留意されたい。この場合、ソースデバイスは、ディスプレイデバイスに対して、オーディオサンプルレートと一サンプル当たりのビット数とを通知する。下のようにオーディオレート及びリンク文字レートを決定することで、ディスプレイデバイスは、オリジナルのオーディオストリームレートを再生する。

40

オーディオレート = (オーディオサンプルレート) × (サンプル当たりビット数) × (チャンネル数)

【0033】

図4Aに図示したメインリンクデータパケット400は、図4Bに図示したようなメインリンクパケットヘッダ402を含み、パケットヘッダ402は、16ビットの形態であり、ビット3～0はストリームID（SID）（最大ストリームカウントが16であることを示す）、ビット4はタイムスタンプ（TS）のLSBである。ビット4が1に等しい時、このパケットヘッダは、タイムスタンプ値の最下位の四ビットを有する（非圧縮ビデオストリームに対してのみ使用される）。ビット5は、ビデオフレームシーケンスビット

50

であり、ビデオフレーム境界において「0」から「1」或いは「1」から「0」にトグルするフレームカウンタの最下位ビットの役割を果たす（非圧縮ビデオストリームに対してのみ使用される）。ビット7及び6はリザーブされるが、ビット8～10は、先行する8ビットのエラーをチェックする四ビットCRC（CRC）となる。ビット15～12は、タイムスタンプ/ストリームID反転（TSP/SIDn）であり、非圧縮ビデオでは、20ビットタイムスタンプ値のうちの四ビットとして使用される。

【0034】

本発明のインタフェースの利点の一つは、それぞれ異なる形式にすることが可能な異なるデータストリームを多重化し、更に、一定のメインリンクデータパケットに多数のサブパケットを包含させる能力である。例えば、図5Aは、本発明の実施形態による、サブパケットエンクロージャ及び複数パケット多重化を提供するように構成されたシステム500を図示している。システム500は、図2に図示したシステム200の特定の実施形態であり、したがって、本発明の範囲及び意図のいずれかを限定するものと解釈されるべきではないことに留意されたい。システム500は、トランスミッタ102に含まれ、ストリーム1補足データストリーム504をデータストリーム210に組み合わせて多重化データストリーム506を形成するために使用される、ストリームソースマルチプレクサ502を含む。多重化データストリーム506は、その後、リンク層マルチプレクサ508に転送される。任意の数のデータストリームを組み合わせ、多重化メインリンクストリーム510を形成し、多重化メインリンクストリーム510は、多数のデータパケット512で形成され、データパケット512の一部は、内部に封入された任意の数のサブパケット514を含んでもよい。リンク層マルチプレクサ516は、ストリームID（SID）と関連サブパケットヘッダとに基づいて、多重化データストリーム510を分割し、構成要素であるデータストリームとし、一方、ストリームシンクマルチプレクサ518は、サブパケットに収容されたストリーム1補足データストリームを更に分割する。

【0035】

図6は、三本のストリームがメインリンク222上で多重化される時の図5に図示したストリーム510の例として、多重化メインリンクストリーム600の高レベル図を図示している。この例の三本のストリームは、UXGAグラフィックス（ストリームID=1）と、1280×720pビデオ（ストリームID=2）と、オーディオ（ストリームID=3）とである。メインリンクパケット400の小さなパケットヘッダサイズは、パケットオーバーヘッドを最小化し、結果として、非常に高いリンク効率を生み出す。パケットヘッダをこれほど小さくできるのは、メインリンク222でのパケットの伝送前に、補助チャネル224を介して、パケット属性が通信されるためである。

【0036】

一般的に言えば、非圧縮ビデオデータストリームはビデオブランキング期間に対応するデータアイドル期間を有するため、サブパケットエンクロージャは、メインパケットストリームが非圧縮ビデオである時に効果的な方式となる。したがって、非圧縮ビデオストリームの形態であるメインリンクトラフィックは、この期間中に一連のNull特殊文字を含むことになる。様々なデータストリームを多重化する能力を十分に利用することで、本発明の特定の実施は、ソースストリームがビデオデータストリームである時に、様々な方法を使用して、メインリンクレートとピクセルデータレートとの間の差異を補正する。例えば、図7に例示したように、ピクセルデータレートは、0.5Gb/秒となり、2ns毎にピクセルデータのビットが送信されるようになる。この例において、リンクレートは、1.25Gb/秒に設定されており、0.8ns毎にピクセルデータのビットが送信されるようになっている。ここで、トランスミッタ102は、図8に例示したように、ピクセルデータ間に特殊文字を散在させる。二つの特殊文字は、ピクセルデータの第一のビットP1とピクセルデータの第二のビットP2との間に配置される。特殊文字によって、レシーバ104は、ピクセルデータの各ビットを区別することが可能となる。ピクセルデータのビット間に特殊文字を散在させることで、更に、リンクが同期を維持可能な安定したデータのストリームが形成される。この例において、特殊文字はNull文字である。こ

10

20

30

40

50

うした方法では、リンクレートが十分に高速であるため、ラインバッファは不要であり、小さなFIFOのみが必要となる。しかしながら、受信側では、ビデオ信号を再構築するために、相対的に多くのロジックが必要となる。レシーバは、特殊文字の開始及び終了の時期を認識する必要がある。

【0037】

散在法の代替は、ピクセルデータの連続ビットを、ヌル値のような特殊文字と交互に配置することである。例えば、P1からP4をトランスミッタ104に含まれるラインバッファに供給し、その後、更なるピクセルデータが利用可能になるまで、一つ以上のヌル値をバッファに供給することができる。こうした実施には、上記の散在法よりも相対的に大きなバッファスペースが必要となる。こうした多くの実施では、相対的に高速なリンク速度のため、ラインバッファを満たすのに必要な時間は、ラインバッファが一杯になった後でデータを伝送するのに必要な時間を上回ることになる。

10

【0038】

図5Aを参照して説明したように、本発明のインタフェースの利点の一つは、様々なデータストリームを多重化する能力だけでなく、更に特定のメインリンクデータパケット内に任意の数のサブパケットを封入することである。図9Aは、本発明の実施形態による、代表的なサブパケット900を図示している。サブパケット900は、サブパケットヘッダ902を含み、説明する実施形態において、サブパケットヘッダ902は2バイトであり、SPS(サブパケットスタート)特殊文字を伴う。サブパケット900が封入されるメインリンクデータパケットがサブパケット900に加えてパケットペイロードを含む場合には、SPE(サブパケットエンド)特殊文字によって、サブパケット900の終了を示す必要がある。そうでない場合には、メインパケットの終了(図9Bに図示した例ではCOM文字に続くことで示される)が、サブパケット902と、封入するメインパケットとの両方の終了を示す。しかしながら、サブパケットは、サブパケットを封入するメインパケットがペイロードを有していない時、SPEで終了する必要はない。図9Bは、本発明の実施形態による、メインリンクパケット内の例示的なサブパケット形式を図示している。ヘッダフィールド及びサブパケットペイロードの定義は、サブパケット902を使用する特定のアプリケーションのプロファイルに応じて変化することに留意されたい。

20

【0039】

特に有用なサブパケットエンクロージャの使用法の例は、図10に図示した非圧縮グラフィックス画像1000の選択的リフレッシュである。フレーム1002全体の属性(水平/垂直合計、画像幅/高さ、その他)については、ストリームが有効であり続ける限り、こうした属性は一定のままであるため、補助チャンネル224を介して通信されることになる。選択的リフレッシュ動作では、画像1000の一部1004のみが、ビデオフレーム毎に更新される。更新された(複数の)長方形(即ち、部分1004)の四つのX-Y座標は、フレーム毎に伝送する必要があり、これは長方形座標の値がフレーム毎に変化するためである。別の例は、256色グラフィックデータに必要なカラーlookupアップテーブル(CLUT)の伝送であり、この場合、8ビットピクセルデータが256エントリCLUTに対するエントリとなり、CLUTのコンテンツは、動的に更新されなければならない。

30

40

【0040】

単一の双方向補助チャンネル224は、リンク設定に有用な、メインリンクの動作をサポートする様々なサポート機能のための経路で、更にUSBトラフィック等の補助アプリケーションデータを伝送する経路を提供する。例えば、補助チャンネル224により、ディスプレイデバイスは、ソースデバイスに、同期消失等のイベント、欠落パケット、及びトレーニングセッションの結果(下記)を通知できる。例えば、特定のトレーニングセッションが失敗した場合、トランスミッタ102は、失敗したトレーニングセッションの事前に選択された或いは決定された結果に基づいて、メインリンクレートを調整する。これにより、調整可能な高速メインリンクと、相対的に低速で非常に信頼性の高い補助チャンネルとを組み合わせることで形成される閉ループは、様々なリンク条件での堅牢な動作を可能に

50

する。一部のケースでは（図 5 B に図示した例）、データをソースデバイス 202 からシンクデバイス 204 に転送するメインリンク 222 の帯域幅の一部 522 と、シンクデバイス 204 からソースデバイス 202 への単方向バックチャンネル 524 とを使用して、論理双方向補助チャンネル 520 を確立することに留意されたい。一部の応用では、この論理双方向補助チャンネルの使用は、図 5 A において説明した半二重双方向チャンネルを使用することより望ましい場合がある。

【0041】

実際のパケットデータストリームの伝送を開始する前に、トランスミッタ 102 は、概念においてモデムのリンク設定に類似するリンクトレーニングセッションを通じて、安定したリンクを確立する。リンクトレーニング中、メインリンクトランスミッタ 102 は、
 10 確実なビット/文字ロックが達成可能かをレシーバ 104 が判断できるように、所定のトレーニングパターンを送信する。説明する実施形態では、トランスミッタ 102 とレシーバ 104 との間でのトレーニングに関連するハンドシェイクが、補助チャンネル上で実行される。本発明の実施形態による、リンクトレーニングパターンの例は、図 11 に図示される。例示のように、トレーニングセッション中、段階 1 は、イコライザを最適化するためにレシーバによって使用される最短のランレングスを表し、一方、段階 2 は、最長のものである。段階 3 において、ビットロック及び文字ロックは、両方とも、リンク品質が妥当である限り達成される。通常、トレーニング期間は、約 10 ms であり、この時間内に、約 107 ビットのデータが送信される。レシーバ 104 は、確実なロックを達成しない場合、補助チャンネル 224 を介してトランスミッタ 102 に通知を行い、トランスミッタ 1
 20 02 は、リンクレートを低下させて、トレーニングセッションを繰り返す。トレーニングセッション経路を提供することに加え、補助チャンネル 224 は、更に、メインリンクパケットストリームの説明を伝送するために使用可能であり、これにより、メインリンク 222 でのパケット伝送のオーバーヘッドは大幅に減少する。更に、補助チャンネル 224 は、全てのモニタで見られるディスプレイデータチャンネル（DDC）の代わりに、拡張表示識別データ（EDID）情報を伝送するように構成できる（EDID は、VESA 規格のデータ形式で、ベンダ情報と、最大画像サイズと、色特性と、工場プリセットタイミングと、周波数範囲限界と、モニタ名及びシリアル番号の文字列とを含め、モニタ及びその機能に関する基本情報を含む。情報は、ディスプレイに格納され、モニタと PC グラフィックス
 30 アダプタとの間に位置する DDC を通じてシステムと通信するために使用される。システムは、この情報を構成の目的で使用するため、モニタ及びシステムは連動が可能となる）。拡張プロトコルモードと呼ばれるものにおいて、補助チャンネルは、キーボード、マウス、及びマイクロフォンといった追加的なデータタイプをサポートするために、必要に応じて、非同期パケット及び等時性パケットの両方を伝送できる。

【0042】

図 12 は、本発明の実施形態による、データスケジューラ 1202 を有する代表的なシステム 1200 を図示している。システム 1200 は、図 5 A 及び 5 B を参照して説明したシステムに基づいており、そのため、本発明の任意の数の実施の一つに過ぎないと考えるべきであることに留意されたい。したがって、ストリームスケジューラ 1202 は、ビデオソース 202 に組み込まれるか、或いは結合され、続いて、マルチプレクサ 1204
 40 と、スケジューラサイクル時間 T_{sched} と呼ばれる期間中にリンクデータストリーム 1208 を送り込むのに使用される着信データストリーム（S1、S2、及び S3）の部分のみを格納するのに適したデータバッファ 1206 とに結合される。説明する実施形態において、データバッファ 1206 は、FireWireTM 等の等時性ビデオ転送プロトコルで代表的な 60 K バイトを超えるものとは対照的に、通常数十バイト程度のサイズである。これにより、リンク効率（利用可能帯域幅で除算したデータストリームのペイロード帯域幅合計の比較に基づく）は、およそ 90% 程度以上となる。

【0043】

説明する実施形態において、リンクデータストリーム 1208 は、データストリーム S1、S2、及び S3 のそれぞれからのサイズデータパケット P1、P2、及び P3（サイ
 50

ズは、リンク帯域幅との関係における各データストリームの相対帯域幅を反映する)を組み合わせるために時分割多重化を使用することで、スケジューラ1202によって形成される。以前に説明したように、各データパケットのサイズは、特定のデータストリームビットレート(SBR)とリンクビットレート(LBR)との関数である。特に、特定のデータストリームビットレートが大きくなれば、表1の例で示したように、特定のデータパケットは大きくなる。例えば、リンクビットレートLBRが2.5Gbps程度であり、最大パケットサイズがおよそ80リンクシンボル程度であり(一リンクシンボルは、一リンククロック当たりのデータ単位として定義され、通常は4nsである)、表1の前提(即ち、SBR1が1.05Gbps、SBR2が0.3125Gbps、SBR3が0.25Gbps)を使用する場合、ストリームS1に関連するデータパケットP1は、32リンクシンボルとなり、一方、ストリームS2及びS3に関連するデータパケットP2及びP3は、それぞれ10リンクシンボル及び8リンクシンボルとなる。各スケジューラサイクル時間 T_{sched} の開始時には、スケジューラ1202が、通常は2リンクシンボル程度のサイズである、レシーバ204用のアライメントツールを提供するレーン間アライメントパケット(ILA)を挿入することに留意されたい。そのため、この例においては、スケジューラサイクル時間 T_{sched} は、 $(32 + 10 + 8 + 2) \cdot 52$ リンクシンボル程度となる(各リンクシンボルが約4nsを表す時、約208nsと解釈される)。

【0044】

更に、各データパケットPは、次のようなストリームビットレート(SBR)及びリンクビットレート(LBR)に関するデータシンボルDの数とスタッフィングシンボルNとの関連アクティブデータ比を有することに留意されたい。

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

【0045】

したがって、データストリームが追加(又は削除)され、スケジューラサイクル時間 T_{sched} の増加(又は減少)が生じる状況においては、データシンボルDの数との関係において、スタッフィングシンボルNの数を変化させることで、パケットサイズPは、一定状態を維持する。データストリームの追加(又は削除)により、特定のパケットサイズは同じ状態を維持し、他のデータストリーム及びスケジューラサイクル時間 T_{sched} は増加(又は減少)するため、スタッフィングシンボルNの数は、 T_{sched} の変化に比例して増加(又は減少)する。単一のデータストリームのみが残る「縮退」のケースでは、スタッフィングシンボルNは存在しなくなる。

【0046】

図13は、実施形態によるデータストリーム1210の更に詳細な部分1300を図示している。特に、図13は、表1に示した値を使用したデータストリーム1208のデータシンボルDとスタッフィングシンボルNの配置を図示している。更に、特定のデータストリームを動的に追加又は削除することで、残りのデータストリームの特定のパケットサイズが影響されないままとすることに留意されたい。したがって、図14は、20リンクシンボルのパケットサイズP4に対応する0.625Gbpsのストリームビットレートを有する第四のデータストリームS4の追加を例示しており、この結果、 T_{sched} は52から72リンクシンボル(4nsに等しいリンクシンボルでは288nsに相当する)に増加する。しかしながら、特定のパケットサイズP1、P2、及びP3を一定に保つために、スタッフィングシンボルNの数は、各データパケットについて増加する。反対に、(例えばS3のような)データストリームが削除される場合、スケジューラサイクル時間 T_{sched} は、スタッフィングシンボルNの数に関するデータシンボルDの数の相応の増加に応じて減少することになる。一つ以外の全データストリームが削除される「縮退」のケースでは、残存するデータパケットは、スタッフィングシンボルNを有しておらず、何らかのバッファの必要性も存在せず、これにより、上で説明したラスタスキャントランスポートプロトコルをシミュレートする。

【0047】

リンクデータストリーム1208が(図15に図示するような)単一の非圧縮ビデオス

トリームである時の縮退接続のケースにおいて、ILAパケットは、S1のアイドル期間（水平ランキング領域）に配置され、アクティブな表示領域は、その後、データシンボルDとスタッフィングシンボルNとの混合によって表現される（図16参照）。

【0048】

更に、特定のデータストリームに関係しないデータシンボルと関連させて、特定のデータストリームのデータシンボルDの数をカウントすることで、問題のデータストリームのストリームクロックが提供されることから、データシンボルDの相対数が、埋め込みタイムスタンプを提供することに留意されたい。例えば、図13に図示したケースにおいて、回復のために、特定のデータストリームのストリームクロック F_{stream_clk} は、単純に、スタッフィングシンボル及びストリームデータシンボルの合計数（P）と比較したストリームデータシンボルの数（M）を確定することで決定できる。更に詳しくは、ストリームクロック F_{stream_clk} は、次のように決定される。

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

ここで、M及びPは、レシーバ204によって測定できる。

【0049】

図17は、本発明の実施形態による、多数のデータストリームをスケジュールするプロセス1700を詳細に表すフローチャートを図示している。プロセス1700は、1702で、トランスミッタがストリーム属性データをレシーバに送信することで開始される。説明する実施形態において、属性データは、補助チャネルを経由して送信される。次に、1704において、レシーバは、着信データストリームを受領及び処理するために（利用可能な場合）十分なリソースを割り当てる。トランスミッタがデータストリーム1706を受領した後、ストリームトランスポートスケジューラは、1708において、リンクデータストリームを形成し、一方、1710において、リンクデータストリームは、メインリンクを経由して、トランスミッタによりレシーバへ送信される。次に、1712において、追加データストリームが追加され、1714において、レシーバが追加対象のデータストリームの受領及び処理の両方を行える場合、1716において、新しいデータストリームは、以前に伝送されたデータストリームのデータパケットサイズが一定に維持されるという点でトランスペアレントに追加される。

【0050】

図18は、本発明の実施形態による、リンクデータストリームを形成するプロセス1800を詳細に表すフローチャートを図示している。処理1800は、処理1708の形成動作1708の特定の実施であることに留意されたい。したがって、トランスポートスケジューラは、1802において、メインリンクのリンクビットレートを決定し、1804において、トランスミッタからレシーバへのリンクを経由してトランスポートされるべきデータストリームのそれぞれについて、ストリームビットレートを決定する。次に1806において、それぞれのデータストリームのためのパケットサイズが、ストリームビットレートと、リンクビットレートと、所定の最大パケットサイズとに基づいて決定される。次に1808において、それぞれのデータストリームのためのデータパケットが、それぞれ多数のデータシンボルとスタッフィングシンボルとを含む状態で形成される。データパケットが形成されると、単一のトランスポートスケジューラサイクル時間中に、トランスポートスケジューラは、1810において、レシーバに送信されるべきデータストリームのそれぞれからのデータパケットを（時分割多重化を使用して）連結し、1812において、連結されたデータパケットにレーン間アライメントパケットを追加する。1814において、トランスポートスケジューラは、連結されたデータパケットをレシーバに伝送するために、トランスミッタに信号を送り、1816において決定されるような伝送の中止まで、1808～1814を繰り返す。

【0051】

図19は、本発明の実施形態による、システム200の論理層構造1900を例示している。正確な実施は用途に応じて変化してよいが、一般に、ソース（ビデオソース202等）は、トランスミッタハードウェアを含むソース物理層1902と、多重化ハードウェア

10

20

30

40

50

ア及びステートマシン（又はファームウェア）を含むソースリンク層 1904 と、オーディオ/ビジュアル/グラフィックスハードウェア及び関連ソフトウェアのようなデータストリームソース 1906 とにより形成されることに留意されたい。同様に、ディスプレイデバイスは、物理層 1908（様々なレシーバハードウェアを含む）と、逆多重化ハードウェア及びステートマシン（又はファームウェア）を含むシンクリンク層 1910 と、ディスプレイ/タイミングコントローラハードウェア及びオプションのファームウェアを含むストリームシンク 1912 とを含む。ソースアプリケーションプロファイル層 1914 は、ソースがリンク層 1904 と通信する形式を定義し、同様に、シンクアプリケーションプロファイル層 1916 は、シンク 1912 がシンクリンク層 1910 と通信する形式を定義する。

10

【0052】

次に、様々な層について、更に詳細に説明する。

【0053】

ソースデバイス物理層：

説明する実施形態において、ソースデバイス物理層 1902 は、電気副層 1902 - 1 と論理副層 1902 - 2 とを含む。電気副層 1902 - 1 は、ホットプラグ/アンプラグ検出回路と、ドライバ/レシーバ/終端レジスタと、パラレル-シリアル/シリアル-パラレル変換と、スペクトラム拡散対応 PLL とのような、インタフェース初期化/動作のための全回路を含む。論理副層 1902 - 2 は、パケット化/逆パケット化と、データスクランブル化/スクランブル解除と、リンクトレーニング用パターン生成と、時間基準回復回路と、メインリンク 222 用の（図 13 に例が図示された）256 リンクデータ文字及び 12 の制御文字を提供する 8B/10B（ANSI X3.230-1994 第 11 条において規定される）及び補助チャンネル 224 用の Manchester II（図 21 参照）のようなデータエンコード/デコードとのための回路を提供する。

20

【0054】

8B/10B エンコードアルゴリズムは、例えば、米国特許第 4,486,739 号において説明されており、参照により本明細書に組み込まれることに留意されたい。当業者に知られている通り、8B/10B コードは、シリアル伝送のために 8 ビットデータブロックを 10 ビットコードワードにエンコードするブロックコードである。加えて、8B/10B 伝送コードは、1 及び 0 のバイト幅ストリームを、最大ランレングスが 5 である 1 及び 0 の DC バランスストリームに変換する。こうしたコードは、トランシーバ 110 等のレシーバによる信頼性の高いクロック回復を可能にするのに十分な信号遷移を提供する。更に、DC バランスデータストリームは、光ファイバ及び電磁線接続にとって有利となる。シリアルストリームにおける 1 及び 0 の平均数は、均等又は均等に近いレベルに維持される。8B/10B 伝送コードは、1 及び 0 の数の格差を、6 及び 4 ビットブロック境界に渡って、-2、0、又は 2 となるように抑制する。このコード方式は、更に、コマンドコードと呼ばれる、信号送信用の追加コードを実施する。

30

【0055】

非圧縮ディスプレイデータが示す反復ビットパターンを回避する（したがって、EMI を低減する）ために、メインリンク 222 上で伝送されるデータは、8B/10B エンコードの前に、まずスクランブル化されることに留意されたい。トレーニングパケットと特殊文字とを除き、全データがスクランブル化されることになる。スクランブル化機能は、リニアフィードバックシフトレジスタ（LFSR）により実施される。データ暗号化が有効である時、LFSR シードの初期値は、暗号鍵セットに応じて決まる。暗号化のないスクランブル化の場合、初期値は固定されることになる。

