

(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2007-267392

(P2007-267392A)

(43) 公開日 平成19年10月11日(2007.10.11)

(51) Int. Cl.		F I		テーマコード (参考)
<b>H04L 1/00 (2006.01)</b>		H04L 1/00	A	5K014
<b>H04L 12/56 (2006.01)</b>		H04L 12/56	Z	5K030

審査請求 未請求 請求項の数 19 O L 外国語出願 (全 51 頁)

(21) 出願番号	特願2007-101084 (P2007-101084)	(71) 出願人	504441048
(22) 出願日	平成19年3月9日 (2007.3.9)		シリコン イメージ, インコーポレイテッド
(31) 優先権主張番号	11/372,866		アメリカ合衆国, カリフォルニア州 94085, サニーヴェイル, イースト・アーケス・アヴェニュー 1060
(32) 優先日	平成18年3月9日 (2006.3.9)	(74) 代理人	100082005
(33) 優先権主張国	米国 (US)		弁理士 熊倉 禎男
(特許庁注: 以下のものは登録商標)		(74) 代理人	100067013
1. イーサネット			弁理士 大塚 文昭
		(74) 代理人	100086771
			弁理士 西島 孝喜
		(74) 代理人	100109070
			弁理士 須田 洋之
		最終頁に続く	

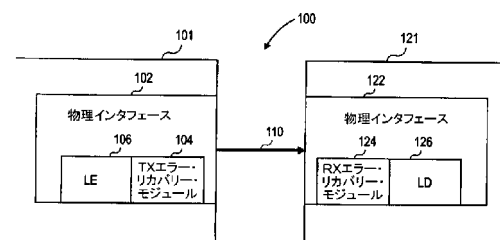
(54) 【発明の名称】 集積回路間のポイントツーポイント通信のための物理インタフェースでのエラー検出

## (57) 【要約】

【課題】 電子装置間の確実なデータ通信をもたらす機器及び方法を提供する。

【解決手段】 集積回路間のデータ通信の送信及び／又は受信中の物理インタフェースにおけるエラーを検出する機器、システム、及び方法。機器は、受信物理インタフェースとして又はその内部で作動するように構成され、符号化データビットの部分集合を復号化して復号化データビットを生成するように構成された復号器を含み、復号化データビットから物理インタフェースエラー検出ビットを抽出するように構成された物理インタフェースエラー検出ビット抽出器を含む。機器は、物理インタフェースエラー検出ビットを使用して符号化データビットが少なくとも1つの誤ったデータビットをエラーとして含むか否かを判断し、物理層内で作動するように構成されたエラー検出器を含み、例えば  $NB / (N + 1)$  Bライン符号器内でエラー検出コードを効率的に送信する。

【選択図】 図1A



## 【特許請求の範囲】

## 【請求項 1】

集積回路（ＩＣ）間のデータ通信を容易にする、物理インタフェースにおけるエラーを検出するための機器であって、

符号化データビットの部分集合を復号化して復号化データビットを生成するように構成された復号器と、

前記復号化データビットから、前記符号化データビットが少なくとも１つの誤ったデータビットをエラーとして含むか否かを判断するのに使用される物理インタフェースエラー検出ビットを抽出するように構成された物理インタフェース（ＰＩ）エラー検出ビット抽出器と、

10

を含むことを特徴とする機器。

## 【請求項 2】

前記エラーを検出するために物理層内で作動するように構成されたエラー検出器、

を更に含み、

前記符号化データビットの前記部分集合は、組込クロックを含む、

ことを特徴とする請求項 1 に記載の機器。

## 【請求項 3】

前記エラー検出器は、前記物理層の上方の「開放型システム相互接続（ＯＳＩ）」７層基準モデルのデータリンク層又はそのいかなる上部層におけるよりも早期にエラー検出を実行することを特徴とする請求項 2 に記載の機器。

20

## 【請求項 4】

前記物理インタフェースエラー検出ビットは、パリティビットであり、前記エラー検出器は、前記符号化データビットのパリティを検査して該パリティが不適合の場合にエラーを検出するように構成されていることを特徴とする請求項 2 に記載の機器。

## 【請求項 5】

前記物理インタフェースエラー検出ビットは、エラー検出コードの一部であり、

前記エラー検出コードは、巡回冗長検査（ＣＲＣ）チェックサムであり、前記エラー検出器は、前記物理インタフェースエラー検出ビットと、前記符号化データビットの他の部分集合からの他の物理インタフェースエラー検出ビットとを蓄積して該エラー検出コードを再構成するように構成されている、