40

【0056】

データストリーム属性は補助チャンネル 224 上で伝送されるため、メインリンクパケットヘッダは、ストリーム識別番号の役割を果たし、これにより、オーバヘッドを大幅に減らし、リンク帯域幅を最大化する。更に、メインリンク 222 と補助リンク 224 とは、どちらも別個のクロック信号線を有していないことに留意されたい。これにより、メイン

50

リンク 2 2 2 及び補助リンク 2 2 4 上のレシーバは、データをサンプリングし、着信データストリームからクロックを抽出する。レシーバ電気副層内の任意の位相同期ループ (PLL) 回路のための高速位同期は、補助チャンネル 2 2 4 が半二重双方向であり、トラフィックの方向が頻繁に変化するため、重要となる。したがって、補助チャンネルレシーバの PLL は、Manchester II (MII) コードの頻繁かつ均一な信号遷移の結果、僅か 1 6 データ期間で位同期を行う。

【 0 0 5 7 】

リンク設定時、メインリンク 2 2 2 のデータレートは、補助チャンネル 2 2 4 上でのハンドシェイクを使用して交渉される。このプロセス中には、トレーニングパケットの既知のセットが、メインリンク 2 2 2 上において、最高リンク速度で送信される。成功又は失敗は、補助チャンネル 2 2 4 を介して、トランスミッタ 1 0 2 に連絡を返す。トレーニングが失敗した場合、メインリンク速度を低減し、成功するまでトレーニングを繰り返す。これにより、ソース物理層 1 9 0 2 は、ケーブルのトラブルに対する抵抗力が更に増し、したがって、モニタアプリケーションの外部ホストとして更に適したものとなる。しかしながら、従来のディスプレイインタフェースとは異なり、メインチャンネルのリンクデータレートは、ピクセルクロックレートから切り離される。リンクデータレートは、リンク帯域幅が伝送ストリームの総帯域幅を上回るように設定される。

10

【 0 0 5 8 】

ソースデバイスリンク層：

ソースリンク層 1 9 0 4 は、リンクの初期化及び管理を扱う。例えば、モニタの電源投入時、或いはソース物理層 1 9 0 2 からのモニタケーブルの接続時に生成されるホットプラグ検出イベントを受領すると、ソースデバイスリンク層 1 9 0 4 は、補助チャンネル 2 2 4 上でのやり取りを介して、レシーバの能力を評価し、トレーニングセッションで決定されるような最大メインリンクデータレートと、レシーバ上の時間基準回復ユニットの数と、両側で利用可能なバッファサイズと、USB 拡張機能の可用性とを決定し、その後、ストリームソース 1 9 0 6 に、関連するホットプラグイベントを通知する。加えて、ストリームソース 1 9 0 6 からの要求時、ソースリンク層 1 9 0 4 は、表示能力 (EDID 又は同等のもの) を読み出す。正常動作中、ソースリンク層 1 9 0 4 は、補助チャンネル 2 2 4 を介してレシーバ 1 0 4 にストリーム属性を送信し、メインリンク 2 2 2 が要求データストリームを処理するのに十分なリソースを有するかをストリームソース 1 9 0 4 に通知し、同期消失及びバッファオーバーフローといったリンク障害イベントをストリームソース 1 9 0 4 に通知し、ストリームソース 1 9 0 4 によって提出された MCCS コマンドをレシーバに対して、補助チャンネル 2 2 4 を介して送信する。ソースリンク層 1 9 0 4 とストリームソース / シンクとの間の全ての通信では、アプリケーションプロファイル層 1 9 1 4 において定義された形式を使用する。

20

30

【 0 0 5 9 】

アプリケーションプロファイル層 (ソース及びシンク)：

一般に、アプリケーションプロファイル層は、ストリームソース (又はシンク) が関連するリンク層とインタフェースする際の形式を定義する。アプリケーションプロファイル層によって定義された形式は、以下のカテゴリ、即ち、アプリケーション独立形式 (リンク状態問い合わせのためのリンクメッセージ) 及びアプリケーション従属形式 (メインリンクデータマッピング、レシーバのための時間基準回復数式、及び該当する場合のシンク能力 / ストリーム属性メッセージサブパケット形式) に分類される。アプリケーションプロファイル層は、以下の色形式、即ち、2 4 ビット RGB、1 6 ビット RG 2 5 6 5、1 8 ビット RGB、3 0 ビット RGB、2 5 6 色 RGB (CLUT に基づく)、1 6 ビット C b C r 4 2 2、2 0 ビット Y C b C r 4 2 2、及び 2 4 ビット Y C b C r 4 4 4 をサポートする。

40

【 0 0 6 0 】

例えば、ディスプレイデバイスのアプリケーションプロファイル層 (APL) 1 9 1 4 は、本質的には、インタフェース 1 0 0 に送信される又はインタフェース 1 0 0 から受領

50

されるデータの提示形式を含む、メインリンク 2 2 2 上でのストリームソース/シンク通信のための形式を記述するアプリケーションプログラミングインタフェース (API) である。APL 1 9 1 4 の一部の態様 (電力管理コマンド形式等) は、基本モニタ機能であるため、インタフェース 1 0 0 のあらゆる使用方法において共通である。データマッピング形式及びストリーム属性形式といった、その他の非基本モニタ機能は、アプリケーション、或いは伝送対象である等時性ストリームのタイプに固有のものとなる。アプリケーションに関係なく、ストリームソース 1 9 0 4 は、ソースリンクレイヤ 1 9 1 4 に問い合わせを行い、メインリンク 2 2 2 上で何らかのパケットストリーム伝送を開始する前に、メインリンク 2 2 2 が (複数の) 保留中のデータストリームを処理できるかを確認する。

【0061】

メインリンク 2 2 2 が (複数の) 保留中のデータストリームを処理できると判断された時、ストリームソース 1 9 9 6 は、ソースリンク層 1 9 1 4 にストリーム属性を送信し、その後、ストリーム属性は補助チャンネル 2 2 4 上でレシーバに伝送される。こうした属性は、特定のストリームのパケットを識別し、ストリームからオリジナルのデータを復元し、ストリームのネイティブデータレートにフォーマットし直すために、レシーバによって使用される情報である。データストリームの属性は、アプリケーション従属である。

【0062】

望ましい帯域幅がメインリンク 2 2 2 上で利用できない場合には、ストリームソース 1 9 1 4 は、例えば、画像リフレッシュレート又は色深度を低減することで、修正措置を講じてよい。

【0063】

ディスプレイデバイス物理層：

ディスプレイデバイス物理層 1 9 1 6 は、ディスプレイデバイスリンク層 1 9 1 0 とディスプレイデバイス APL 1 9 1 6 とを、リンクデータの送受信に使用される信号伝送技術から分離させる。メインリンク 2 2 2 及び補助チャンネル 2 2 4 は、独自の物理層を有し、それぞれの物理層は、論理副層とコネクタ仕様を含む電気副層とで構成される。例えば、半二重双方向補助チャンネル 2 2 4 は、図 2 2 に示すように、リンクの各端部にトランスミッタとレシーバとの両方を有する。補助リンクトランスミッタ 2 9 0 2 には、論理副層 1 9 0 8 - 1 によってリンク文字が提供され、その後、リンク文字はシリアル化され、対応する補助リンクレシーバ 2 9 0 4 に伝送される。レシーバ 2 9 0 4 は、次に、補助リンク 2 2 4 からのシリアル化されたリンク文字を受領し、リンク文字クロックレートでデータを非シリアル化する。ソース論理副層の主要な機能は、信号エンコードと、パケット化と、(EMI 低減のための) データスクランブル化と、トランスミッタポート用のトレーニングパターン生成とを含むことに留意されたい。一方、レシーバポートについて、レシーバ論理副層の主要な機能は、信号デコードと、非パケット化と、データスクランブル解除と、時間基準回復とを含む。

【0064】

補助チャンネル：

補助チャンネル論理副層のような機能は、データエンコード/デコードと、データのフレーム化/非フレーム化とを含み、補助チャンネルプロトコルには二種類のオプションが存在し、スタンドアロンプロトコル (ポイントツーポイントトポロジにおけるリンク設定/管理機能に限定される) は、リンク層ステートマシン又はファームウェアによって管理可能な軽量プロトコルと、USB トラフィック等の他のデータタイプ及びデジタイズ方式でつながれたデバイス等のトポロジをサポートする拡張プロトコルとになる。データエンコード及びデコード方式はプロトコルに関係なく同一だが、データのフレーム化は二種類の間で異なることに留意されたい。

【0065】

更に図 2 9 を参照すると、補助チャンネル電気副層は、トランスミッタ 2 9 0 2 及びレシーバ 2 9 0 4 を含む。トランスミッタ 2 9 0 2 には、論理副層によってリンク文字が提供され、リンク文字はシリアル化され外部に伝送される。レシーバ 2 9 0 4 は、シリアル化

10

20

30

40

50

されたリンク文字をリンク層から受領し、その後、リンク文字クロックレートで非シリアル化する。補助チャンネル 224 の正及び負の信号は、図示したように、リンクの各端部で、50 オーム終端抵抗を介してアースさせ終端とする。説明する実施において、ドライバ電流は、リンク条件に応じてプログラム可能であり、約 8 mA ~ 約 24 mA の範囲となり、結果として、約 400 mV ~ 約 1.2 V の $V_{\text{differential_pp}}$ の範囲が生じる。電気的アイドルモードでは、正及び負の信号は、どちらも駆動されない。電気的アイドル状態から伝送を開始する時には、SYNC パターンを伝送し、リンクを再び確立する必要がある。説明する実施形態において、SYNC パターンは、補助チャンネル差動ペア信号をクロックレートで 28 回トグルさせ、Manchester II コードの四つの 1 を続けることで構成される。ソースデバイスの補助チャンネルマスタは、補助チャンネル 224 の正及び負の信号を周期的に駆動及び測定することで、ホットプラグ及びホットアンプラグイベントを検出する。

【0066】

メインリンク：

説明する実施形態において、メインリンク 222 は、ローカル水晶周波数の整数の倍数である個別の変換リンクレートをサポートする (24 MHz のローカル水晶周波数を有するリンクレート定数の代表的なセットについては図 3 を参照)。図 16 に図示したように、メインリンク 222 (単方向チャンネル) は、ソースデバイスにおいてトランスミッタ 1602 のみを有し、ディスプレイデバイスにおいてレシーバ 1604 のみを有する。

【0067】

図示したように、ケーブル 2304 は、一組のツイストペア線を含む形態をとり、通常の RGB カラーに基づくビデオシステム (PAL に基づくテレビシステム等) において提供される赤 (R)、緑 (G)、及び青 (B) のそれぞれについて一本となる、当業者に知られている通り、ツイストペアケーブルは、独立して絶縁して互いに撚り合わせた二本の線で構成されるケーブルのタイプである。一方の線は信号を伝送し、他方の線はアースされ、信号の干渉を吸収する。他の一部のシステムにおいて、信号は、NTSC ビデオテレビシステムで使用されるコンポーネントに基づく信号 (Pb、Pr、Y) にもできることに留意されたい。ケーブル内で、各ツイストペアは、個別にシールドされる。+12 V の電力及び接地用の二本のピンが提供される。各差動ペアの特性インピーダンスは、100 オーム \pm 20% である。ケーブル全体もシールドされる。この外部シールド及び個別シールドは、両方の端部でコネクタシェルに短絡させる。コネクタシェルは、ソースデバイスにおいて、アースに短絡させる。図 24 に図示したコネクタ 2400 は、13 ピンを一列で有し、ソースデバイス端部のコネクタとディスプレイデバイス端部のコネクタとの両方で同一なピンアウトを有する。ソースデバイスは、電力を供給する。

【0068】

メインリンク 222 は、両方の端部で終端され、メインリンク 222 が AC 結合であることから、終端電圧は 0 V (アース) ~ +3.6 V のいずれかにすることができる。説明する実施において、駆動電流は、リンク条件に応じてプログラム可能であり、約 8 mA ~ 約 24 mA の範囲となり、結果として、約 400 mV ~ 約 1.2 V の $V_{\text{differential_pp}}$ の範囲が生じる。最小電圧振幅は、トレーニングパターンを使用して、各接続について選択される。電気的アイドル状態は、電力管理モードを提供する。電気的アイドルにおいて、正及び負の信号は、どちらも駆動されない。電気的アイドル状態から伝送を開始する時、トランスミッタは、レシーバとのリンクを再確立するために、トレーニングセッションを実行する必要がある。

【0069】

状態図：

次に、本発明を、下で説明する図 25 及び 26 に図示した状態図の観点から説明する。したがって、図 25 は、下で説明する状態図を図示している。オフ状態 2502 において、システムはオフとなり、ソースが無効化されるようになる。ソースが有効となった場合、システムは、省電力及びレシーバ検出に適した待機状態 2504 へ遷移する。レシーバが存在するかどうか (即ち、ホットプラグ/プレイ) を検出するために、補助チャンネルに

は周期的にパルスが送られ(10ms毎に1us等)、駆動中の終端レジスタに渡る電圧降下の度合いを測定する。測定された電圧降下に基づいて、レシーバが存在すると判断された場合、システムは、レシーバが検出されたこと、即ち、ホットプラグイベントが検出されたことを示す検出レシーバ状態2506に遷移する。しかしながら、レシーバが検出されなかった場合、レシーバ検出は、発生する場合はレシーバが検出される時点まで、或いはタイムアウトが経過した時点まで、継続される。一部のケースにおいて、ソースデバイスは、更なるディスプレイ検出が試行されない「オフ」状態への移行を選択してもよいことに留意されたい。

【0070】

状態2506において、ディスプレイのホットアンプラグイベントが検出された場合、システムは、再び待機状態2504に遷移する。そうでない場合、ソースは、正及び負の信号により補助チャンネルを駆動し、レシーバのスリープ解除を行い、存在する場合は、レシーバのその後の応答をチェックする。応答が受領されない場合、レシーバはスリープ解除されておらず、ソースは、状態2506にとどまる。しかしながら、ディスプレイから信号が受領された場合、ディスプレイはスリープ解除されており、ソースでは、レシーバのリンク能力(最大リンクレート、バッファサイズ、及び時間基準回復ユニット数等)を読み出す準備が整い、システムは、メインリンク初期化状態2508に遷移し、トレーニング開始通知段階を開始する準備が整う。

【0071】

このとき、トレーニングセッションは、メインリンクにおいて、設定されたリンクレートでトレーニングパターンを送信することで開始され、関連するトレーニング状態をチェックする。レシーバは、三段階のそれぞれについて成功/失敗ビットを設定し、トランスミッタは、成功のみの検出によって次の段階に進み、成功が検出される時、メインリンクはそのリンクレートで準備されるようになる。このとき、インタフェースは、正常動作状態2510に遷移し、そうでない場合には、リンクレートが低減され、トレーニングセッションが繰り返される。正常動作状態2510の間、ソースは、リンク状態のインデックスを周期的にモニタし続け、失敗した場合には、ホットアンプラグイベントが検出され、システムは待機状態2504に遷移し、ホットプラグ検出イベントを待つ。しかしながら、同期消失が検出された場合、システムは、メインリンク再始動イベントのために状態2508に遷移する。

【0072】

図26は、下で説明するディスプレイ状態図2600を図示している。状態2602では、電圧は検出されず、ディスプレイは、オフ状態へ進む。待機モード状態2604では、メインリンクレシーバ及び補助チャンネルスレーブは、両方とも、電氣的にアイドルとなり、補助チャンネルスレーブポートの終端レジスタに渡る電圧降下が、所定の電圧についてモニタされる。所定の電圧が検出された場合、補助チャンネルスレーブポートは、オンになってホットプラグイベントを示し、システムは表示状態2606へ移行し、そうでない場合、ディスプレイは待機状態2604にとどまる。状態2606(メインリンク初期化段階)において、ディスプレイが検出された場合、補助スレーブポートは、完全にオンとなり、トランスミッタは、レシーバリリンク能力読み出しコマンドに応答し、ディスプレイ状態は2608に遷移し、そうでない場合には、所定の期間を超えて補助チャンネルでの活動が存在しなければ、補助チャンネルスレーブポートは、待機状態2604に入る。

【0073】

トレーニング開始通知段階の間、ディスプレイは、トレーニングパターンを使用してイコライザを調整し、各段階の結果を更新することで、トランスミッタによるトレーニング始動に応答する。トレーニングが失敗した場合には、別のトレーニングセッションを待ち、トレーニングが成功した場合には、正常動作状態2610へ移行する。所定の時間(例えば、10ms)を超えて補助チャンネル又は(トレーニングのための)メインリンクでの活動が存在しない場合、補助チャンネルスレーブポートは、待機状態2604に設定される。

10

20

30

40

50

【 0 0 7 4 】

図 2 7 ~ 3 1 は、クロスプラットフォームインタフェースの特定の実装を図示している。

【 0 0 7 5 】

図 2 7 は、本発明によるトランスミッタ 2 7 0 4 を組み込んだオンボードグラフィックスエンジン 2 0 0 2 を有する P C マザーボード 2 0 0 を図示している。トランスミッタ 2 7 0 4 は、図 1 に図示したトランスミッタ 1 0 2 の特定の例であることに留意されたい。説明する実施形態において、トランスミッタ 2 7 0 4 は、マザーボード 2 0 0 0 にマウントされた（コネクタ 2 4 0 0 の線に沿った）コネクタ 2 7 0 6 に結合され、次にコネクタ 2 7 0 6 は、ディスプレイデバイス 2 7 1 0 を結合するツイストペアケーブル 2 7 1 0 を

10

【 0 0 7 6 】

この技術で知られているように、PCI Express（カリフォルニア州サンタクララの Intel Corporationが開発）は、広帯域幅で低ピン数のシリアル相互接続技術であり、既存の P C I インフラとのソフトウェア互換性も維持する。この構成において、PCI Expressポートは、図示したマザーボードマウントコネクタを使用しても、ディスプレイデバイスを直接的に駆動できるクロスプラットフォームインタフェースの要件に準拠するように増強される。

【 0 0 7 7 】

マザーボードにコネクタをマウントするのが実際的ではない状況において、信号は、P C I Expressマザーボードの S D V O スロットを通じてルートを定め、図 2 8 に図示したような受動カードコネクタを使用して、P C の背面に伝えることができる。現世代のアドイングラフィックスカードと同様に、アドイングラフィックスカードは、図 3 0 に図示したようなオンボードグラフィックスエンジンに取って代わることができる。

20

【 0 0 7 8 】

ノートブックでの応用の場合、マザーボードグラフィックスエンジン上のトランスミッタにより、パネルを直接的に駆動する一体式レシーバ / T C O N が、内部配線を通じて駆動される。最も費用効率に優れた実施では、レシーバ / T C O N は、パネル上にマウントされ、これにより、相互接続線の数は、図 3 1 に図示したように八本又は十本に低減される。

30

【 0 0 7 9 】

上の全ての例では、一体式トランスミッタを想定している。しかしながら、P C I 及び P C I Express環境に、それぞれ A G P 又は S D V O スロットを通じて統合される、スタンドアロントランスミッタとしても、完全に実現可能である。スタンドアロントランスミッタは、グラフィックスハードウェア又はソフトウェアを全く変化させることなく、出力ストリームを可能にする。

【 0 0 8 0 】

フローチャートの実施形態：

次に、本発明の方法について、本発明を可能にする特定のプロセスをそれぞれが説明する多数のフローチャートの観点から説明する。具体的には、図 3 2 ~ 3 6 は、単独で或いは任意の組み合わせで使用される時に本発明の態様を説明する、多数の相互に関連するプロセスを表している。

40

【 0 0 8 1 】

図 3 2 は、本発明の実施形態による、インタフェース 1 0 0 の動作モードを決定するプロセス 3 2 0 0 を詳細に表すフローチャートを図示している。このプロセスにおいて、動作モードは、ビデオソース及びディスプレイデバイスが共にデジタルである場合、デジタルモードのみに設定されることになる。そうでない場合、動作モードは、アナログモードに設定されることになる。このコンテキストにおける「アナログモード」は、従来の V G A モードと、埋め込みアライメント信号及び双方向側波帯を備えた差動アナログビデオを有する拡張アナログモードとの両方を含むことに留意されたい。この拡張アナログモード

50

については、下で説明する。

【0082】

ステップ3202において、ビデオソースは、ビデオソースがアナログ又はデジタルデータをサポートするかを判断するために問い合わせを受ける。ビデオソースがアナログデータのみをサポートする場合、結合デバイス100の動作モードは、アナログに設定され（ステップ3208）、プロセスは終了することになる（ステップ3212）。

【0083】

ビデオソースがデジタルデータを出力できる場合、プロセスは、ステップ3206へ続く。ディスプレイデバイスは、次に、ディスプレイデバイスがデジタルデータを受領するように構成されているかを判断するために問い合わせを受ける。ディスプレイデバイスがアナログデータのみをサポートする場合、結合デバイス100の動作モードは、アナログに設定され（ステップ3208）、プロセスは終了することになる（ステップ3212）。そうでない場合、結合デバイス100の動作モードは、デジタルに設定される（ステップ3210）。例えば、プロセッサは、モードをデジタルに設定するために、結合デバイス内のスイッチを制御してよい。一般に、結合デバイスは、ビデオソース及びビデオシンクが対応するデジタルモードで動作する時のみ、完全デジタルモードで動作するように構成される。

10

【0084】

図33は、本発明の一部の態様による、リアルタイムビデオ画質チェックを提供するプロセス3300を詳細に表すフローチャートを図示している。この例において、プロセス3300の全ての判定は、ディスプレイインタフェースに結合されたプロセッサにより行われる。

20

【0085】

ステップ3300において、ビデオ信号が、ビデオソースから受領される。次に、信号品質テストパターンが、受領ビデオ信号に関連するビデオソースによって提供される（ステップ3302）。ステップ3304において、品質テストパターンに基づいて、ビット誤り率が判定される。その後、ビット誤り率が閾値より大きいかについて判定される（ステップ3306）。ビット誤り率が閾値より大きくないと判定された場合には、ビデオフレームが更に存在するかどうかについて判定される（ステップ3314）。ビデオフレームが更に存在しないと判定された場合には、プロセスは、ステップ3300に戻る。そうでない場合、プロセスは終了する。

30

【0086】

しかしながら、ステップ3306において、ビット誤り率が閾値より大きいと判定された場合、ビットレートが最小ビットレートより大きいかについての判定が行われる（ステップ3308）。ビットレートが最小ビットレートより大きい場合、ビットレートを低下させ（ステップ3310）、プロセスはステップ3306に戻る。ビットレートが最小ビットレートより大きくない場合は、モードをアナログモードに変更し（ステップ3312）、プロセスを終了する。

【0087】

図34は、本発明の実施形態による、リンク設定プロセス3400のフローチャートを図示している。プロセス3400は、3402において、ホットプラグ検出イベント通知の受領によって開始される。3404において、最大データレートと、レシーバに含まれる時間基準回復ユニット数と、利用可能なバッファサイズとを決定するために、関連する補助チャネルを経由して、メインリンクの問い合わせが行われる。次に、3406において、最大リンクデータレートが、トレーニングセッションを経由して検証され、3408において、データストリームソースは、ホットプラグイベントを通知される。3410において、ディスプレイの能力が、補助チャネルを経由して（例えば、E D I Dを使用して）決定され、ディスプレイは、3412において、問い合わせに回答し、次に、3414において、メインリンクトレーニングセッションのための協働が生じる。

40

【0088】

50

次に、3416において、ストリームソースは、補助チャネルを経由して、レシーバにストリーム属性を送信し、3418において、ストリームソースは、メインリンクが3420での要求データストリーム数をサポートできるかを更に通知される。3422において、様々なデータパケットが、関連するパケットヘッダを追加することで形成され、3424において、多数のソースストリームの多重化がスケジュールされる。3426において、リンク状態がOKかどうかについて、判定が行われる。リンク状態がOKではない場合、(複数の)ソースは、3428においてリンク障害イベントの通知を受け、そうでない場合、リンクデータストリームは、3430において様々なパケットヘッダに基づいて、ネイティブストリームに再構築される。3432において、再構築されたネイティブストリームは、ディスプレイデバイスに渡される。

10

【0089】

図35は、本発明の実施形態による、トレーニングセッションを実行するプロセス3500を詳細に表すフローチャートを図示している。トレーニングセッションプロセス3500は、図32において説明した動作3206の一実施であることに留意されたい。トレーニングセッションは、3502において、設定リンクレートにより、メインリンク上でトレーニングパターンをレシーバに送信することで開始される。通常のリントレーニングパターンは、本発明の実施形態による、通常のリントレーニングパターンは、図11に図示される。例示のように、トレーニングセッション中、段階1は、最短のランレングスを表し、一方、段階2は、最長のものとなる。レシーバは、こうした二つの段階を使用して、イコライザを最適化する。段階3において、ビットロック及び文字ロックは、両方とも、リンク品質が妥当である限り達成される。3504において、レシーバは、関連するトレーニング状態をチェックし、トレーニング状態のチェックに基づいて、レシーバは、3506において、三段階のそれぞれとトランスミッタとについて、成功/失敗ビットを設定する。各段階において、レシーバは、成功のみが検出された時に次の段階に進み、3510において、レシーバが成功を検出しない場合、レシーバは、リンクレートを低減し、トレーニングセッションを繰り返す。メインリンクでは、3512において、成功が検出されたリンクレートでの準備が整う。

20

【0090】

図36は、本発明を実施するために利用されるコンピュータシステム3600を例示している。コンピュータシステム3600は、本発明の実施可能なグラフィックスシステムの例に過ぎない。コンピュータシステム3600は、中央処理装置(CPU)3610と、ランダムアクセスメモリ(RAM)3620と、読み出し専用メモリ(ROM)3625と、一つ以上の周辺機器3630と、グラフィックスコントローラ3660と、一次記憶デバイス3640及び3650と、デジタルディスプレイユニット3670とを含む。この技術でよく知られているように、ROMは、データ及び命令を単方向でCPU3610に転送する役割を果たし、一方、RAMは、通常、データ及び命令を双方向の形で転送するために使用される。CPU3610は、一般に、任意の数のプロセッサを含んでもよい。一次記憶デバイス3640及び3650は、両方とも、任意の適切なコンピュータ読取可能な媒体を含んでもよい。通常は大容量メモリデバイスである二次記憶媒体880も、双方向でCPU3610に結合され、追加的なデータ記憶容量を提供する。大容量メモリデバイス880は、コンピュータコード、データ、及びその他を含む、プログラムを格納するのに使用し得るコンピュータ可読媒体である。通常、大容量メモリデバイス880は、一次記録デバイス3640、3650よりも一般的に低速な、ハードディスク又はテープのような記憶媒体である。大容量メモリ記憶デバイス880は、磁気又は紙テープリダ、或いは他の何らかの広く知られたデバイスの形態をとってもよい。大容量メモリデバイス880内で保持される情報は、適切である場合、標準的な形で、仮想メモリとしてRAM3620の一部に組み込んでもよいと理解される。

30

40

【0091】

CPU3610は、更に、ビデオモニタ、トラックボール、マウス、キーボード、マイク、クロフォン、タッチセンシティブディスプレイ、トランスデューサカードリダ、磁気又

50

は紙テーブリーダ、タブレット、スタイラス、音声又は手書認識器、或いは、当然ながら、他のコンピュータ等、その他の広く知られた入力デバイスといったデバイスを一部として含んでもよい一つ以上の入出力デバイス 890 に結合される。最後に、CPU 3610 は、3695 において一般的に図示したようなネットワーク接続を使用して、コンピュータ、或いはインターネットネットワーク又はイントラネットネットワーク等の電気通信ネットワークに、随意的に結合してもよい。こうしたネットワーク接続により、CPU 3610 は、上記の方法のステップを実行する過程で、ネットワークから情報を受領してよく、或いは、ネットワークに情報を出力してよいと考えられる。こうした情報は、CPU 3610 を使用して実行されるべき命令のシーケンスとして表現される場合が多く、例えば、搬送波内で具現化されるコンピュータデータ信号の形態で、ネットワークから受領され、ネットワークに出力されてよい。上記のデバイス及び材料は、コンピュータハードウェア及びソフトウェアに関わる当業者によく知られている。

【0092】

グラフィックスコントローラ 3660 は、アナログ画像信号と対応する基準信号とを生成し、両方をデジタルディスプレイユニット 3670 に提供する。アナログ画像データは、例えば、CPU 3610 又は外部エンコード（図示なし）から受領したピクセルデータに基づいて生成できる。一実施形態において、アナログ画像データは、RGB 形式で提供され、基準信号は、この技術で広く知られた VSYNC 及び HSYNC 信号を含む。しかしながら、本発明は、その他の形式のアナログ画像、データ、及び/又は基準信号により実施することも可能であると理解するべきである。例えば、アナログ画像データは、対応する時間基準信号を更に備えるビデオ信号データを含むことができる。

【0093】

本発明のいくつかの実施形態のみを説明してきたが、本発明は、本発明の趣旨又は範囲から逸脱することなく他の多数の特定の形態で実施し得ると理解されるべきである。この説明は、制限的ではなく例示的なものと考えられるべきであり、本発明は、本明細書に記載した詳細に限定されず、付記する請求項の範囲内及び等価物の全範囲内で、変形し得る。

【0094】

以上、好適な実施形態の観点から本発明を説明してきたが、本発明の範囲内に含まれる変更、置換、及び等価物が存在する。また、本発明のプロセス及び装置の両方を実現する多数の代替方法が存在することに留意されたい。したがって、本発明は、本発明の本来の趣旨及び範囲内に入るこうした全ての変更、置換、及び等価物を含むものであると解釈されるべきである。

【図面の簡単な説明】

【0095】

【図 1】本発明の実施形態によるクロスプラットフォームディスプレイインタフェース 100 の一般化表現を示す図。

【図 2 A】本発明の多数の実施形態による、ビデオソースとビデオディスプレイユニットとを接続するために使用されるビデオインタフェースシステムを例示する図。

【図 2 B】本発明の多数の実施形態による、ビデオソースとビデオディスプレイユニットとを接続するために使用されるビデオインタフェースシステムを例示する図。

【図 2 C】本発明の多数の実施形態による、ビデオソースとビデオディスプレイユニットとを接続するために使用されるビデオインタフェースシステムを例示する図。

【図 3】本発明の実施形態による、例示的なメインリンクレートを示す図。

【図 4 A】本発明の実施形態による、メインリンクデータパケットを示す図。

【図 4 B】本発明の実施形態による、メインリンクパケットヘッダを示す図。

【図 5 A】本発明の実施形態による、サブパケットエンクロージャと多数のパケットの多重化を提供するように構成されたシステムを示す図。

【図 5 B】図 5 A に図示したシステムの別の実施を示す図。

【図 6】図 5 に図示したストリームの例としての多重化メインリンクストリームの高レベ

ル図。

- 【図 7】本発明の実施形態による、データストリームの別の例を示す図。
- 【図 8】本発明の実施形態による、多重化データストリームの更に別の例を示す図。
- 【図 9 A】本発明の実施形態による、代表的なサブパケットを示す図。
- 【図 9 B】本発明の実施形態による、代表的なメインリンクデータパケットを示す図。
- 【図 10】選択的にリフレッシュされるグラフィックス画像の例を示す図。
- 【図 11】本発明の実施形態による、例示的なリンクトレーニングパターンを示す図。
- 【図 12】本発明の実施形態による、データストリームスケジューラを有する代表的なシステムを示す図。
- 【図 13】図 12 に図示したデータストリームの一部を更に詳細に示す図。 10
- 【図 14】本発明の実施形態による、第四のデータストリームの追加を例示する図。
- 【図 15】本発明の実施形態による、単一非圧縮データストリームの縮退のケースを示す図。
- 【図 16】図 15 に図示した縮退単一データストリームに基づいたビデオ表示を示す図。
- 【図 17】本発明の実施形態による、多数のデータストリームをスケジュールするプロセスを詳細に表すフローチャート。
- 【図 18】本発明の実施形態による、リンクデータストリームを形成するプロセスを詳細に表すフローチャート。
- 【図 19】本発明の実施形態による、システムの論理層状化を例示する図。
- 【図 20】本発明の実施形態による、8 B / 10 B を使用する例示的な特殊文字のマッピングを示す図。 20
- 【図 21】本発明の実施形態による、例示的な Manchester II エンコード方式を示す図。
- 【図 22】本発明の実施形態による、代表的な補助チャネル電気副層を示す図。
- 【図 23】本発明の実施形態による、代表的なメインリンク電気副層を示す図。
- 【図 24】本発明の実施形態による、代表的なコネクタを示す図。
- 【図 25】本発明の実施形態によるソース状態図。
- 【図 26】本発明の実施形態によるディスプレイ状態図。
- 【図 27】本発明の様々なコンピュータに基づく実施を例示する図。
- 【図 28】本発明の様々なコンピュータに基づく実施を例示する図。
- 【図 29】本発明の様々なコンピュータに基づく実施を例示する図。 30
- 【図 30】本発明の様々なコンピュータに基づく実施を例示する図。
- 【図 31】本発明の様々なコンピュータに基づく実施を例示する図。
- 【図 32】本発明の実施形態による、インタフェースの動作モードを決定するプロセスを詳細に表すフローチャート。
- 【図 33】本発明の一部の態様による、リアルタイムビデオ画質チェックを提供するプロセスを詳細に表すフローチャート。
- 【図 34 A】本発明の実施形態による、リンク設定プロセスのフローチャート。
- 【図 34 B】本発明の実施形態による、リンク設定プロセスのフローチャート。
- 【図 35】本発明の実施形態による、トレーニングセッションを実行するプロセスを詳細に表すフローチャート。 40
- 【図 36】本発明を実施するために利用されるコンピュータシステムを例示する図。

【 図 1 】

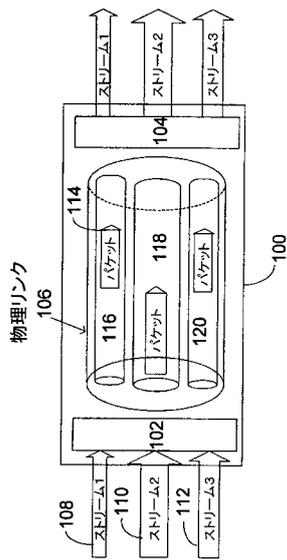


Fig. 1

【 図 2 A 】

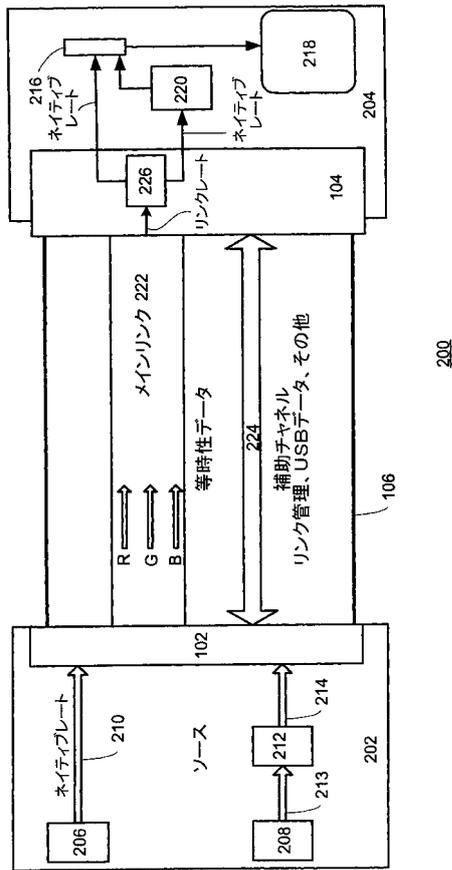


Fig. 2A

【 図 2 B 】

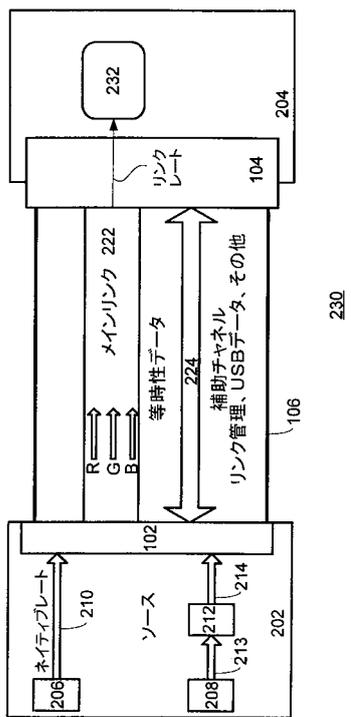


Fig. 2B

【 図 2 C 】

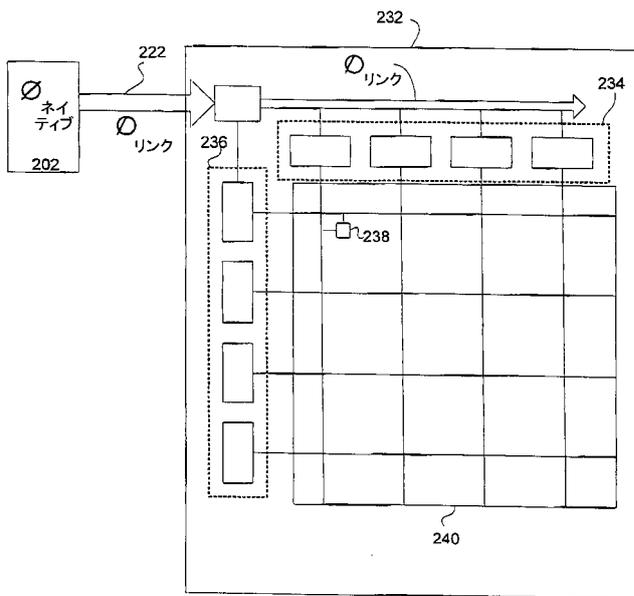


Fig. 2C

【 図 3 】

メインリンクデータレート

チャンネル当たり 公称ポーレート (Gbit/秒)	チャンネル当たり 実ポーレート (Gbit/秒)	24MHz水晶からの クロック増倍率
1.0	0.960	x40
1.35	1.344	x56
1.7	1.728	x72
2.1	2.112	x88
2.5	2.496	x104

Fig. 3

【 図 4 A 】

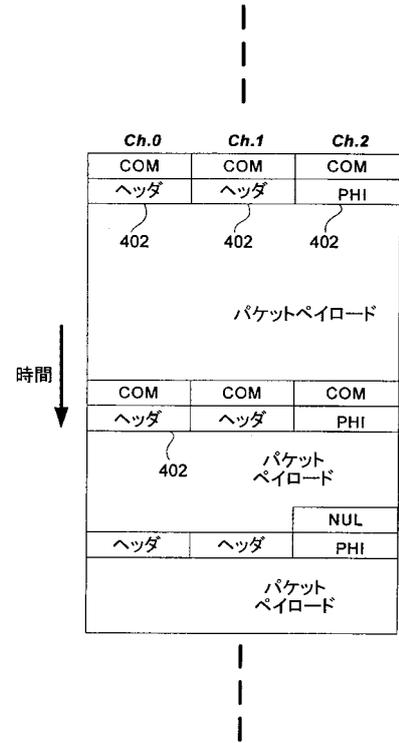


Fig. 4A

メインリンクパケット形式

【 図 4 B 】

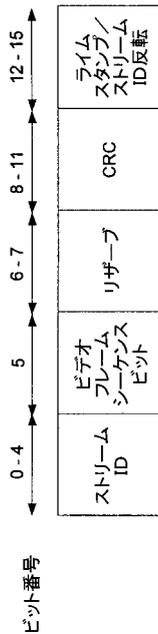


Fig. 4B

【 図 5 A 】

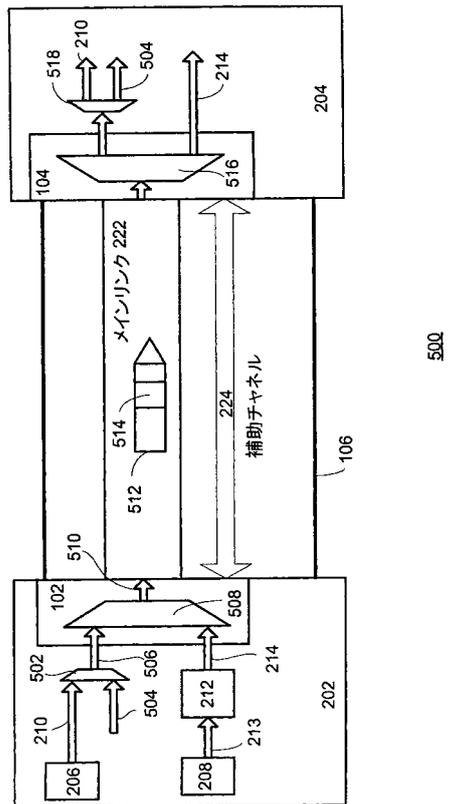
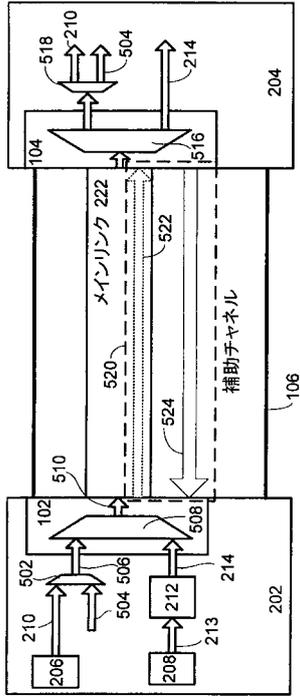


Fig. 5A

【 図 5 B 】



500

Fig. 5B

【 図 6 】

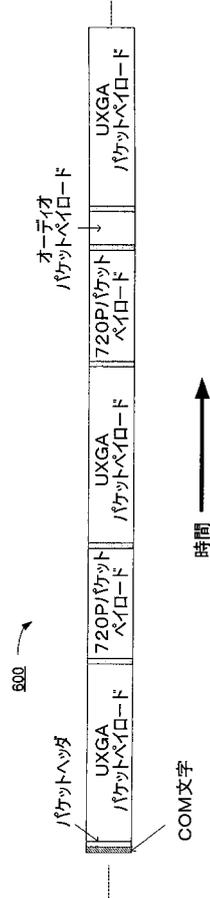


Fig. 6

三本のストリームを備えるリンクフライトパツクの例の高レベル図

【 図 7 】

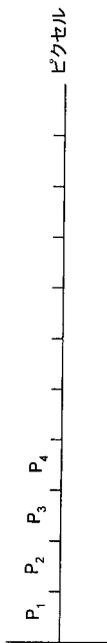


FIG. 7

【 図 8 】

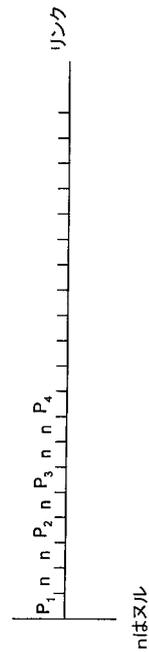
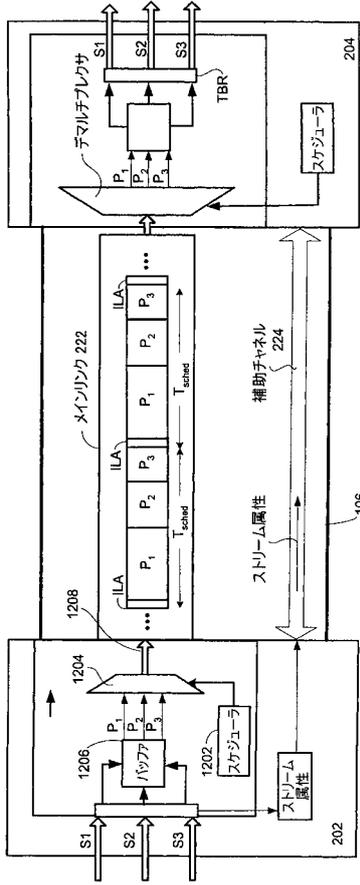