30

ことを特徴とする請求項 2 に記載の機器。

## 【請求項 6】

N をデータビットのあらゆる整数を表すものとして、

（１）N 個のアプリケーションデータビットと共に物理インタフェースエラー検出ビットを挿入して N + 1 個の未符号化データビットを形成し、かつ

（２）前記 N + 1 個の未符号化データビットを符号化して N + 2 個の符号化データビットを生成する、

ように構成された N ビットから N + 2（N ビット / N + 2）ビット物理層（PHY）符号器と、

前記物理インタフェースエラー検出ビットを含むエラー検出コードを構成するいくつかのビットを発生するように構成されたエラー検出コード発生器と、

40

を更に含むことを特徴とする請求項 1 に記載の機器。

## 【請求項 7】

物理的媒体を通じて第 2 の IC に結合された第 1 の集積回路（IC）、

を含み、

前記第 1 の IC は、該第 1 の IC の第 1 の基板部分上に形成された物理インタフェースを含んで第 2 の基板部分上に形成された前記第 2 の IC と交換されたデータ内の送信エラーを検出し、

符号化データビットの部分集合を復号化して復号化データビットを生成するように構成された復号器と、

50

前記復号化データビットから、前記符号化データビットが少なくとも1つの誤ったデータビットをエラーとして含むか否かを判断するのに使用される物理インタフェースエラー検出ビットを抽出するように構成された物理インタフェースエラー検出ビット抽出器と、  
を更に含むことを特徴とするシステム。

【請求項8】

前記物理インタフェースは、  
着信符号化データビットを受信するように構成された入力ポートの第1の部分集合と、  
着信復号化データビットを前記別のICに送信するように構成された出力ポートの第1の部分集合とを含む複数の入力ポート及び出力ポートと、

前記複数の入力ポート及び出力ポートの間に結合された1つ又はそれよりも多くのエラー・リカバリー・モジュールと、  
を含み、

前記1つ又はそれよりも多くのエラー・リカバリー・モジュールの第1のエラー・リカバリー・モジュールは、前記第1の部分集合の入力ポートの少なくとも1つと前記第1の部分集合の出力ポートの少なくとも1つとの間に結合されている、

ことを特徴とする請求項7に記載のシステム。

【請求項9】

前記第1のエラー・リカバリー・モジュールは、前記着信符号化データビット内のエラーを検出して該エラーを訂正するアクションを開始するように構成された物理層(PHY)復号器を含むことを特徴とする請求項8に記載のシステム。

【請求項10】

前記複数の入力ポート及び出力ポートは、発信未符号化データビットを前記別のICから受信するように構成された入力ポートの第2の部分集合と、発信符号化データビットを送信するように構成された出力ポートの第2の部分集合とを含み、

前記1つ又はそれよりも多くのエラー・リカバリー・モジュールの第2のエラー・リカバリー・モジュールは、前記第2の部分集合の入力ポートの少なくとも1つと前記第2の部分集合の出力ポートの少なくとも1つとの間に結合され、該第2のエラー・リカバリー・モジュールは、前記発信符号化データビットに対するエラー検出コードを発生させるように構成されたPHY符号器を含む、

ことを特徴とする請求項9に記載のシステム。

【請求項11】

符号化データビットを発生させ、該符号化データビットには物理インタフェースエラー検出ビットが符号化されたNビット/N+2ビット符号器を含み、かつ前記物理的媒体の第1の端部に結合された送信装置と、

前記符号化データビットを復号化するN+2ビット/Nビット復号器、及び

前記物理インタフェースエラー検出ビットを使用してエラーを判断するように構成されたエラー検出器、

を含み、かつ前記物理的媒体の第2の端部に結合された受信装置と、

を更に含むことを特徴とする請求項7に記載のシステム。

【請求項12】

前記送信装置は、遷移最小化差動信号伝達(TMD S)送信器及び高品位マルチメディアインタフェース(HDMI)送信器であり、前記受信装置は、TMD S受信器及びHDMI受信器であることを特徴とする請求項11に記載のシステム。