FIG. 8

【 図 1 2 】



1200

Fig. 12

【 図 1 3 】

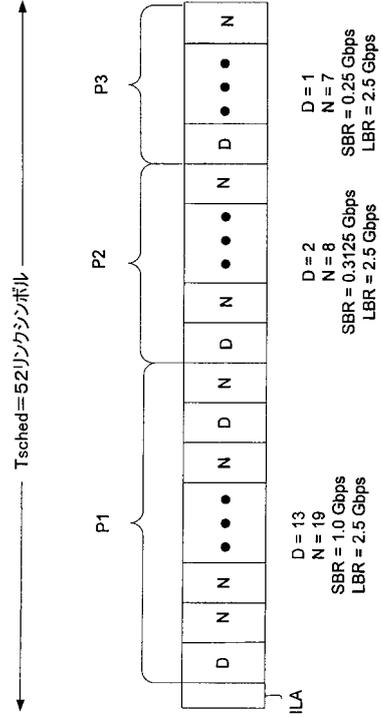


Fig. 13

【 図 1 4 】

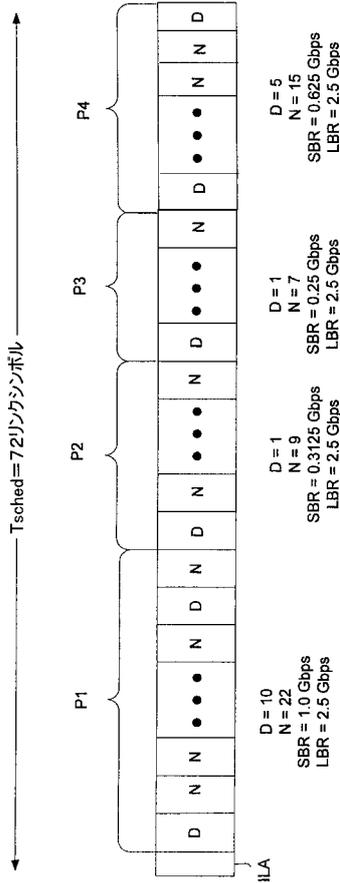


Fig. 14

【 図 1 5 】

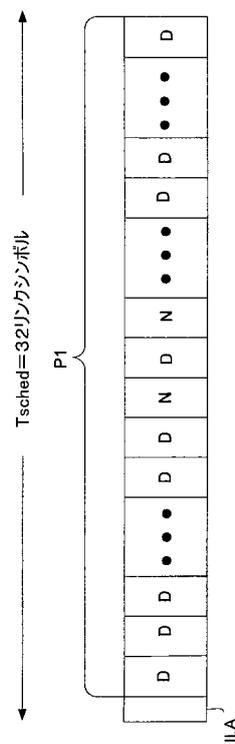


Fig. 15

【図16】

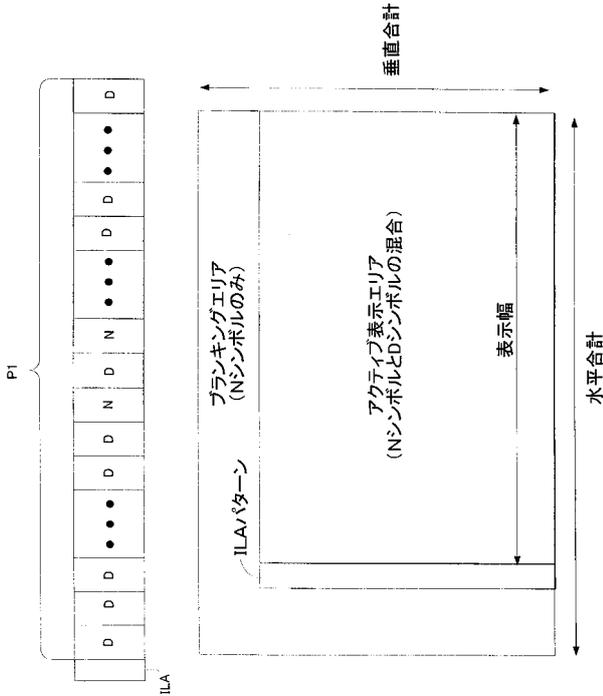


Fig. 16

【図17】

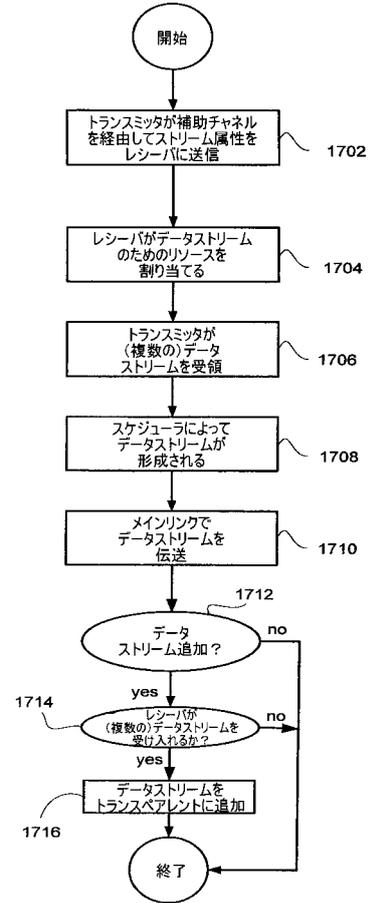


Fig. 17

【図18】

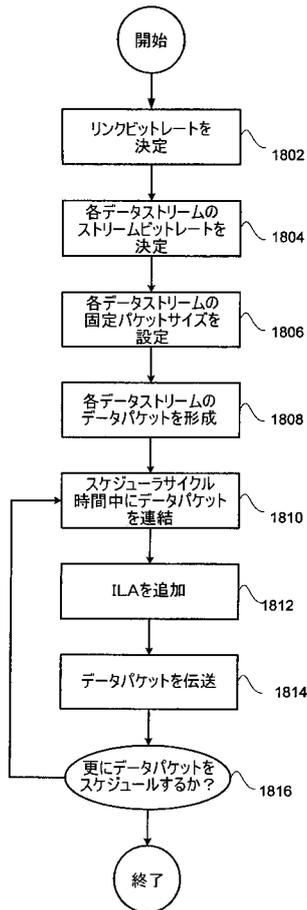


Fig. 18

【図19】

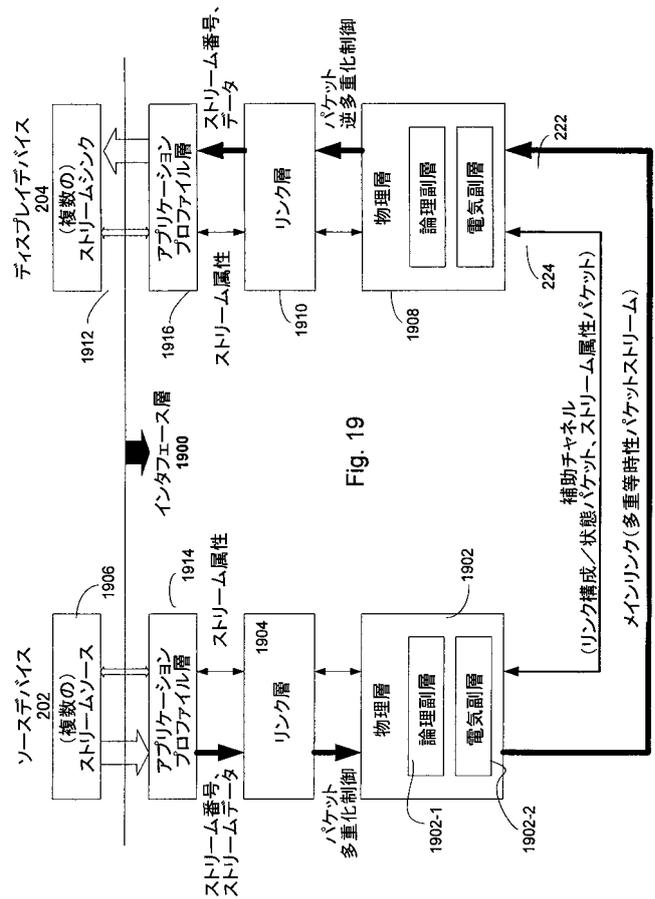


Fig. 19

【 図 2 0 】

8B/10B特殊文字の使用法

エンコード	名称	説明
K28.5	カンマ(COM)	パケット間に挿入される。テストパターンの一部としても使用される。
K28.7	トレーニングパターン(TPN)	トレーニングパターン伝送中にビット/バイトクロックのために送信される。
K23.7	ヌル(NUL)	伝送するデータがない時、パケット期間内に送信される。
K28.2	サブパケット開始(SPS)	パケットに挿入されるサブパケットの開始を示す。
K29.7	サブパケット終了(SPE)	パケットに挿入されるサブパケットの終了を示す。
K28.0	パケットヘッダインジケータ(PHI)	ヘッダの識別のために16ビットのヘッドと共に送信される。
K28.1	リザーブ	
K28.3	リザーブ	
K28.4	リザーブ	
K28.6	リザーブ	
K25.7	リザーブ	
K27.7	リザーブ	

【 図 2 2 】

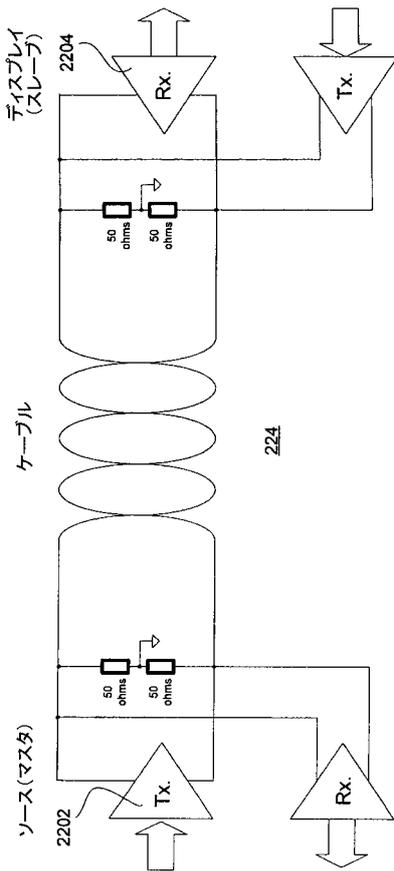


Fig. 22

【 図 2 1 】

Fig. 20

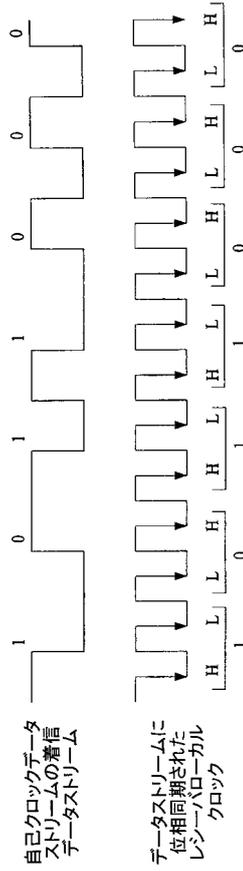


Fig. 21

【 図 2 3 】

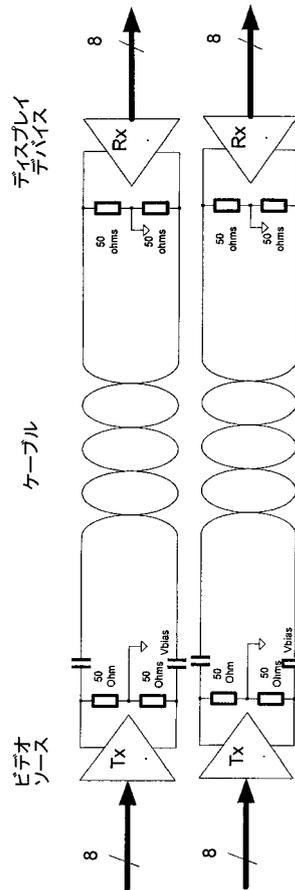


Fig. 23

【 図 2 4 】

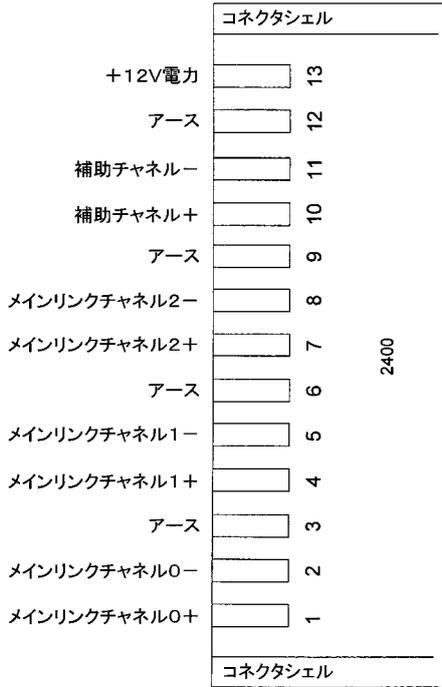


Fig. 24

【 図 2 5 】

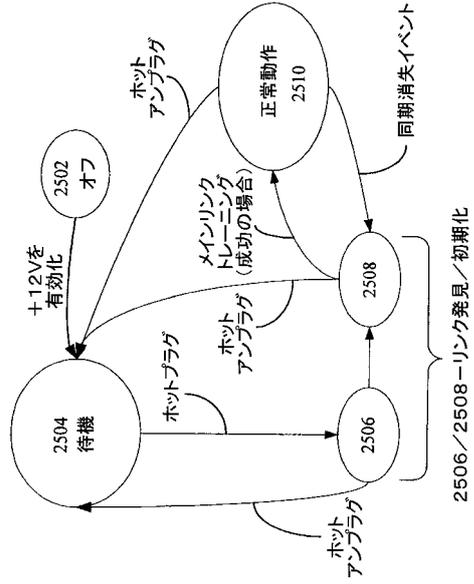


Fig. 25

【 図 2 6 】

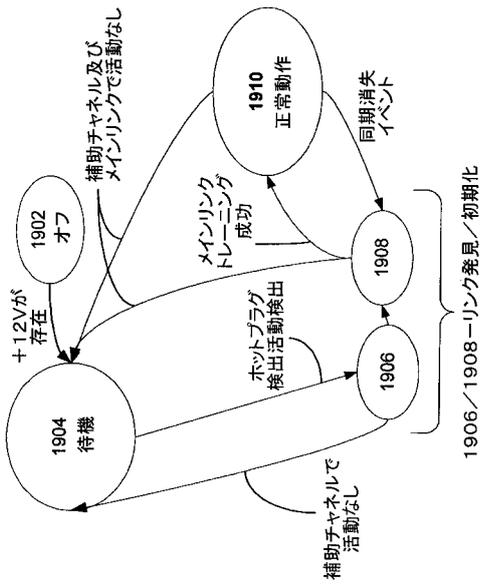
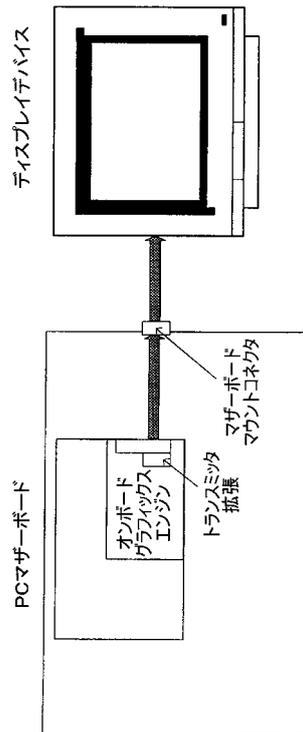


Fig. 26

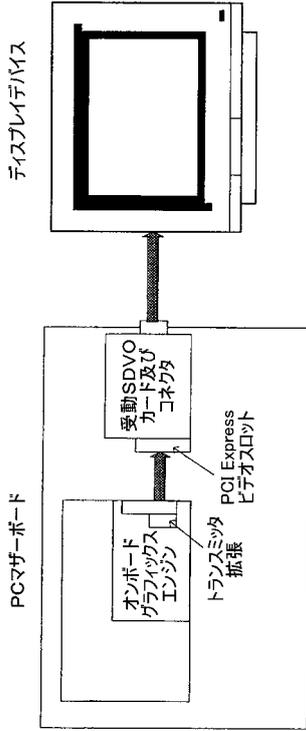
【 図 2 7 】



専用のDD-Expressコネクタを備えた
PCI Expressマザーボード

Fig. 27

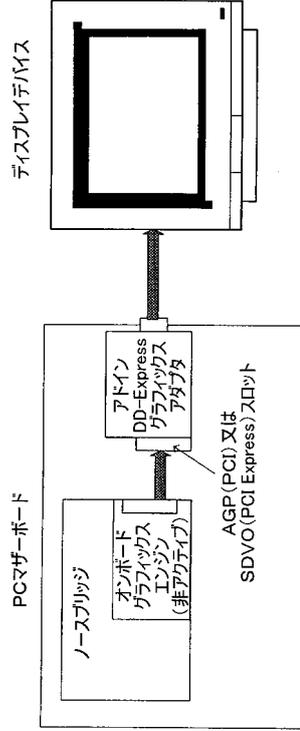
【 図 28 】



受動コネクタカードを備えた
PCI Expressマザーボード

Fig. 28

【 図 29 】



アドインDD-Expressグラフィックスカードを備えた
PCI Expressマザーボード

Fig. 29

【 図 30 】

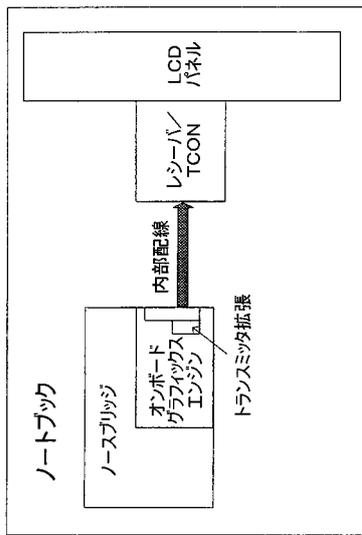
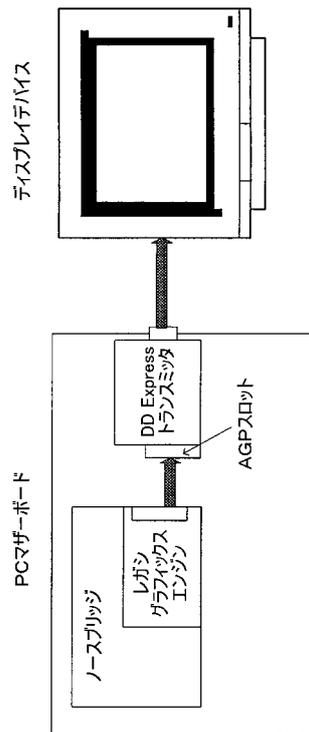


Fig. 30

【 図 31 】



レガシバスカードスロットにマウントされた
レガシグラフィックスアクセラレータ/トランスミッタは、
デジタルラスタデータ/タイミング信号を
メインリンクストリームに変換する

Fig. 31

【 図 3 5 】

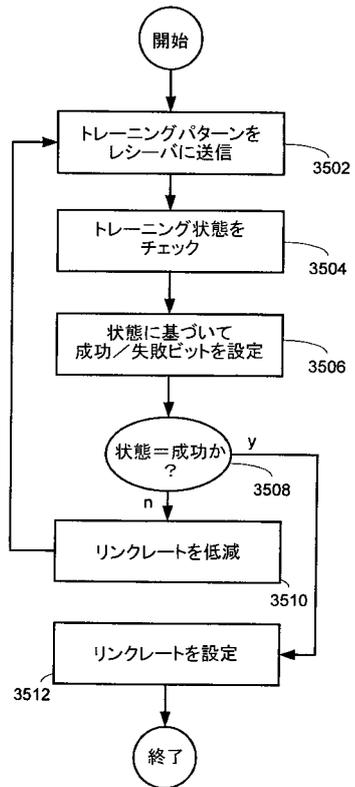


Fig. 35

【 図 3 6 】

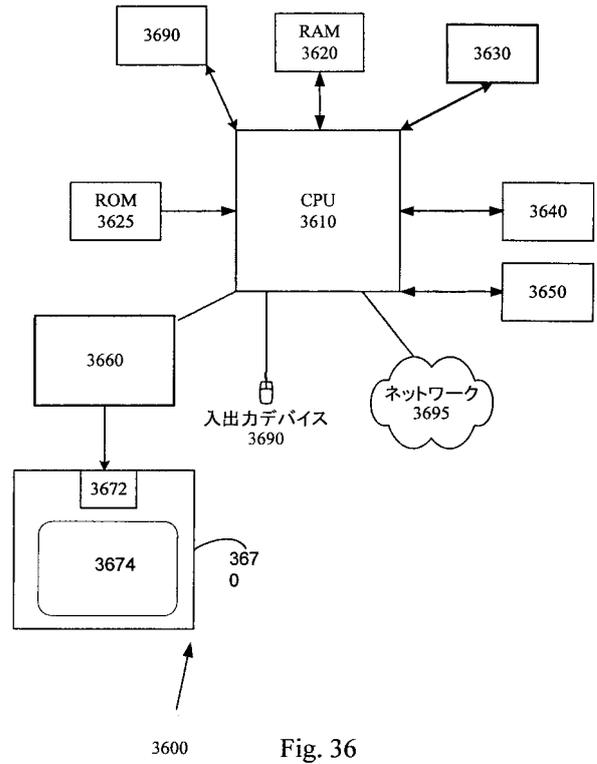


Fig. 36

フロントページの続き

(72)発明者 小林 修

アメリカ合衆国 カリフォルニア州 9 4 0 2 4 ロス・アルトス, フォーレン・リーフ・レーン,
1 4 6 4

Fターム(参考) 5C082 AA01 BA12 BA34 BA35 BB01 BB44 BC03 BD01 BD02 CA84
CB01 DA06 DA26 DA87 DA89 MM02
5K030 GA03 GA06 HB02 JA07 KX18

【 外国語明細書 】

1. TITLE OF THE INVENTION

[0001] PACKET BASED STREAM TRANSPORT SCHEDULER AND METHODS OF USE THEREOF

2. DETAILED DESCRIPTION OF THE INVENTION

Field of the Invention

[0002] The invention relates to multimedia devices. More specifically, the invention describes a data packet stream scheduler and methods of use thereof.

Background of the invention

[0003] Raster scan video transport protocols were originally developed for use with cathode ray tube (CRT) based display systems that must take into account the fact that an electron gun(s) is used to physically "paint" the displayed image one line at a time. For example, a standard definition (VGA) video image is formed of an active region that nominally includes 480 active display lines each of which is formed of 640 pixels (i.e., 640 x 480 resolution). In addition to the active region, however, a blanking region that is not displayed but nonetheless is included in the video signal since it represents that amount of time that is required for both horizontal and vertical retrace. For example, each frame of a VGA image (i.e., one full frame being 480 lines of 640 pixels each) requires approximately 160 pixel clocks per line for horizontal retrace and a period of time equal to approximately 45 line periods for vertical retrace. In this way (assuming one pixel per pixel clock) the video signal required to transport the video data necessary to display the VGA image must be on the order to 800 pixel clocks (640 active pixel clocks + 160 blanking pixel clocks). Therefore, the transport efficiency (as defined as the bandwidth of the displayable data over the total data stream bandwidth) is on the order of 80% (i.e., 640/800).

[0004] More recently, as the resolution of CRTs has increased in order to accommodate HDTV and other high end graphics applications, the efficiency of raster scan video transport protocols have been increased to approximately 90% by requiring that the horizontal retrace be limited to 160 pixel clocks (thereby reducing the associated blanking period). For example, given a UVGA image (i.e., 1600 x 1200), the transport efficiency is approximately 90% when the horizontal retrace is maintained at 160 pixel clocks (1600/(1600+160)) Although raster scan video transfer protocols are efficient (on the order of 90%) and do not require large buffers, they are, however, inflexible in that it is essentially capable of only displaying data as it is rendered.

[0005] In addition to raster scan video transport protocols, the emergence of digital video based systems has created the need for digital video transport protocols. One such digital video transport protocol referred to I.E.E.E. 1394, or FireWireTM is based upon isochronous packet transport that relies upon a large buffer (on the order of 60Kb) in order to guarantee a uniform bit rate and maintain synchronicity between multiple data streams (such as a video stream and an associated soundtrack in the form of an audio stream). Although isochronous packet transfer protocols are inherently flexible (due to their packet based nature), the large buffer requirements can be very costly.

[0006] Therefore, what is desirable is a data stream transport protocol that has the efficiency (in terms of both transport efficiency and memory resource utilization) of the raster scan transfer protocol and the flexibility of the isochronous packet transfer protocol.

SUMMARY OF THE INVENTION

[0007] A method of coupling a multimedia source device to a multimedia sink device by providing a source device having a transmitter unit coupled thereto, providing sink device having a receiver unit coupled thereto, receiving a source data stream in accordance with a native stream rate by the transmitter unit, coupling the transmitter unit and the receiver unit by way of a linking unit, forming a multimedia data packet stream formed of a number of multimedia data packets and generating a transport schedule for transferring the multimedia data packet stream in accordance with a link rate between the transmitter unit and the receiver unit wherein the multimedia data packets are each a fixed size based upon the link rate and a data stream bit rate.

[0008] A method of scheduling a transport of a number of data packets between a data source and a data sink by way of a data link by sending data packet attributes from the data packet source to the data packet sink, comparing a stream bit rate to a data link bit rate for each of a number of data streams to be sent from the source to the sink, setting a packet size for each of the data streams based upon the comparing wherein the packet size is a fixed packet size, combining at least one of each data packet, and transporting the combined data packets from the source to the sink.

[0009] In yet another embodiment, computer program product for scheduling a transport of a number of data packets between a data source and a data sink by way of a data link that includes computer code for sending data packet attributes from the data packet source to the data packet sink, computer code for comparing a stream bit rate to a data link bit rate for each of a number of data streams to be sent from the source to the sink, computer code for setting a packet size for each of the data streams based upon the comparing wherein the packet size is a fixed packet size, computer code for combining at least one of each data packet, computer code for transporting the combined data packets from the source to the sink, and computer readable medium for storing the code.

DETAILED DESCRIPTION OF SELECTED EMBODIMENTS

[0010] Reference will now be made in detail to a particular embodiment of the invention an example of which is illustrated in the accompanying drawings. While the invention will be described in conjunction with the particular embodiment, it will be understood that it is not intended to limit the invention to the described embodiment. To the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims.

[0011] The invention will now be described in terms of a video display system having a video source coupled to a video sink, or receiver, by way of a packet based digital interface. A transmitter unit coupled to the source device receives any number of packetized video data streams each having associated stream attributes. In terms of the video system under discussion, such attributes include video format, color depth, etc. The receiver unit is coupled to the source by way of a data, or main link, and an associated auxiliary link used, in part, to transfer the stream attribute data from the source to the receiver prior to the transmission of the data packets by way of the main link. In this way, the packet headers are used to primarily identify which data stream a data packet is associated and therefore can include only a stream ID, or other such identifier. In thi

s way packet overhead is substantially reduced packet overhead preserving main link bandwidth for multimedia content, such as video and audio data providing an efficient packet transport mechanism. In order to co-ordinate the transmission of the data in the main link, a transport stream scheduler provides for a flexible, efficient system, method, and apparatus for scheduling any of a number of packetized data streams over the data link where an auxiliary channel for sending stream attribute data from a source to a sink may also be provided.

[0012] Each data stream is formed of a number of associated data packets whose size depends upon the relative portion of link bandwidth required for that particular data stream. For a particular data stream i an associated packet size PS_i is related to a maximum packet size MPS , a link bit rate LBR and a stream bit rate SBR_i by way of the following relationship:

$$PS_i = MPS * SBR_i / (LBR_i + 1).$$

[0013] In this way, a packet size is determined for each data stream based upon the relative bandwidth of the data stream compared to a data link bandwidth. For example, in the case where the maximum packet size is 64 link symbols, and the link bit rate LBR is 2.5Gbps per lane Table 1 shows comprehensive packet sizes corresponding to selected stream bit rates

Table 1

Link Bit Rate (LBR) Stream Bit Rate (SBR) Packet Size (PS)

2.5 Gbps/lane	1.0 Gbps/lane	32 link symbols
2.5 Gbps/lane	0.3125 Gbps/lane	10 link symbols
2.5 Gbps/lane	0.25 Gbps/lane	8 link symbols
InterLaneAlignment (ILA)		2 link symbols

[0014] The scheduler time division multiplexes (at the transmitter) and demultiplexes (at the receiver) the packets of the multiple streams into a corresponding link data stream. In the described embodiment, the transmitter reads the capability of the receiver in terms of, for example, the maximum data link rate, the available buffer size and the number of time base recovery (TBR) units. With this knowledge, the transmitter is able to determine the most efficient transport configuration and whether or not subsequent data streams can be accommodated by the receiver all without the need to send additional inquiries to the receiver. Prior to commencing the data stream transport, the transmitter notifies the receiver of stream attributes such as in the case of video data, color format and depth, geometry as well as the packet size associated with each data stream.

By communicating this attribute data, the size of the packet headers can be substantially reduced to the point where only a stream ID is required. In this way, the transport efficiency is greatly increased over that provided in conventional packet based transport protocols that require substantially more overhead due to the larger size of the packet headers.

[0015] In order to provide a basis for the discussion of the inventive stream scheduler, a representative digital video system is described well suited for imp

Implementation of the invention.

[0016] Accordingly, Fig. 1 shows a generalized representation of a packet based digital video display interface 100 well suited for implementing any of a number of embodiments of the invention. The interface 100 connects a transmitter 102 to a receiver 104 by way of a physical link 106 (also referred to as a pipe). In the described embodiment, a number of data streams 108 _ 112 are received at the transmitter 102 that, if necessary, packetizes each into a corresponding number of data packets 114. These data packets are then formed into corresponding data streams each of which are passed by way of an associated virtual pipe 116 _ 120 to the receiver 104. It should be noted that the link rate (i.e., the data packet transfer rate) for each virtual link can be optimized for the particular data stream resulting in the physical link 106 carrying data streams each having an associated link rate (each of which could be different from each other depending upon the particular data stream). The data streams 110 _ 114 can take any number of forms such as video, graphic, audio, etc.

[0017] Typically, when the source is a video source, the data streams 110 _ 114 include various video signals that can have any number and type of well-known formats, such as composite video, serial digital, parallel digital, RGB, or consumer digital video. The video signal can be an analog video signal provided the source 102 includes some form of an analog video source such as for example, an analog television, still camera, analog VCR, DVD player, camcorder, laser disk player, TV tuner, set top box (with satellite DSS or cable signal) and the like.

The source 102 can also include a digital image source such as for example a digital television (DTV), digital still camera, and the like. The digital video signal can be any number and type of well known digital formats such as, SMPTE 274M-1995 (1920 x 1080 resolution, progressive or interlaced scan), SMPTE 296M-1997 (1280 x 720 resolution, progressive scan), as well as standard 480 progressive scan video.

[0018] In the case where the source 102 provides an analog image signal, an analog-to-digital converter (A/D) converts an analog voltage or current signal into a discrete series of digitally encoded numbers (signal) forming in the process an appropriate digital image data word suitable for digital processing. Any of a wide variety of A/D converters can be used. By way of example, other A/D converters include, for example those manufactured by: Philips, Texas Instrument, Analog Devices, Brooktree, and others.

[0019] For example, if the data stream 110 is an analog type signal, the an analog to digital converter (not shown) included in or coupled to the transmitter 102 will digitize the analog data which is then packetize by a packetizer that converts the digitized data stream 110 into a number of data packets 114 each of which will be transmitted to the receiver 104 by way of the virtual link 116. The receiver 104 will then reconstitute the data stream 110 by appropriately recombining the data packets 114 into their original format. It should be noted that the link rate is independent of the native stream rates. The only requirement is that the link bandwidth of the physical link 106 be higher than the aggregate bandwidth of data stream(s) to be transmitted. In the described embodiment, the incoming data (such as pixel data in the case of video data) is packed over the respective virtual link based upon a data mapping definition. In this way, the physical link 106 (or any of the constituent virtual links) does not, as does conventional interconnects such as DVI, carry one pixel data per link character clock.

[0020] In this way, the interface 100 provides a scaleable medium for the transport of not only video and graphics data, but also audio and other application data as may be required. In addition, the invention supports hot-plug event detection and automatically sets the physical link (or pipe) to its optimum transmission rate. The invention provides for a low pin count, purely digital display interconnect for all displays suitable for multiple platforms. Such platforms include host to display, laptop/all-in-one as well as HDTV and other consumer electronics applications.

[0021] In addition to providing video and graphics data, display timing information can be embedded in the digital stream providing essentially perfect and instant display alignment, obviating the need for features like "Auto-Adjust" and the like. The packet based nature of the inventive interface provides scalability to support multiple, digital data streams such as multiple video/graphics streams and audio streams for multimedia applications. In addition, a universal serial bus (USB) transport for peripheral attachment and display control can be provided without the need for additional cabling.

[0022] Other embodiments of the inventive display interface will be discussed below.

[0023] Fig. 2 illustrates a system 200 based upon the system 100 shown in Fig. 1 that is used to connect a video source 202 and a video display unit 204. In the illustrated embodiment, the video source 202 can include either or both a digital image (or digital video source) 206 and an analog image (or analog video source) 208. In the case of the digital image source 206, a digital data stream 210 is provided to the transmitter 102 whereas in the case of the analog video source 208, an A/D converter unit 212 coupled thereto, converts an analog data stream 213 to a corresponding digital data stream 214. The digital data stream 214 is then processed in much the same manner as the digital data stream 210 by the transmitter 102. The display unit 204 can be an analog type display or a digital type display or in some cases can process either analog or digital signals provided thereto. In any case, the display unit 204 includes a display interface 216 that interfaces the receiver 104 with a display 218 and a D/A converter unit 220 in the case of an analog type display. In the described embodiment, the video source 202 can take any number of forms (such as a personal desktop computer, digital or analog TV, set top box, etc.) whereas the video display unit 104 can take the form of a video display (such as an LCD type display, CRT type display, etc.).

[0024] Regardless of the type of video source or video sink, however, the various data streams are digitized (if necessary) and packetized prior to transmission over the physical link 106 which includes a uni-directional main link 222 for isochronous data streams and a bi-directional auxiliary channel 224 for link set up and other data traffic (such as various link management information, Universal serial bus (USB) data, etc.) between the video source 202 and the video display 204.

[0025] The main link 222 is thereby capable of simultaneously transmitting multiple isochronous data streams (such as multiple video/graphics streams and multi-channel audio streams). In the described embodiment, the main link 222 includes a number of different virtual channels, each capable of transferring isochronous data streams (such as uncompressed graphics/video and audio data) at multiple gigabits per second (Gbps). From a logical viewpoint, therefore, the main link 222 appears as a single physical pipe and within this single physical pipe, mul

multiple virtual pipes can be established. In this way, logical data streams are not assigned to physical channels rather, each logical data stream is carried in its own logical pipe (i.e., virtual channel described above).

[0026] In the described embodiment, the speed, or transfer rate, of the main link 222 is adjustable to compensate for link conditions. For example, in one implementation, the speed of the main link 222 can be adjusted in a range approximated by a slowest speed of about 1.0 Gbps to about 2.5 Gbps per channel in approximately 0.4Gbps increments (see Fig. 3). At 2.5 Gbps per channel, the main link 222 can support SXGA 60Hz with a color depth of 18 bits per pixel over a single channel. It should be noted that a reduction in the number of channels reduces not only the cost of interconnect, but also reduces the power consumption which is an important consideration (and desirable) for power sensitive applications such as portable devices and the like. However, by increasing the number of channels to four, the main link 222 can support WQSXGA (3200 x 2048 image resolution) with a color depth of 24-bits per pixel at 60Hz. or QSXGA (2560 x 2048) with a color depth of 18-bits per pixel at 60Hz, without data compression. Even at the lowest rate of 1.0 Gbps per channel, only two channels are required to support an uncompressed HDTV (i.e., 1080i or 720p) data stream.

[0027] In the described embodiment, a main link data rate is chosen whose bandwidth exceeds the aggregate bandwidth of the constituent virtual links. Data sent to the interface arrives at the transmitter at its native rate. A time-base recovery (TBR) unit 226 within the receiver 104 regenerates the stream's original native rate using time stamps embedded in the main link data packets, if necessary. It should be noted, however, that for appropriately configured digital display devices 232 shown in Fig. 2B, time base recovery is unnecessary since display data is sent to the display driver electronics at the link character clock rate, thereby greatly reducing the number of channels required with a commensurate reduction in complexity and cost for the display. For example, Fig. 2C illustrates an exemplary LCD panel 232 configured in such a way that no time base recovery since display data is essentially pipelined to the various column drivers 234 that are used in combination with row drivers 236 to drive selected display elements 238 in the array 240.

[0028] Other embodiments describe a simple enumeration method for the link rate and the pixel/audio clock rate. It has been researched and understood that all the standard pixel/audio clock frequencies that exist today are a subset of the following master frequency:

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

[0029] This means that a pixel (or audio) clock rate can be expressed with four parameters, A, B, C, and D as:

$$\text{Pixel clock rate} = 2A \times 3B \times 5C \times 11D$$

$$A = 4 \text{ bits}, B = 2 \text{ bits}, C = 3 \text{ bits}, \text{ and } D = 1 \text{ bit.}$$

[0030] Even for a link whose link rate (which is the serial link bit rate / 10 for a link that uses 10-bit character such as 8B/10B characters) may be different from the pixel clock rate, there is a benefit in defining the link rate with these four parameters, A', B', C', and D': The benefit is the simplicity in regenerating pixel/audio clocks from a link clock. For example, let's say the link rate is set as A' = 6, B' = 3, C' = 7, and D' = 0 and the corresponding link rate is 135MHz. However, suppose the pixel clock rate is set as A = 8, B = 3, C = 6, and D = 0 (= 108MHz), then the pixel clock can be generated from link clock as pixel clock rate is equal to the link rate * 22 / 51 .

[0031] Referring back to those systems requiring time base recovery, the time-base recovery unit 226 may be implemented as a digital clock synthesizer. For an uncompressed video stream, the time stamp is stored in the packet header which as described in more detail below, is a 20-bit value. For a given stream, four of 20 bits are stored in each header successively (TS3-0, TS7-4, TS11-8, TS15-12, TS19-16). Native stream frequency (Freq_{native}) is obtained from link character clock frequency (Freq_{link_char}) as:

$$\text{Freq}_{\text{native}} = \text{Freq}_{\text{link_char}} * (\text{TS19-0})/220.$$

[0032] The transmitter 102 generates this time stamp by counting the number of native stream clocks in 220cycles of the link character clock frequency period. The counter updates the value every 220 cycles of the link character clock. Since these two clocks are asynchronous with each other, the time stamp value will change by 1 over time. Between updates, the transmitter 102 will repeatedly send the same time stamp in the header of the given packet stream. A sudden change of the time stamp value (by more than 1 count) may be interpreted by the receiver as an indication of an unstable condition of the stream source.

[0033] It should be noted that, no time stamp is communicated for an audio stream. In this case, the source device informs the display device of the audio sample rate and number of bits per sample. By determining the audio rate as below and the link character rate, the display device regenerates the original audio stream rate.

$$\text{Audio rate} = (\text{audio sample rate}) \times (\# \text{ bits per sample}) \times (\# \text{ channels})$$

[0034] A main link data packet 400 shown in Fig. 4A includes a main link packet header 402 as shown in Fig. 4B that is formed of 16 bits where bits 3-0 are the Stream ID (SID) (indicating that maximum stream count is 16), bit 4 is the Time Stamp (TS) LSB. When bit 4 is equal to 1, this packet header has the least significant 4 bits of Time Stamp value (used only for uncompressed video stream). Bit 5 is a Video frame sequence bit which acts as the least significant bit of the frame counter which toggles from "0" to "1" or from "1" to "0" at the video frame boundary (used only for uncompressed video stream). Bits 7 and 6 are reserved whereas bits 8 through 10 are a 4-bit CRC (CRC) that checks errors for the previous eight bits. Bits 15-12 are Time Stamp/Stream ID Inversion. (TSP/SIDn) which for uncompressed video are used as four bits of 20-bit Time Stamp value.

[0035] One of the advantages of the inventive interface is the ability to multiplex different data streams each of which can be different formats as well as have certain main link data packets include a number of sub packets. For example, Fig. 5A shows a system 500 arranged to provide sub-packet enclosure and multiple-packet multiplexing in accordance with an embodiment of the invention. It should be noted that the system 500 is a particular embodiment of the system 200 shown in Fig. 2 and should therefore not be construed as limiting either the scope or intent of the invention. The system 500 includes a stream source multiplexer 502 included in the transmitter 102 used to combine a stream 1 supplemental data stream 504 with the data stream 210 to form a multiplexed data stream 506. The multiplexed data stream 506 is then forwarded to a link layer multiplexer 508 that combines any of a number of data streams to form a multiplexed main link stream 510 formed of a number of data packets 512 some of which may include any of a number of sub packets 514 enclosed therein. A link layer de-multiplexer 516 splits the multiplexed data stream 510 into its constituent data streams based on the stream IDs (SIDs) and associated sub packet headers while a stream sink de-multiplexer 518 further splits off the stream 1 supplemental data stream contain

ned in the sub-packets.

[0036] Fig. 6 shows a high-level diagram of the multiplexed main link stream 600 as an example of the stream 510 shown in Fig. 5 when three streams are multiplexed over the main link 222. The three streams in this example are: UXGA graphics (Stream ID = 1), 1280x720p video (Stream ID = 2), and audio (Stream ID = 3). The small packet header size of main link packet 400 minimizes the packet overhead, which results in the very high link efficiency. The reason the packet header can be so small is that the packet attributes are communicated via the auxiliary channel 224 prior to the transmission of the packets over main link 222.

[0037] Generally speaking, the sub-packet enclosure is an effective scheme when the main packet stream is an uncompressed video since an uncompressed video data stream has data idle periods corresponding to the video-blanking period. Therefore, main link traffic formed of an uncompressed video stream will include series of Null special character packets during this period. By capitalizing on the ability to multiplex various data streams, certain implementations of the present invention use various methods to compensate for differences between the main link rate and the pixel data rate when the source stream is a video data stream. For example, as illustrated in Fig. 7, the pixel data rate is .5 Gb/sec, such that a bit of pixel data is transmitted every 2 ns. In this example, the link rate has been set to 1.25 Gb/sec, such that a bit of pixel data is transmitted each .8 ns. Here, transmitter 102 intersperses special characters between pixel data as illustrated in Fig. 8. Two special characters are disposed between a first bit of pixel data P1 and a second bit of pixel data P2. The special characters allow receiver 104 to distinguish each bit of pixel data. Interspersing the special characters between bits of pixel data also creates a steady stream of data that allows the link to maintain synchronization. In this example, the special characters are Null characters. No line buffer is needed for such methods, only a small FIFO, because the link rate is sufficiently fast. However, relatively more logic is required on the receiving side to reconstruct the video signal.

The receiver needs to recognize when the special characters begin and end.

[0038] An alternative to the interspersing method is to alternate consecutive bits of pixel data with special characters, such as null values. For example, P1 through P4 could be fed into a line buffer included in the transmitter 104, then one or more null values could be fed into the buffer until more pixel data are available. Such implementations require a relatively larger buffer space than the interspersing methods described above. In many such implementations, the time required to fill the line buffer will exceed the time required to transmit the data after the line buffer is full, due to the relatively high link speeds.

[0039] As discussed with reference to Fig. 5A, one of the advantages of the inventive interface is the ability to not only multiplex various data streams, but also the enclosing of any of a number of sub packets within a particular main link data packet. Fig. 9A shows a representative sub-packet 900 in accordance with an embodiment of the invention. The sub-packet 900 includes a sub-packet header 902 that in the described embodiment is 2 bytes and is accompanied by SPS (Sub-Packet Start) special character. If the main link data packet in which the sub-packet 900 is enclosed contains a packet payload in addition to the sub-packet 900, the end of the sub-packet 900 must be marked by SPE (Sub-Packet End) special character. Otherwise, the end of the main packet (as indicated by ensuing COM character in the example shown in Fig. 9B) marks the end of both the sub-packet 902 and the main packet into which it is enclosed. However, a sub-packet does no

t need to end with SPE when its enclosing main packet has no payload. Fig 9B shows an exemplary sub-packet format within a main link packet in accordance with an embodiment of the invention. It should be noted that the definition of the header field and sub-packet payload is dependent on the specific application profile that uses the sub-packet 902.

[0040] A particularly useful example of sub-packet enclosure usage is selective refresh of an uncompressed graphics image 1000 illustrated in Fig. 10. The attributes of the entire frame 1002 (Horizontal/Vertical Total, Image Width/Height, etc.) will be communicated via the auxiliary channel 224 since those attributes stay constant as long as the stream remains valid. In selective refresh operation, only a portion 1004 of the image 1000 is updated per video frame. The four X-Y coordinates of the updated rectangle(s) (i.e., the portion 1004) must be transmitted every frame since the values of the rectangle coordinates changes from frame to frame. Another example is the transmission of color look-up table (CLUT) data for required for 256-color graphic data where the 8-bit pixel data is an entry to the 256-entry CLUT and the content of the CLUT must be dynamically updated.

[0041] The single bi-directional auxiliary channel 224 provides a conduit to for various support functions useful for link set up and supporting main link operations as well as to carry auxiliary application data such as USB traffic. For example, with the auxiliary channel 224, a display device can inform the source device of events such as sync loss, dropped packets and the results of training sessions (described below). For example, if a particular training session fails, the transmitter 102 adjusts the main link rate based upon pre-selected or determined results of the failed training session. In this way, the closed loop created by combining an adjustable, high speed main link with a relatively slow and very reliable auxiliary channel allows for robust operation over a variety of link conditions. It should be noted that in some cases (an example of which is shown in Fig. 5B), a logical bi-directional auxiliary channel 520 can be established using a portion 522 of the bandwidth of the main link 222 to transfer data from the source device 202 to the sink device 204 and a uni-directional back channel 524 from the sink device 204 to the source device 202. In some applications, use of this logical bi-directional auxiliary channel may be more desirable than using a half-duplex bi-directional channel described in Fig. 5A.

[0042] Prior to starting the transmission of actual packet data streams the transmitter 102 establishes a stable link through a link training session that is analogous in concept to the link setup of the modem. During link training, the main link transmitter 102 sends a pre-defined training pattern so that the receiver 104 can determine whether it can achieve a solid bit/character lock. In the described embodiment, training related handshaking between the transmitter 102 and the receiver 104 is carried on the auxiliary channel. An example of a link training pattern is shown in Fig. 11 in accordance with an embodiment of the invention. As illustrated, during the training session, a phase 1 represents the shortest run length while phase 2 is the longest that are used by the receiver to optimize an equalizer. In phase 3, both bit lock and character lock are achieved as long as the link quality is reasonable. Typically, the training period is about 10 ms, in which time, approximately 107 bits of data are sent. If the receiver 104 does not achieve solid lock, it informs the transmitter 102 via the auxiliary channel 224 and the transmitter 102 reduces the link rate and repeats the training session.

In addition to providing a training session conduit, the auxiliary channel 224 can be also used to carry main link packet stream descriptions thereby greatly reducing the overhead of packet transmissions on the main link 222. Furthermore, the auxiliary channel 224 can be configured to carry Extended Display Identification Data (EDID) information replacing the Display Data Channel (DDC) found on all monitors (EDID is a VESA standard data format that contains basic information about a monitor and its capabilities, including vendor information, maximum image size, color characteristics, factory pre-set timings, frequency range limits, and character strings for the monitor name and serial number. The information is stored in the display and is used to communicate with the system through the DDC which sits between the monitor and the PC graphics adapter. The system uses this information for configuration purposes, so the monitor and system can work together). In what is referred to as an extended protocol mode, the auxiliary channel can carry both asynchronous and isochronous packets as required to support additional data types such as keyboard, mouse and microphone.

[0043] Fig. 12 shows a representative system 1200 having a data stream scheduler 1202 in accordance with an embodiment of the invention. It should be noted that the system 1200 is based upon the system 500 described with reference to figures 5A and 5B and should therefore be considered only one of any number of implementations of the invention. Accordingly, the stream scheduler 1202 is incorporated in or coupled to the video source 202 and is, in turn, coupled to a multiplexor 1204 and a data buffer 1206 suitable for storing only those portions of the incoming data streams (S1, S2, and S3) used to populate a link data stream 1208 during a period of time referred to as a scheduler cycle time T_{sched} . In the described embodiment, the data buffer 1206 is typically on the order of tens of bytes in size as opposed to over 60K bytes typical for isochronous video transfer protocols such as FireWire免och. In this way, the link efficiency (based upon a comparing of the aggregate of the payload bandwidth of data streams divided by the available link bandwidth) is approximately on the order of 90% or greater.

[0044] In the described embodiment, the link data stream 1208 is formed by the scheduler 1202 by using time division multiplexing to combine size data packets P1, P2, and P3 (whose size reflects the relative bandwidth for each data stream in relation to a link bandwidth) from each of the data streams S1, S2, and S3, respectively. As described previously, the size of each data packet is a function of the particular data stream bit rate (SBR) and the link bit rate (LBR). In particular, the greater the particular data stream bit rate, the larger the particular data packet size as shown by the example of Table 1. For example, if the link bit rate LBR is on the order of 2.5 Gbps and the maximum packet size is approximately on the order of 80 link symbols (where one link symbol is defined as a data unit per link clock and is typically 4ns) and using the assumptions of Table 1 (i.e., SBR1 is 1.0 Gbps, SBR2 is 0.3125 Gbps and SBR3 is 0.25 Gbps) the data packet P1 associated with stream S1 is 32 link symbols while the data packets P2 and P3 associated with streams S2 and S3 are 10 link symbols and 8 link symbols, respectively. It should be noted that at the beginning of each scheduler cycle time T_{sched} , the scheduler 1202 inserts an interlane alignment packet (ILA) that provides an alignment tool for the receiver 204 that is typically on the order of 2 link symbols in size. In this example, therefore, the scheduler cycle time T_{sched} is on the order of $(32 + 10 + 8 + 2)$ 52 link symbols (which would translate to approximately 208 ns when each link symbol represents approximately 4 ns).

[0045] It should also be noted that each data packet P has an associated active data ratio of the number of data symbols D and stuffing symbols N related to the stream bit rate (SBR) and the link bit rate (LBR) by

$$\text{SBR/LBR} = D/(D+N).$$

[0046] Therefore in those saturations where a data stream is either added (or deleted) resulting in an increase (or decrease) of the scheduler cycle time T_{sched} , the packet size P for remains constant by varying the number of stuffing symbols N in relation to the number of data symbols D. Since with the addition (or deletion) of a data stream, since the particular packet sizes remain the same for the other data streams and the scheduler cycle time T_{sched} increases (or decreases) then the number of stuffing symbols N increases (or decreases) commensurate with the change in T_{sched} . In the "degenerate" case where there remains only a single data stream, then there are no stuffing symbols N.

[0047] Fig. 13 shows a more detailed portion 1300 of the data stream 1210 in accordance with an embodiment. In particular, Fig. 13 shows the arrangement of data symbols D and stuffing symbols N for the data stream 1208 using the values shown in Table 1. It should also be noted that dynamically adding or deleting a particular data stream leaves the particular packet sizes for the remaining data stream unaffected. Accordingly, Fig. 14 illustrates the addition of a fourth data stream S4 having a stream bit rate of .625 Gbps that corresponds to a packet size P4 of 20 link symbols resulting in an increase in T_{sched} from 52 to 72 link symbols (that corresponds to 288 ns for a link symbol equal to 4 ns). However, in order to keep the particular packet sizes P1, P2, and P3 constant, the number of stuffing symbols N increases for each data packet. Conversely, in the case where a data stream is deleted (such as S3, for example) the scheduler cycle time T_{sched} will decrease accordingly with an commensurate increase in the number of data symbols D in relation to the number of stuffing symbols N. In the "degenerate" case where all data streams save one is deleted, then the remaining data packet has no stuffing symbols N and there is no need for any buffer thereby simulating the raster scan transport protocols discussed above.

[0048] Also, in the case where of the degenerate connection when the link data stream 1208 is a single, uncompressed video stream (as shown in Fig. 15) the ILA packets are placed in the idle period (the horizontal blanking region) of S1 and the active display region is then represented by a mixture of data symbols D and stuffing symbols N (see Fig. 16).

[0049] It should also be noted that the relative number of data symbols D provides an embedded time stamp in that by counting the number of data symbols D for a particular data stream with relation to data symbols not related to the particular data stream provides the stream clock for the data stream in question. For example, in the case shown in Fig. 13, in order to recover a stream clock $F_{\text{stream_clk}}$ for a particular data stream can be determined by simply determining the number of stream data symbols (M) as compared to the total number of stuffing symbols and stream data symbols (P). More particularly, the stream clock $F_{\text{stream_clk}}$ is determined by the following:

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

where M and P can be measured by the receiver 204.

[0050] Fig. 17 shows a flowchart detailing a process 1700 for scheduling a number of data streams in accordance with an embodiment of the invention. The process 1700 begins at 1702 by the transmitter sending stream attribute data to the receiver. In the described embodiment, the attribute data is sent by way of an a

uxiliary link that couples the transmitter to the receiver. Next, at 1704, the receiver allocates sufficient resources (if available) to receive and process the incoming data streams. After the transmitter has received the data streams at 1706, the stream transport scheduler forms a link data stream at 1708 while at 1710 the link data stream is sent by way of a main link by the transmitter to the receiver. Next, at 1712, in those cases where an additional data stream is to be added and the receiver is capable of both receiving and processing the to be added data stream at 1714, the new data stream is added transparently at 1716 in that the data packet size for the previously transmitted data streams remain constant.

[0051] Fig. 18 shows a flowchart detailing a process 1800 for forming the link data stream in accordance with an embodiment of the invention. It should be noted that the process 1800 is a particular implementation of forming operation 1708 of the process 1700. Accordingly, the transport scheduler determines the link bit rate for the main link at 1802 and the stream bit rate for each of the data streams to be transported by way of the link from the transmitter to the receiver at 1804. Next, at 1806, the packet size for each of the data streams is determined based upon the stream bit rate, the link bit rate, and a predetermined maximum packet size. Next at 1808, data packets for each of the data streams is formed each including a number of data symbols and stuffing symbols. Once the data packets are formed and during a single transport scheduler cycle time, the transport scheduler (using time division multiplexing) concatenates a data packet from each of the data streams to be sent to the receiver at 1810 and adds an interlane alignment packet to the concatenated data packets at 1812. At 1814, the transport scheduler signals the transmitter to transmit the concatenated data packets to the receiver and repeats 1808 through 1814 until the cessation of the transmission as determined at 1816.

[0052] Fig. 19 illustrates a logical layering 1900 of the system 200 in accordance with an embodiment of the invention. It should be noted that while the exact implementation may vary depending upon application, generally, a source (such as the video source 202) is formed of a source physical layer 1902 that includes transmitter hardware, a source link layer 1904 that includes multiplexing hardware and state machine (or firmware), and a data stream source 1906 such as Audio/Visual/Graphics hardware and associated software. Similarly, a display device includes a physical layer 1908 (including various receiver hardware), a sink link layer 1910 that includes de-multiplexing hardware and state machine (or firmware) and a stream sink 1912 that includes display/timing controller hardware and optional firmware. A source application profile layer 1914 defines the format with which the source communicates with the link layer 1904 and similarly, a sink application profile layer 1916 defines the format with which the sink 1912 communicates with the sink link layer 1910.

[0053] The various layers will now be discussed in more detail.

SOURCE DEVICE PHYSICAL LAYER

[0054] In the described embodiment, the source device physical layer 1902 includes an electrical sub layer 1902-1 and a logical sub layer 1902-2. The electrical sub layer 1902-1 includes all circuitry for interface initialization/operation such as hot plug/unplug detection circuit, drivers/receivers/termination resistors, parallel-to-serial/serial-to-parallel conversions, and spread-spectrum-capable PLL's. The logical sub layer 1902-2 includes circuitry for, packetizing/d

e-packetizing, data scrambling/de-scrambling, pattern generation for link training, time-base recovery circuits, and data encoding/decoding such as 8B/10B (as specified in ANSI X3.230-1994, clause 11) that provides 256 link data characters and twelve control characters (an example of which is shown as Fig. 13) for the main link 222 and Manchester II for the auxiliary channel 224 (see Fig. 21).

[0055] It should be noted that the 8B/10B encoding algorithm is described, for example, in U.S. Patent Number 4,486,739, which is hereby incorporated by reference. As known by those of skill in the art, the 8B/10B code is a block code that encodes 8-bit data blocks into 10-bit code words for serial transmission. In addition, the 8B/10B transmission code converts a byte wide data stream of random 1s and 0s into a DC balanced stream of 1s and 0s with a maximum run length of 5. Such codes provide sufficient signal transitions to enable reliable clock recovery by a receiver, such as transceiver 110. Moreover, a DC balanced data stream proves to be advantageous for fiber optic and electromagnetic wire connections. The average number of 1s and 0s in the serial stream is maintained at equal or nearly equal levels. The 8B/10B transmission code constrains the disparity between the number of 1s and 0s to be ± 2 , 0, or 2 across 6 and 4 bit block boundaries. The coding scheme also implements additional codes for signaling, called command codes.

[0056] It should be noted that in order to avoid the repetitive bit patterns exhibited by uncompressed display data (and hence, to reduce EMI), data transmitted over main link 222 is first scrambled before 8B/10B encoding. All data except training packets and special characters will be scrambled. The scrambling function is implemented with Linear Feedback Shift Registers (LFSRs). When data encryption is enabled, the initial value of an LFSR seed is dependent on an encryption key set. If it is data scrambling without encryption, the initial value will be fixed.

[0057] Since data stream attributes are transmitted over the auxiliary channel 224, the main link packet headers serve as stream identification numbers thereby greatly reducing overhead and maximizing link bandwidth. It should also be noted that neither the main link 222 nor the auxiliary link 224 has separate clock signal lines. In this way, the receivers on main link 222 and auxiliary link 224 sample the data and extract the clock from the incoming data stream. Fast phase locking for any phase lock loop (PLLs) circuit in the receiver electrical sub layer is important for since the auxiliary channel 224 is half-duplex bi-directional and the direction of the traffic changes frequently. Accordingly, the PLL on the auxiliary channel receiver phase locks in as few as 16 data periods thanks to the frequent and uniform signal transitions of Manchester II (MII) code

[0058] At link set up time, the data rate of main link 222 is negotiated using the handshake over auxiliary channel 224. During this process, known sets of training packets are sent over the main link 222 at the highest link speed. Success or failure is communicated back to the transmitter 102 via the auxiliary channel 224. If the training fails, main link speed is reduced and training is repeated until successful. In this way, the source physical layer 1902 is made more resistant to cable problems and therefore more suitable for external host to monitor applications. However, unlike conventional display interfaces, the main channel link data rate is decoupled from the pixel clock rate. A link data rate is set so that link bandwidth exceeds the aggregate bandwidth of the transmitted streams.

SOURCE DEVICE LINK LAYER

[0059] The source link layer 1904 handles the link initialization and management. For example, upon receiving a hot plug detect event generated upon monitor power-up or connection of the monitor cable from the source physical layer 1902, the source device link layer 1904 evaluates the capabilities of the receiver via interchange over the auxiliary channel 224 to determine a maximum main link data rate as determined by a training session, the number of time-base recovery units on the receiver, available buffer size on both ends, availability of USB extensions and then notifies the stream source 1906 of an associated hot plug event. In addition, upon request from the stream source 1906, the source link layer 1904 reads the display capability (EDID or equivalent). During a normal operation, the source link layer 1904 sends the stream attributes to the receiver 104 via the auxiliary channel 224, notifies the stream source 1904 whether the main link 222 has enough resource for handling the requested data streams, notifies the stream source 1904 of link failure events such as sync loss and buffer overflow, and sends MCCS commands submitted by the stream source 1904 to the receiver via the auxiliary channel 224. All communications between the source link layer 1904 and the stream source/sink use the formats defined in the application profile layer 1914.

APPLICATION PROFILE LAYER(SOURCE AND SINK)

[0060] In general, the Application Profile Layer defines formats with which a stream source (or sink) will interface with the associated link layer. The formats defined by the application profile layer are divided into the following categories, Application independent formats (Link Message for Link Status inquiry) and Application dependent formats (main link data mapping, time-base recovery equation for the receiver, and sink capability/stream attribute messages sub-packet formats, if applicable). The Application Profile Layer supports the following color formats 24-bit RGB, 16-bit RG2565, 18-bit RGB, 30-bit RGB, 256-color RGB (CLUT based), 16-bit, CbCr422, 20-bit YCbCr422, and 24-bit YCbCr444.

[0061] For example, the display device application profile layer (APL) 1914 is essentially an application-programming interface (API) describing the format for Stream Source/Sink communication over the main link 222 that includes a presentation format for data sent to or received from the interface 100. Since some aspects of the APL 1914 (such as the power management command format) are baseline monitor functions, they are common to all uses of the interface 100. Whereas other non-baseline monitor functions, such as such as data mapping format and stream attribute format, are unique to an application or a type of isochronous stream that is to be transmitted. Regardless of the application, the stream source 1904 queries the source link layer 1914 to ascertain whether the main link 222 is capable of handling the pending data stream(s) prior to the start any packet stream transmission on the main link 222.

[0062] When it is determined that the main link 222 is capable of supporting the pending packet stream(s), the stream source 1906 sends stream attributes to the source link layer 1914 that is then transmitted to the receiver over the auxiliary channel 224. These attributes are the information used by the receiver to identify the packets of a particular stream, to recover the original data from the stream and to format it back to the stream's native data rate. The attributes of the data stream are application dependent.

[0063] In those cases where the desired bandwidth is not available on the main

link 222, the stream source 1914 may take corrective action by, for example, reducing the image refresh rate or color depth.

DISPLAY DEVICE PHYSICAL LAYER

[0064] The display device physical layer 1916 isolates the display device link layer 1910 and the display device APL 1916 from the signaling technology used for link data transmission/reception. The main link 222 and the auxiliary channel 224 have their own physical layers, each consisting of a logical sub layer and an electrical sub layer that includes the connector specification. For example, the half-duplex, bi-directional auxiliary channel 224 has both a transmitter and a receiver at each end of the link as shown in Fig 22. An auxiliary link transmitter 2902 is provided with link characters by a logical sub layer 1908-1 that are then serialized and transmitted to a corresponding auxiliary link receiver 2904. The receiver 2904, in turn, receives serialized link character from the auxiliary link 224 and de-serializes the data at a link character clock rate. It should be noted that the major functions of the source logical sub layers include signal encoding, packetizing, data scrambling (for EMI reduction), and training pattern generation for the transmitter port. While for the receiver port, the major functions of the receiver logical sub layer includes signal decoding, de-packetizing, data de-scrambling, and time-base recovery.

AUXILIARY CHANNEL

[0065] The major functions of auxiliary channel logical sub layer include data encoding and decoding, framing/de-framing of data and there are two options in a auxiliary channel protocol: standalone protocol (limited to link setup/management functions in a point-to-point topology) is a lightweight protocol that can be managed by the Link Layer state-machine or firmware and extended protocol that supports other data types such as USB traffic and topologies such as daisy-chained sink devices. It should be noted that the data encoding and decoding scheme is identical regardless of the protocol whereas framing of data differs between the two.

[0066] Still referring to Fig. 29, the auxiliary channel electrical sub layer contains the transmitter 2902 and the receiver 2904. The transmitter 2902 is provided with link characters by the logical sub layer, which it serializes and transmits out. The receiver 2904 receives serialized link character from the link layer and subsequently de-serializes it at link character clock rate. The positive and negative signals of auxiliary channel 224 are terminated to ground via 50-ohm termination resistors at each end of the link as shown. In the described implementation, the drive current is programmable depending on the link condition and ranges from approximately 8mA to approximately 24mA resulting in a range of $V_{\text{differential_pp}}$ of approximately 400mV to approximately 1.2V. In electrical idle modes, neither the positive nor the negative signal is driven. When starting transmission from the electrical idle state, the SYNC pattern must be transmitted and the link reestablished. In the described embodiment, the SYNC pattern consists of toggling a auxiliary channel differential pair signals at clock rate 28 times followed by four 1's in Manchester II code. The auxiliary channel master in the source device detects hot-plug and hot-unplug events by periodically driving or measuring the positive and negative signals of auxiliary channel 224.

MAIN LINK

[0067] In the described embodiment, the main link 222 supports discrete, variable link rates that are integer multiples of the local crystal frequency (see Fig 3 for a representative set of link rates consonant with a local crystal frequency of 24-MHz). As shown in Fig. 16, the main link 222 (being an unidirectional channel) has only a transmitter 1602 at the source device and only a receiver 1604 at the display device.

[0068] As shown, the cable 2304 takes the form includes a set of twisted pair wires, one for each of the Red (R), Green(G), and Blue(B) video signals provides in a typical RGB color based video system (such as PAL based TV systems). As known by those of skill in the art, twisted pair cable is a type of cable that consists of two independently insulated wires twisted around one another. One wire carries the signal while the other wire is grounded and absorbs signal interference. It should be noted that in some other systems, the signals could also be component based signals (Pb, Pr, Y) used for NTSC video TV systems. Within the cable, each twisted pair is individually shielded. Two pins for +12V power and ground are provided. The characteristic impedance of each differential pair is 100 ohms +/- 20%. The entire cable is also shielded. This outer shield and individual shields are shorted to the connector shells on both ends. The connector shells are shorted to ground in a source device. A connector 2400 as shown in Fig. 24 has 13 pins in one row having a pinout that is identical both for the connector on the source device end and that on the display device end. The source device supplies the power.

[0069] The main link 222 is terminated on both ends and since the main link 222 is AC coupled, the termination voltage can be anywhere between 0V (ground) to +3.6V. In the described implementation, the drive current is programmable depending on the link condition and ranges from approximately 8mA to approximately 24mA resulting in a range of $V_{\text{differential_pp}}$ of approximately 400mV to approximately 1.2V. The minimum voltage swing is selected for each connection using a training pattern. An electrical idle state is provided for power management modes.

In electrical idle, neither the positive nor the negative signals are driven. When starting a transmission from electrical idle state, the transmitter must conduct a training session in order re-establish the link with the receiver.

STATE DIAGRAMS

[0070] The invention will now be described in terms of state diagrams shown in Figs. 25 and 26 described below. Accordingly, Fig.25 shows the source state diagram described below. At an off state 2502, the system is off such that the source is disabled. If the source is enabled, then the system transitions to a standby state 2504 suitable for power saving and receiver detection. In order to detect whether or not the receiver is present (i.e., hot plug/play), the auxiliary channel is periodically pulsed (such as for 1us every 10ms) and a measure of a voltage drop across the termination resistors during the driving is measured. If it is determined that a receiver is present based upon the measured voltage drop, then the system transitions to a detected receiver state 2506 indicating that a receiver has been detected, i.e, a hot plug event has been detected. If, however, there is no receiver detected, then the receiver detection is continued until such time, if ever, a receiver is detected or a timeout has elapsed. It should be noted that in some cases the source device may choose to go to "OFF" state from which no further display detection is attempted.

[0071] If at the state 2506 a display hot unplug event is detected, then the sy

stem transitions back to the standby state 2504. Otherwise the source drives the auxiliary channel with a positive and negative signal to wake up receiver and the receiver's subsequent response, if any, is checked. If there is no response received, then the receiver has not woken up and source remains in the state 2506. If, however, a signal is received from the display, then the display has woken up and the source is ready read the receiver link capabilities (such as max link rate, buffer size, and number of time-base recovery units) and the system transitions to a main link initialization state 2508 and is ready to commence a training start notification phase.

[0072] At this point, a training session is started by sending a training pattern over the main link at a set link rate and checks an associated training status. The receiver sets a pass/fail bit for each of three phases and the transmitter will proceed to the next phase upon detection of pass only such that when a pass is detected, the main link is ready at that link rate. At this point, the interface transitions to a normal operation state 2510, otherwise, the link rate is reduced and the training session is repeated. During the normal operation state 2510, the source continues to periodically monitor a link status index, which if fails, a hot unplug event is detected and the system transitions to the standby state 2504 and waits for a hot plug detection event. If, however, a sync loss is detected, then the system transitions to state 2508 for a main link re-initiation event.

[0073] Fig. 26 shows the display state diagram 2600 described below. At a state 2602, no voltage is detected, the display goes to an OFF state. At a standby mode state 2604, both main link receiver and auxiliary channel slave are in electrical idle, a voltage drop across the termination resistors of auxiliary channel slave port are monitored for a predetermined voltage. If the voltage is detected, then the auxiliary channel slave port is turned on indicating a hot plug event and the system moves to a display state 2606, otherwise, the display remains in the standby state 2604. At the state 2606 (main link initialization phase), if a display is detected, then the auxiliary slave port is fully turned on, and the transmitter responds to a receiver link capability read command and the display state transitions to 2608, otherwise, if there is no activity on the auxiliary channel for more than a predetermined period of time then the auxiliary channel slave port is put into the standby state 2604.

[0074] During a training start notification phase, the display responds to the training initiation by the transmitter by adjusting the equalizer using training patterns, updating the result for each phase. If the training fails, then wait for another training session and if the training passes, then go to normal operation state 2610. If there is no activity on the auxiliary channel or on the main link (for training) for more than a predetermined (10ms, for example), the auxiliary channel slave port is set to the standby state 2604.

[0075] Figs. 27 _ 31 show particular implementations of the cross platform display interface.

[0076] Fig. 27 shows a PC motherboard 2000 having an on-board graphics engine 2002 that incorporates a transmitter 2704 in accordance with the invention. It should be noted that the transmitter 2704 is a particular example of the transmitter 102 shown in Fig. 1. In the described embodiment, the transmitter 2704 is coupled to an connector 2706 (along the lines of the connector 2400) mounted on the motherboard 2000 which in turn is connected to a display device 2708 by way of a twisted pair cable 2710 couples a display device 2710.

[0077] As known in the art, PCI Express (developed by Intel Corporation of Santa Clara, CA) is a high-bandwidth, low pin count, serial, interconnect technology that also maintains software compatibility with existing PCI infrastructure. In this configuration, the PCI Express port is augmented to become compliant with the requirements of the cross platform interface which can directly drive a display device either using a motherboard mounted connector as shown.

[0078] In situations where it is not practical to mount the connector on the motherboard, the signals can be routed through the SDVO slot of the PCI Express motherboard and brought to the back of the PC using a passive card connector as shown in Fig. 28. As is the case with the current generation of add-in graphics cards, an add-in graphics card can supplant the onboard graphics engine as shown in Fig. 30.

[0079] In the case of notebook applications, the transmitter on the motherboard graphics engine would drive through internal cabling, an integrated receiver/TCON which would drive the panel directly. For the most cost effective implementation, the receiver/TCON would be mounted on the panel thereby reducing the number of interconnect wires to 8 or 10 as shown in Fig. 31

[0080] All of the above examples assume integrated transmitters. However, it is quite feasible to implement as a standalone transmitter integrating into PCI and PCI Express environments through the AGP or SDVO slots, respectively. A standalone transmitter will enable output streams without any change in graphics hardware or software.

FLOWCHART EMBODIMENTS

[0081] The methodology of the invention will now be described in terms of a number of flowcharts each describing a particular process for enabling the invention. Specifically, FIGS 32 _ 36 describe a number of interrelated processes that when used singly or in any combination described aspects of the invention.

[0082] Fig. 32 shows a flowchart detailing a process 3200 for determining an operational mode of the interface 100 in accordance with an embodiment of the invention. In this process, the operational mode will only be set to a digital mode if the video source and the display device are both digital. Otherwise, the operational mode will be set to analog mode. It should be noted that "analog mode" in this context can include both conventional VGA mode as well as enhanced analog mode having differential analog video with embedded alignment signal and bidirectional sideband. This enhanced analog mode will be described below.

[0083] In step 3202, a video source is interrogated to determine whether the video source supports analog or digital data. If the video source supports only analog data, the operational mode of coupling device 100 will be set to analog (step 3208), then the process will end (step 3212).

[0084] If the video source can output digital data, the process continues to step 3206. The display device is then interrogated to determine whether the display device is configured to receive digital data. If the display device supports only analog data, the operational mode of coupling device will be set to analog (step 3208), then the process will end (step 3212). Otherwise, the operational mode of the coupling device is set to digital (step 3210). For example, a processor may control switches within the coupling device to set the mode to digital. In general, the coupling device is configured to operate in a fully digital mode only when both the video source and the video sink are operating in a corresponding digital mode.

[0085] Fig. 33 shows a flowchart detailing a process 3300 for providing a real time video image quality check in accordance with some aspects of the invention.

In this example, all determinations of process 3300 are made by a processor coupled to the display interface.

[0086] In step 3300, a video signal is received from a video source. Next, a signal quality test pattern is provided by the video source associated with the received video signal (step 3302). In step 3304, a determination of a bit error rate is made, based upon the quality test pattern. Then, a determination is made of whether the bit error rate is greater than a threshold value (step 3306). If the bit error rate is determined to not be greater than the threshold value, then a determination is made (step 3314) of whether or not there are more video frames. If it is determined that there are more video frames, then the process returns to step 3300. Otherwise, the process ends.

[0087] However, if the bit error rate is determined to be greater than the threshold value in step 3306, a determination is made (step 3308) as to whether the bit rate is greater than a minimum bit rate. If the bit rate is greater than a minimum bit rate, then the bit rate is lowered (step 3310) and the process returns to step 3306. If the bit rate is not greater than the minimum bit rate, then the mode is changed to analog mode (step 3312) and the process ends.

[0088] Fig. 34 shows a flowchart for a link set up process 3400 in accordance with an embodiment of the invention. The process 3400 begins at 3402 by the receiving of a hot plug detection event notification. At 3404 a main link inquiry is made by way of an associated auxiliary channel to determine a maximum data rate, a number of time base recovery units included in a receiver, and available buffer size. Next, at 3406, the maximum link data rate is verified by way of a training session and at 3408, a data stream source is notified of the hot plug event. At 3410, the capabilities of the display (using EDID, for example) are determined by way of the auxiliary channel and the display responds to the inquiry at 3412 which, in turn, results a collaboration of the main link training session at 3414.

[0089] Next, at 3416, the stream source sends stream attributes to the receiver by way of the auxiliary channel and at 3418, the stream sources are further notified whether the main link is capable of supporting the requested number of data streams at 3420. At 3422, the various data packets are formed by adding associated packet headers and the multiplexing of a number of source streams is scheduled at 3424. At 3426 a determination is made whether or not the link status is OK. When the link status is not OK, then the source(s) are notified of a link failure event at 3428, otherwise, the link data streams are reconstructed into the native streams based upon the various packet headers at 3430. At 3432, the reconstructed native data streams are then passed to the display device.

[0090] Fig. 35 shows a flowchart detailing a process 3500 for performing a training session in accordance with an embodiment of the invention. It should be noted that the training session process 3500 is one implementation of the operation 3206 described in Fig. 32. A training session is started at 3502 by sending a training pattern over the main link at a set link rate to the receiver. A typical link training pattern is shown in Fig. 11 in accordance with an embodiment of the invention. As illustrated, during the training session, a phase 1 represents the shortest run length while phase 2 is the longest. The receiver is to use these two phases to optimize the equalizer. In phase 3, both bit lock and character lock are achieved as long as the link quality is reasonable. At 3504, th

e receiver checks an associated training status and based upon the training status check, the receiver sets a pass/fail bit for each of three phases and the transmitter at 3506. At each phase, the receiver will proceed to the next phase upon detection of pass only and at 3510 and if the receiver does not detect a pass then the receiver reduces the link rate and repeats the training session. The main link is ready at that link rate at which a pass is detected at 3512.

[0091] Fig. 36 illustrates a computer system 3600 employed to implement the invention. Computer system 3600 is only an example of a graphics system in which the present invention can be implemented. Computer system 3600 includes central processing unit (CPU) 3610, random access memory (RAM) 3620, read only memory (ROM) 3625, one or more peripherals 3630, graphics controller 3660, primary storage devices 3640 and 3650, and digital display unit 3670. As is well known in the art, ROM acts to transfer data and instructions uni-directionally to the CPUs 3610, while RAM is used typically to transfer data and instructions in a bi-directional manner. CPUs 3610 may generally include any number of processors. Both primary storage devices 3640 and 3650 may include any suitable computer-readable media. A secondary storage medium 880, which is typically a mass memory device, is also coupled bi-directionally to CPUs 3610 and provides additional data storage capacity. The mass memory device 880 is a computer-readable medium that may be used to store programs including computer code, data, and the like. Typically, mass memory device 880 is a storage medium such as a hard disk or a tape which generally slower than primary storage devices 3640, 3650. Mass memory storage device 880 may take the form of a magnetic or paper tape reader or some other well-known device. It will be appreciated that the information retained within the mass memory device 880, may, in appropriate cases, be incorporated in standard fashion as part of RAM 3620 as virtual memory.

[0092] CPUs 3610 are also coupled to one or more input/output devices 890 that may include, but are not limited to, devices such as video monitors, track balls, mice, keyboards, microphones, touch-sensitive displays, transducer card readers, magnetic or paper tape readers, tablets, styluses, voice or handwriting recognizers, or other well-known input devices such as, of course, other computers. Finally, CPUs 3610 optionally may be coupled to a computer or telecommunications network, e.g., an Internet network or an intranet network, using a network connection as shown generally at 3695. With such a network connection, it is contemplated that the CPUs 3610 might receive information from the network, or might output information to the network in the course of performing the above-described method steps. Such information, which is often represented as a sequence of instructions to be executed using CPUs 3610, may be received from and outputted to the network, for example, in the form of a computer data signal embodied in a carrier wave. The above-described devices and materials will be familiar to those of skill in the computer hardware and software arts.

[0093] Graphics controller 3660 generates analog image data and a corresponding reference signal, and provides both to digital display unit 3670. The analog image data can be generated, for example, based on pixel data received from CPU 3610 or from an external encode (not shown). In one embodiment, the analog image data is provided in RGB format and the reference signal includes the VSYNC and HSYNC signals well known in the art. However, it should be understood that the present invention can be implemented with analog image, data and/or reference signals in other formats. For example, analog image data can include video signal data also with a corresponding time reference signal.

[0094] Although only a few embodiments of the present invention have been described, it should be understood that the present invention may be embodied in many other specific forms without departing from the spirit or the scope of the present invention. The present examples are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope of the appended claims along with their full scope of equivalents.

[0095] While this invention has been described in terms of a preferred embodiment, there are alterations, permutations, and equivalents that fall within the scope of this invention. It should also be noted that there are many alternative ways of implementing both the process and apparatus of the present invention. It is therefore intended that the invention be interpreted as including all such alterations, permutations, and equivalents as fall within the true spirit and scope of the present invention.

3. BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a generalized representation of a cross platform display interface 100 in accordance with an embodiment of the invention.

Figs. 2A _ 2C illustrates a video interface system that is used to connect a video source and a video display unit in accordance with a number of embodiments of the invention.

Fig. 3 shows exemplary main link rates in accordance with an embodiment of the invention.

Fig. 4A shows a main link data packet in accordance with an embodiment of the invention.

Fig. 4B shows a main link packet header in accordance with an embodiment of the invention.

Fig. 5A shows a system arranged to provide sub-packet enclosure and multiple-packet multiplexing in accordance with an embodiment of the invention.

Fig. 5B shows another implementation of the system shown in Fig. 5A.

Fig. 6 shows a high-level diagram of the multiplexed main link stream as an example of the stream shown in Fig. 5.

Fig. 7 show another example of a data stream in accordance with the invention.

Fig. 8 shows yet another example of a multiplexed data stream in accordance with an embodiment of the invention.

Fig. 9A shows a representative sub-packet in accordance with an embodiment of the invention.

Fig. 9B shows a representative main link data packet in accordance with an embodiment of the invention.

Fig. 10 shows an example of a selectively refreshed graphics image.

Fig. 11 shows an exemplary link training pattern in accordance with an embodiment of the invention.

Fig. 12 shows a representative system having a data stream scheduler in accordance with an embodiment of the invention.

Fig. 13 shows a more detailed portion of the data stream shown in Fig. 12.

Fig. 14 illustrates the addition of a fourth data stream in accordance with an embodiment of the invention.

Fig. 15 shows a degenerate case of a single uncompressed data stream in accordance with an embodiment of the invention.

Fig. 16 shows a video display based upon the degenerate single data stream sho

wn in Fig. 15.

Fig. 17 shows a flowchart detailing a process for scheduling a number of data streams in accordance with an embodiment of the invention.

Fig. 18 shows a flowchart detailing a process for forming the link data stream in accordance with an embodiment of the invention.

Fig. 19 illustrates a logical layering of the system in accordance with an embodiment of the invention.

Fig. 20 shows an exemplary special character mapping using 8B/10B in accordance with an embodiment of the invention.

Fig. 21 shows an exemplary Manchester II encoding scheme in accordance with an embodiment of the invention.

Fig. 22 shows a representative auxiliary channel electrical sub layer in accordance with an embodiment of the invention.

Fig. 23 shows a representative main link electrical sub layer in accordance with an embodiment of the invention.

Fig. 24 shows a representative connector in accordance with an embodiment of the invention.

Fig. 25 shows a source state diagram in accordance with an embodiment of the invention.

Fig. 26 shows a display state diagram in accordance with an embodiment of the invention.

Figs. 27 _ 31 illustrate various computer based implementations of the invention.

Fig. 32 shows a flowchart detailing a process for determining an operational mode of the interface in accordance with an embodiment of the invention.

Fig. 33 shows a flowchart detailing a process for providing a real time video image quality check in accordance with some aspects of the invention.

Figs. 34A and 34B show a flowchart for a link set up process in accordance with an embodiment of the invention.

Fig. 35 shows a flowchart detailing a process for performing a training session in accordance with an embodiment of the invention.

Fig. 36 illustrates a computer system employed to implement the invention.

1. A packet based display interface arranged to couple a multimedia source device to a multimedia sink device, comprising:

- a transmitter unit coupled to the source device arranged to receive a source packet data stream in accordance with a native stream rate;

- a receiver unit coupled to the sink device;

- a linking unit coupling the transmitter unit and the receiver unit arranged to transfer a multimedia data packet stream formed of a number of multimedia data packets based upon the source packet data stream in accordance with a link rate that is independent of the native stream rate between the transmitter unit and the receiver unit; and

- a data packet scheduler unit coupled to the linking unit arranged to schedule data packets for transport over the linking unit for the selected ones of the data streams wherein the data packet size for each data stream is an associated fixed size that depends upon a ratio between a data stream bit rate and a link bit rate.

2. A display interface as recited in claim 1, wherein the multimedia data packet stream is one of a number of multimedia data packet streams each having an

associated adjustable data stream link rate that is independent of the native stream rate.

3. A display interface as recited in claim 1, wherein the link unit further comprises:

a unidirectional main link arranged to carry the multimedia data packets from the transmitter unit to the receiver unit; and

a bi-directional auxiliary channel arranged to transfer information between the transmitter unit and the receiver unit and vice versa.

4. A display interface as recited in claim 3, wherein the bi-directional auxiliary channel is formed of a uni-directional back channel configured to carry information from the sink device to the source device and a uni-directional forward channel included as part of the main channel for carrying information from the source device to the sink device in concert with the back channel.

5. A display interface as recited in claim 2, wherein the main link unit further comprises:

a number of virtual links each being associated with a particular one of the multimedia data packet streams wherein each of said virtual links has an associated virtual link bandwidth and a virtual link rate.

6. A display interface as recited in claim 5, wherein a main link bandwidth is at least equal to an aggregate of the virtual link bandwidths.

7. A method of coupling a multimedia source device to a multimedia sink device, comprising:

providing a source device having a transmitter unit coupled thereto;

providing sink device having a receiver unit coupled thereto;

receiving a source data stream in accordance with a native stream rate by the transmitter unit;

coupling the transmitter unit and the receiver unit by way of a linking unit;

forming a multimedia data packet stream formed of a number of multimedia data packets;

generating a transport schedule for transferring the multimedia data packet stream in accordance with a link rate between the transmitter unit and the receiver unit wherein the multimedia data packets are each a fixed size based upon the link rate and a data stream bit rate.

8. A method as recited in claim 7, further comprising:

providing a unidirectional main link arranged to carry the multimedia data packets from the transmitter unit to the receiver unit; and

providing a bi-directional auxiliary channel arranged to transfer information between the transmitter unit and the receiver unit and vice versa.

9. A method as recited in claim 8, wherein the bi-directional auxiliary channel is formed of a uni-directional back channel configured to carry information from the sink device to the source device and a uni-directional forward channel included as part of the main channel for carrying information from the source device to the sink device in concert with the back channel.

10. A method as recited in claim 9, wherein the main link unit further comprises:

a number of virtual links each being associated with a particular one of the multimedia data packet streams wherein each of said virtual links has an associated virtual link bandwidth and a virtual link rate.

11. A method as recited in claim 10, wherein a main link bandwidth is at least equal to an aggregate of the virtual link bandwidths.

12. A method of scheduling a transport of a number of data packets between a data source and a data sink by way of a data link, comprising:

sending data packet attributes from the data packet source to the data packet sink;

comparing a stream bit rate to a data link bit rate for each of a number of data streams to be sent from the source to the sink;

setting a packet size for each of the data streams based upon the comparing wherein the packet size is a fixed packet size;

combining at least one of each data packet; and

transporting the combined data packets from the source to the sink.

13. Computer program product for scheduling a transport of a number of data packets between a data source and a data sink by way of a data link, comprising:

computer code for sending data packet attributes from the data packet source to the data packet sink;

computer code for comparing a stream bit rate to a data link bit rate for each of a number of data streams to be sent from the source to the sink;

computer code for setting a packet size for each of the data streams based upon the comparing wherein the packet size is a fixed packet size;

computer code for combining at least one of each data packet;

computer code for transporting the combined data packets from the source to the sink; and

computer readable medium for storing the code.

1. ABSTRACT

A method of coupling a multimedia source device to a multimedia sink device by providing a source device having a transmitter unit coupled thereto, providing a sink device having a receiver unit coupled thereto, receiving a source data stream in accordance with a native stream rate by the transmitter unit, coupling the transmitter unit and the receiver unit by way of a linking unit, forming a multimedia data packet stream formed of a number of multimedia data packets and generating a transport schedule for transferring the multimedia data packet stream in accordance with a link rate between the transmitter unit and the receiver unit wherein the multimedia data

2. REPRESENTATIVE DRAWING

Fig. 1

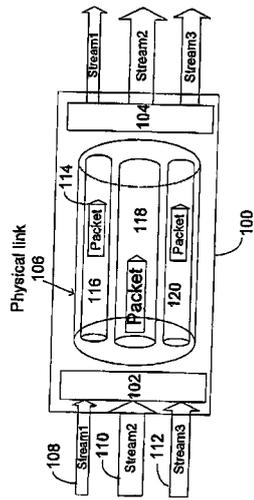


Fig. 1

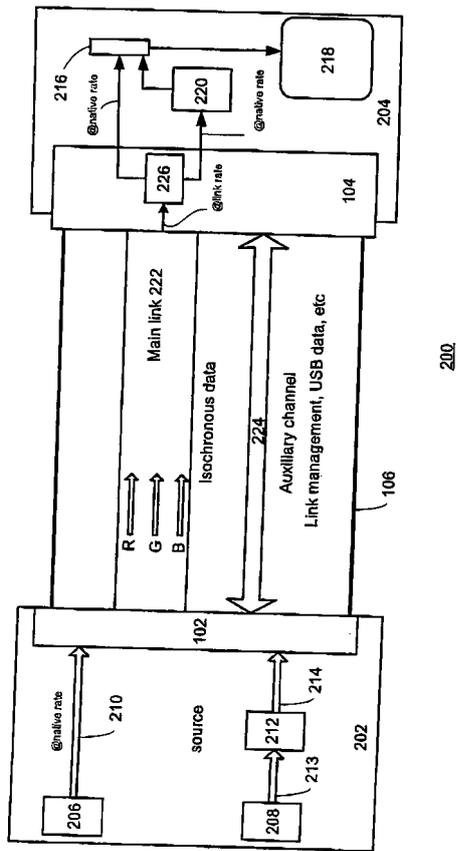


Fig. 2A

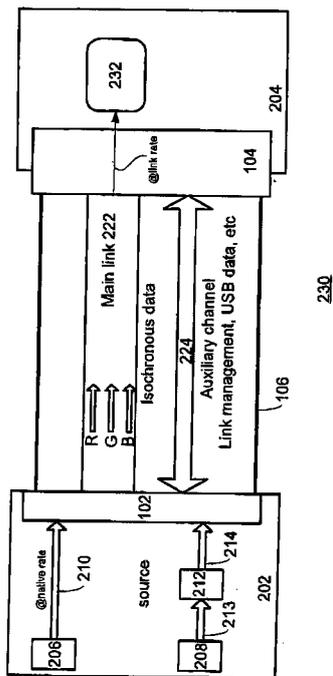


Fig. 2B

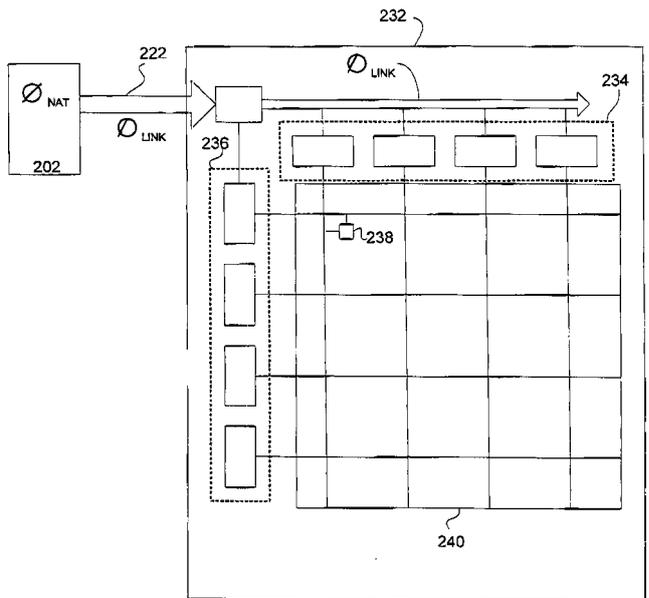


Fig. 2C

Main Link Data Rates

Nominal Baud Rate per channel (Gbits/second)	Actual Baud Rate per channel (Gbits/second)	Clock Multiplication Factor from 24-MHz crystal
1.0	0.960	x40
1.35	1.344	x56
1.7	1.728	x72
2.1	2.112	x88
2.5	2.496	x104

Fig. 3

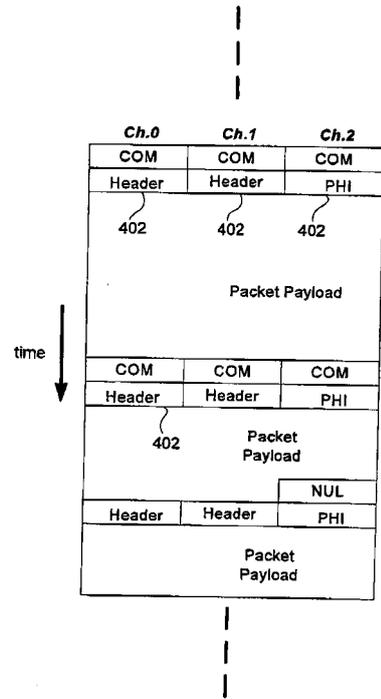


Fig. 4A

Main Link Packet Format

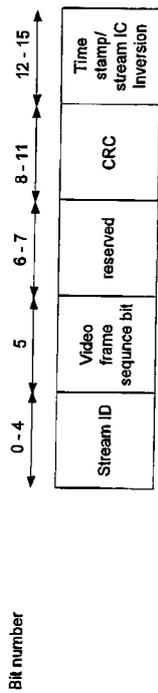


Fig. 4B

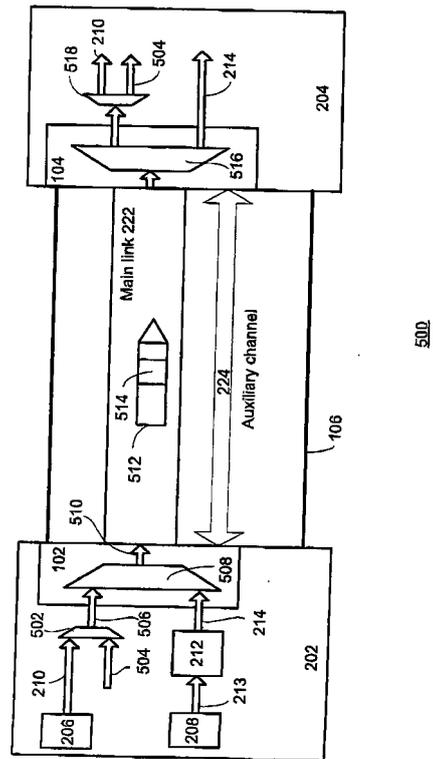
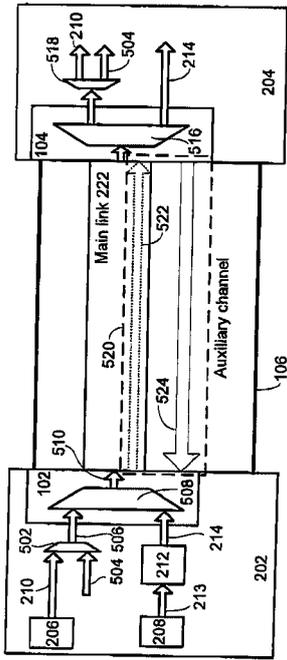


Fig. 5A



500

Fig. 5B

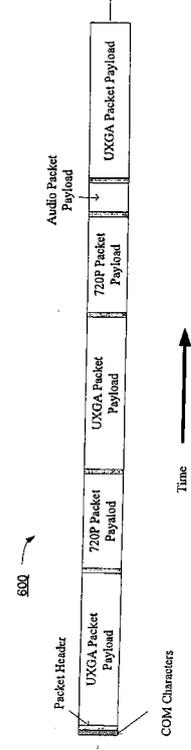


Fig. 6

High-level diagram of link traffic example with three streams

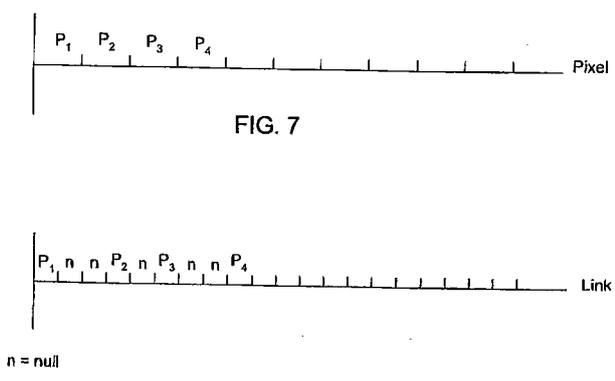


FIG. 7

FIG. 8

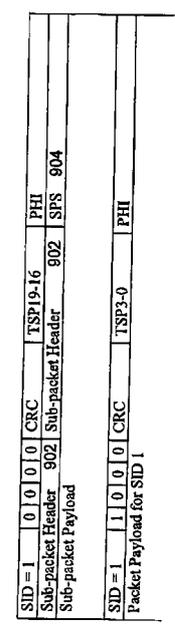


Fig. 9A

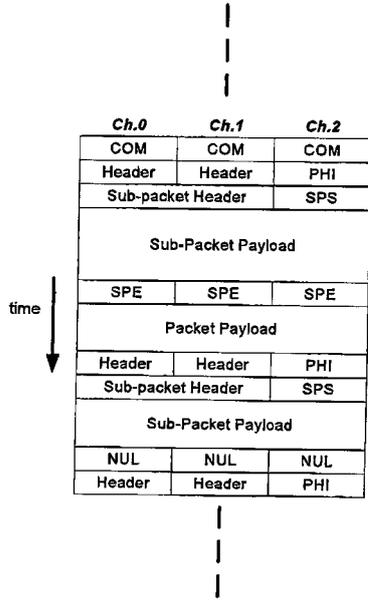


Fig. 9B

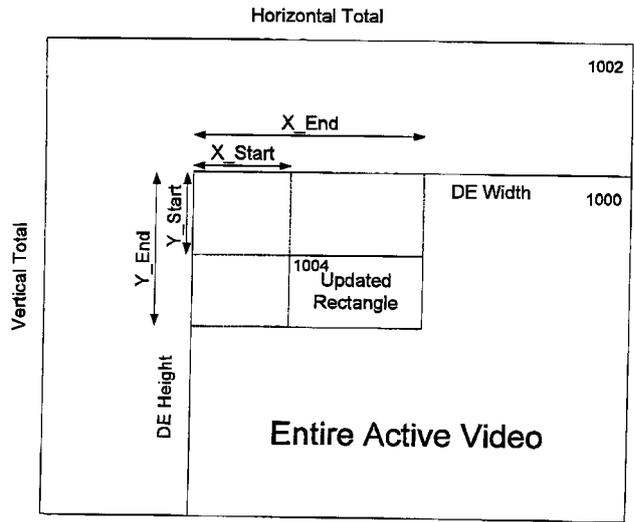


Fig. 10

Phase	Transmitted Link Characters	Binary pattern
1	D10.2	0101010101 0101010101 0101010101 0101010101 0101010101
2	K28.7	0011111000 0011111000 0011111000 0011111000 0011111000
3	K28.5, and three D10.2	0011111010 0101010101 0101010101 0101010101 1100000101

Main Link Training Pattern

Fig. 11

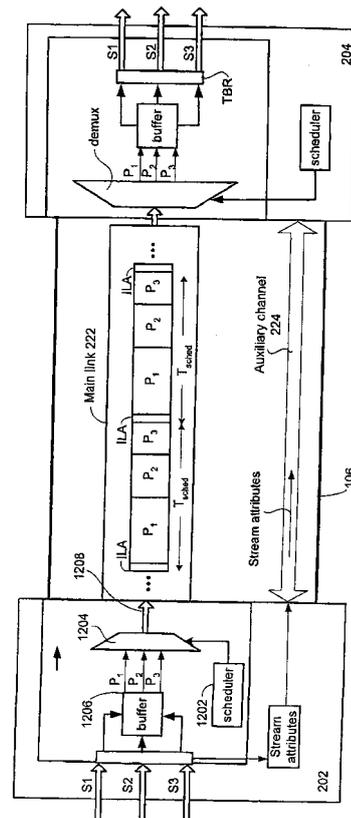


Fig. 12

1200

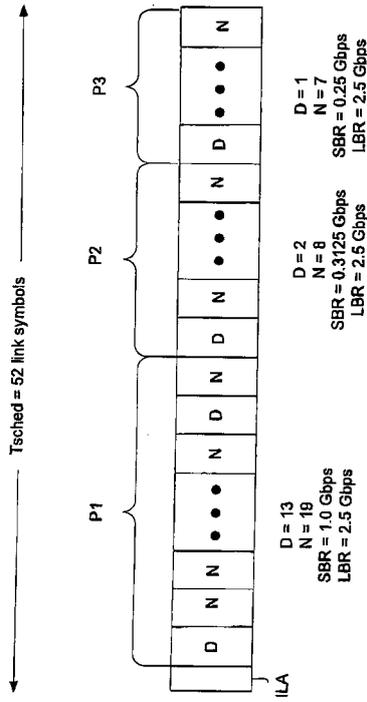


Fig. 13

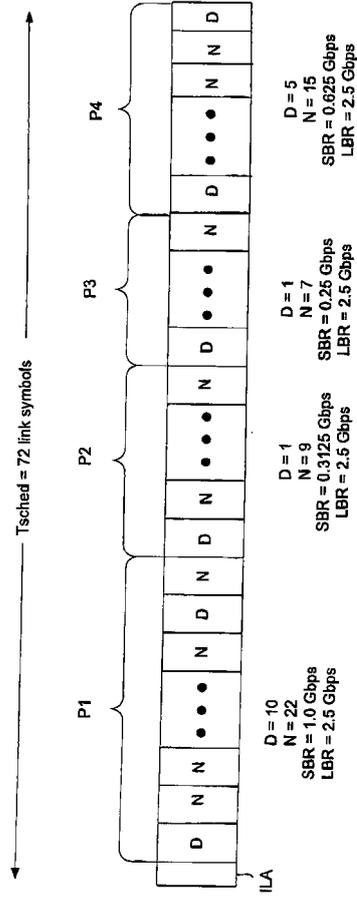


Fig. 14

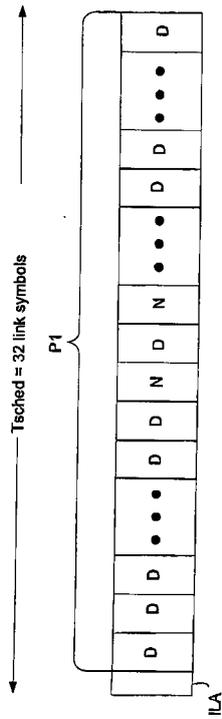


Fig. 15

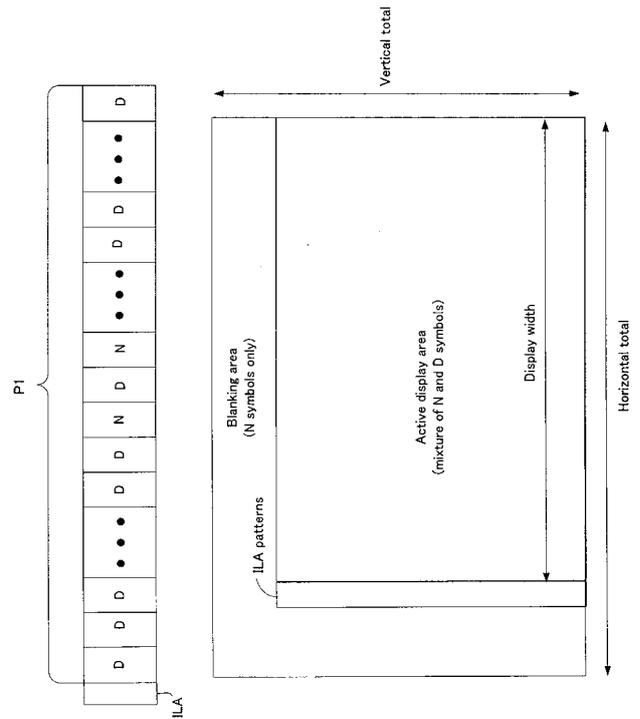


Fig. 16

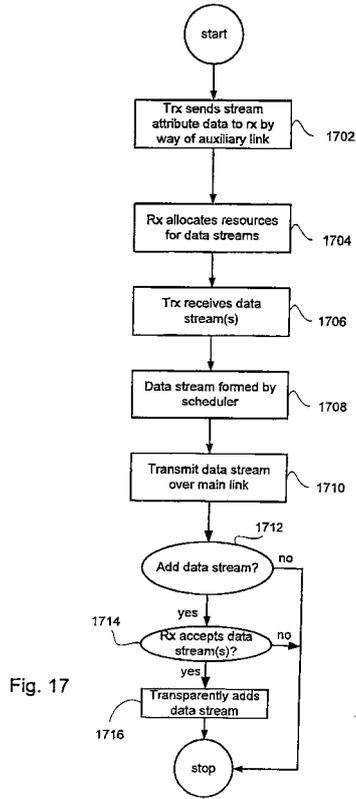


Fig. 17

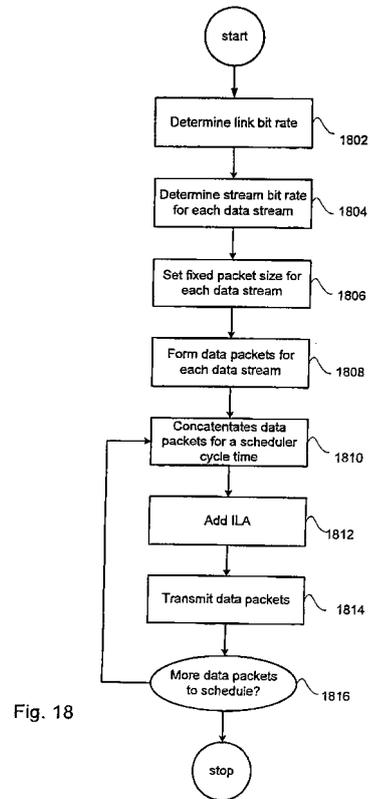


Fig. 18

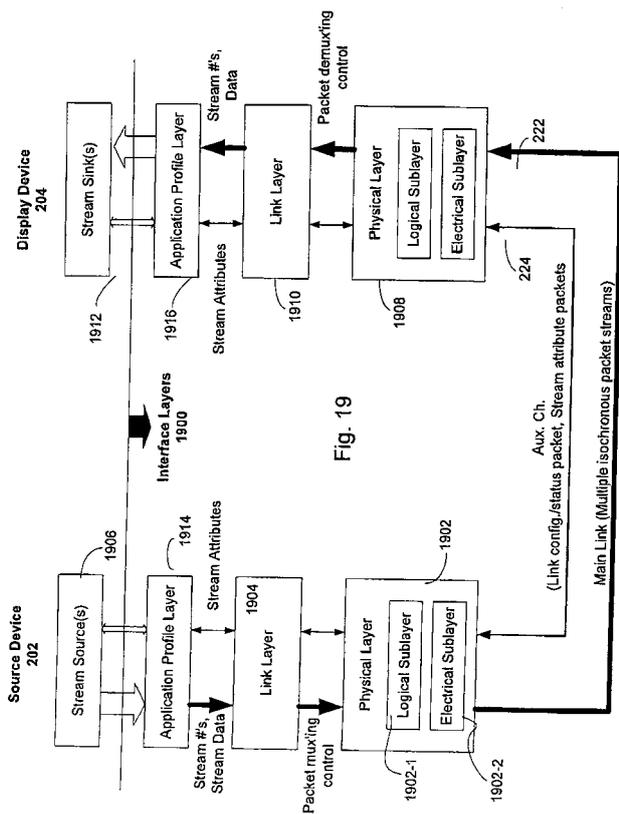


Fig. 19

8B/10B Special Characters Usage

Encoding	Name	Description
K28.5	Comma (COM)	Inserted between packets. Also used as part of Test Pattern
K28.7	Training Pattern (TPN)	Sent during Training Pattern transmission for bit/byte clock lock.
K23.7	Null (NUL)	Sent within the packet period when there is no data to transmit.
K28.2	Sub-packet Start (SPS)	Indicate a start of sub-packet inserted in a packet
K29.7	Sub-packet End (SPE)	Indicate an end of sub-packet inserted in a main packet.
K28.0	PacketHeaderIndicator (PHI)	Sent along with 16 bits of header for header identification.
K28.1	Reserved	Reserved
K28.3	Reserved	Reserved
K28.4	Reserved	Reserved
K28.6	Reserved	Reserved
K25.7	Reserved	Reserved
K27.7	Reserved	Reserved

Fig. 20

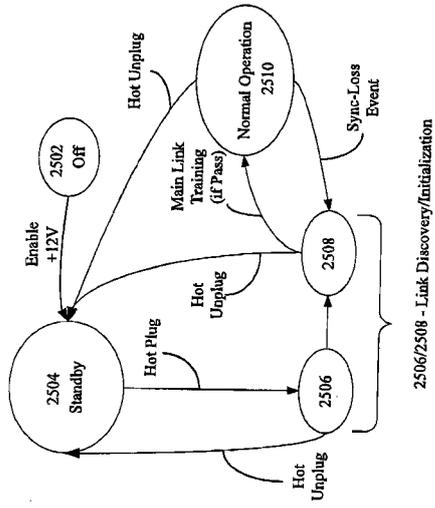


Fig. 25

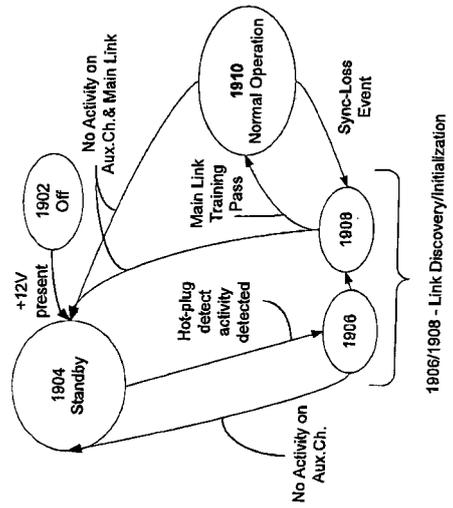
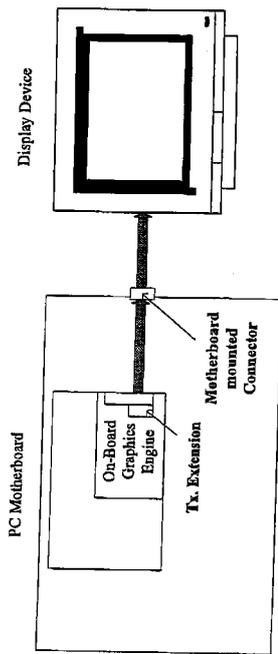
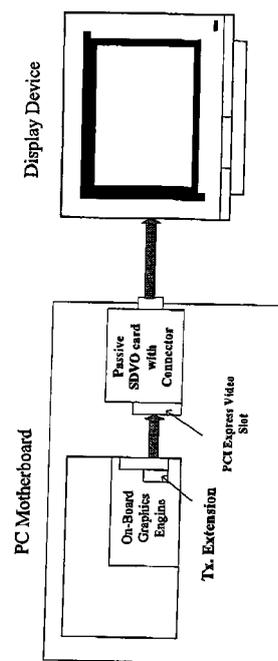


Fig. 26



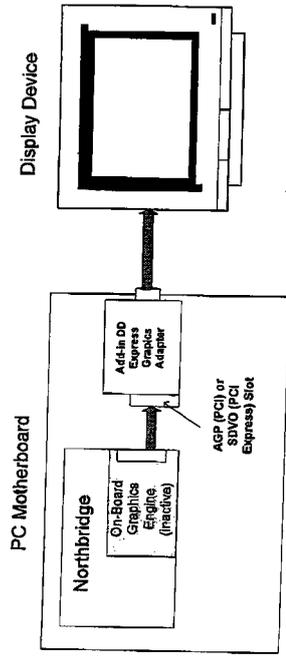
PCI EXPRESS MOTHERBOARD WITH DEDICATED DD- EXPRESS CONNECTOR

Fig. 27



PCI Express motherboard with passive connector card.

Fig. 28



PCI Express motherboard with add-in DD-Express graphics card

Fig. 29

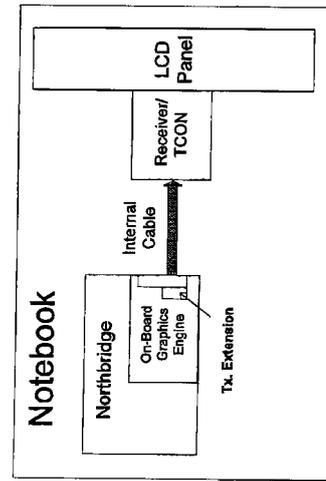
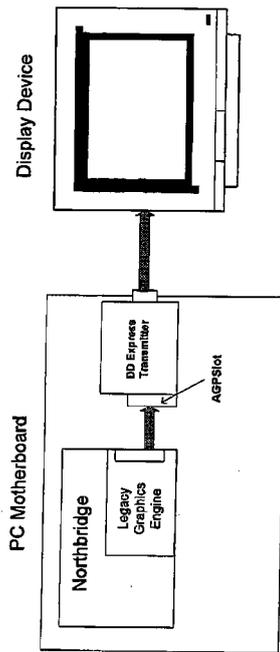


Fig. 30



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

Fig. 31

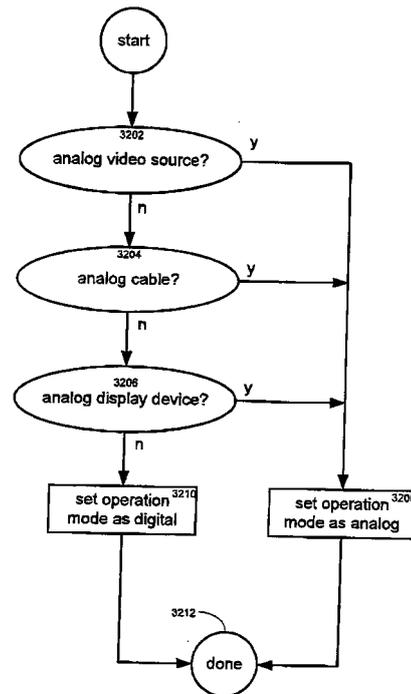


Fig. 32

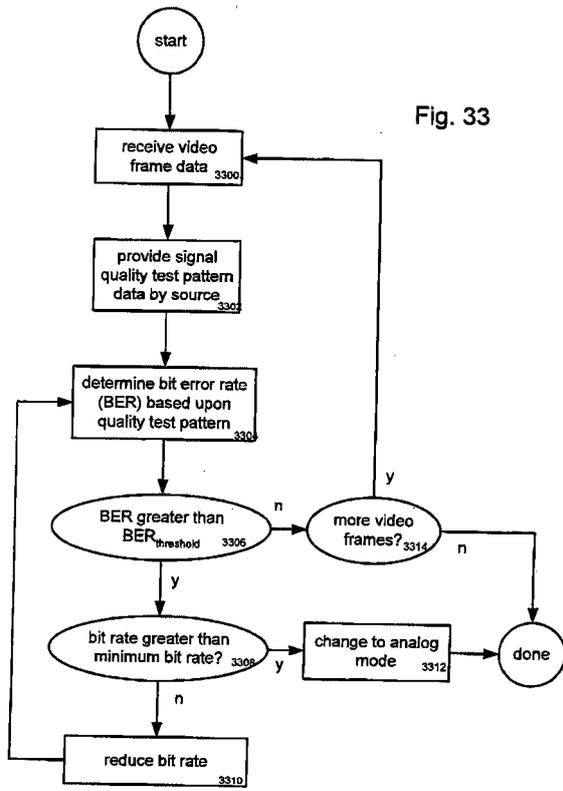


Fig. 33

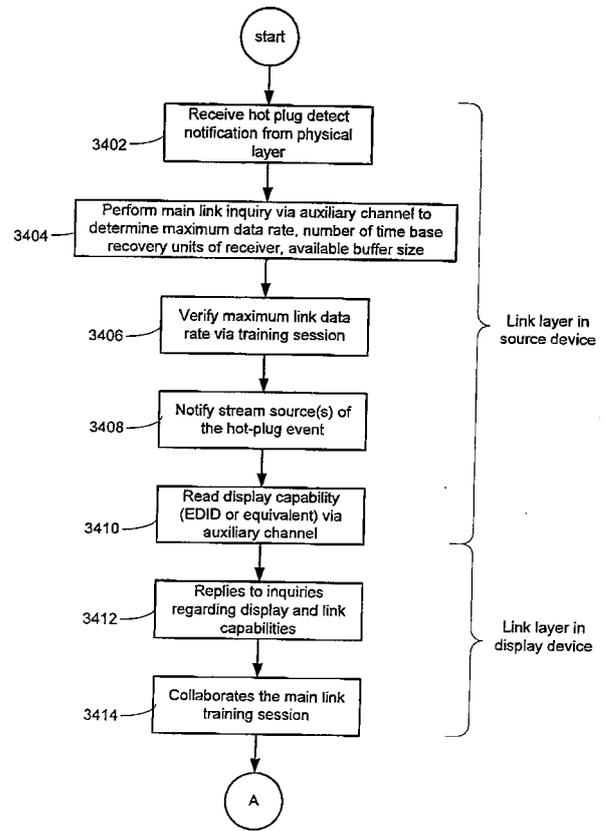


Fig. 34A

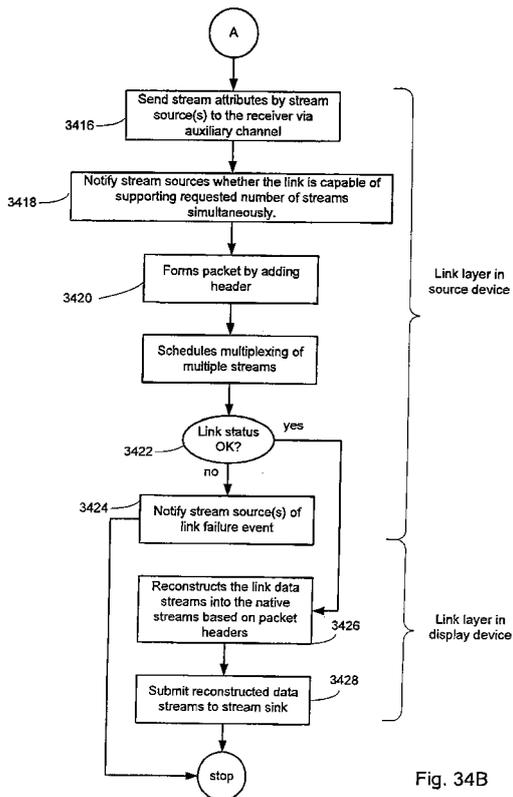


Fig. 34B

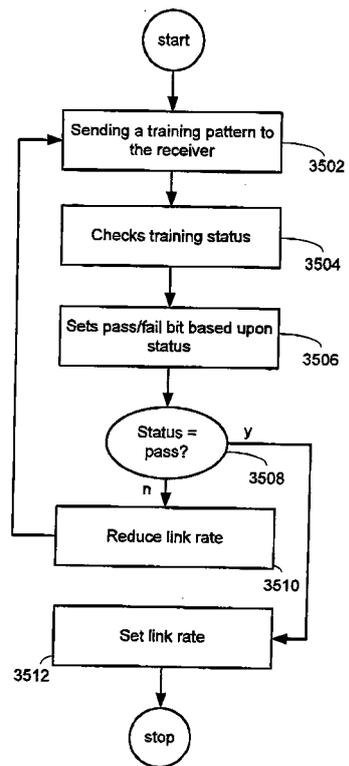


Fig. 35

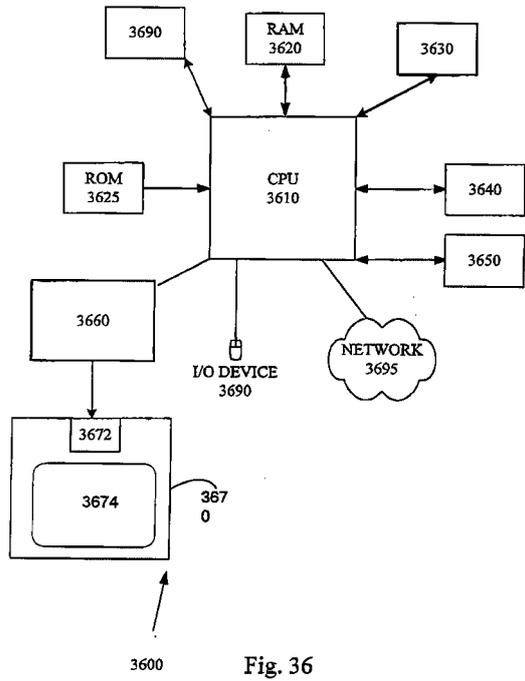


Fig. 36