【請求項13】

データビットを復号化して物理インタフェースにおいて少なくともエラーを検出する方法であって、

組込非同期クロックを有する符号化ビットストリームの部分集合を復号化して復号化データビットを生成する段階と、

前記復号化データビットから物理インタフェースエラー検出ビットを抽出する段階と、

少なくとも前記物理インタフェースエラー検出ビットに基づいて、前記符号化ビットス

10

20

30

40

50

トリームが正しくないビットを含むことを判断する段階と、  
前記物理インタフェースにおいて前記エラーを訂正する段階と、  
を含むことを特徴とする方法。

【請求項 14】

前記符号化ビットストリームの前記部分集合が前記正しくないビットを含むことを判断する段階は、

前記物理インタフェースエラー検出ビットを用いて前記符号化ビットストリームの部分集合のパリティを判断する段階と、

前記パリティが所定のパリティと適合しない場合に前記エラーが存在することを示す段階と、

を更に含む、

ことを特徴とする請求項 13 に記載の方法。

【請求項 15】

前記ビットストリームの前記部分集合が前記正しくないビットを含むことを判断する段階は、

前記物理インタフェースエラー検出ビットを他の物理インタフェースエラー検出ビットと組み合わせて第 1 のエラー検出コードを形成する段階と、

前記復号化データビットからのアプリケーションデータビットと、前記ビットストリームの他の部分集合から復号化された他の復号化データビットからの他のアプリケーションデータビットとに基づいて、第 2 のエラー検出コードを発生させる段階と、

前記第 1 のエラー検出コードを前記第 2 のエラー検出コードに対して比較する段階と、

前記第 2 のエラー検出コードが前記第 1 のエラー検出コードと適合しない場合に前記エラーが存在することを示す段階と、

を含む、

ことを特徴とする請求項 13 に記載の方法。

【請求項 16】

集積回路 (IC) 間のデータ通信の送信のために物理インタフェースでエラー検出コードを発生させるための機器であって、

N をデータビットのあらゆる整数を表すものとして、

N 個のアプリケーションデータビットと共に物理インタフェースエラー検出ビットを挿入して N + 1 個の未符号化データビットを形成し、かつ

前記 N + 1 個の未符号化データビットを符号化して N + 2 個の符号化データビットを生成する、

ように構成された N ビットから N + 2 (N ビット / N + 2) ビット物理層 (PHY) 符号器と、

前記物理インタフェースエラー検出ビットを含むエラー検出コードを構成するいくつかのビットを発生するように構成されたエラー検出コード発生器と、

を含むことを特徴とする機器。

【請求項 17】

第 1 の基板部分上に集積回路 (IC) として形成されて、第 2 の基板部分上に形成された別の IC と交換されたデータにおける送信エラーを検出する物理インタフェースであって、

着信符号化データビットを受信するように構成された入力ポートの第 1 の部分集合と、着信復号化データビットを別の IC に送信するように構成された出力ポートの第 1 の部分集合とを含む複数の入力ポート及び出力ポートと、

前記複数の入力ポート及び出力ポートの間に結合された 1 つ又はそれよりも多くのエラー・リカバリー・モジュールと、

を含み、

前記 1 つ又はそれよりも多くのエラー・リカバリー・モジュールの第 1 のエラー・リカバリー・モジュールは、前記第 1 の部分集合の入力ポートの少なくとも 1 つと前記第 1 の

10

20

30

40

50

部分集合の出力ポートの少なくとも１つとの間に結合されている、  
ことを特徴とする物理インタフェース。

【請求項１８】

少なくとも２つの集積回路（ＩＣ）間の高速シリアル通信リンクであって、  
通信データを伝達するための物理的媒体と、  
符号化データビットを発生させ、該符号化データビットには物理インタフェースエラー  
検出ビットが符号化された $N$ ビット/ $N+2$ ビット符号器を含み、かつ前記物理的媒体の  
第１の端部に結合された送信装置と、  
前記符号化データビットを復号化する $N+2$ ビット/ $N$ ビット復号器、及び  
前記物理インタフェースエラー検出ビットを使用してエラーを判断するように構成され  
たエラー検出器、  
を含み、かつ前記物理的媒体の第２の端部に結合された受信装置と、  
を含むことを特徴とする高速シリアル通信リンク。

【請求項１９】

パラレルデータリンクを通じてポイントツーポイント通信中に物理層において送信エラ  
ーを検出するための物理インタフェースであって、  
部分集合が物理インタフェースエラー検出ビットを含むパラレル通信データをパラレル  
データリンクを通じて受信するためのパラレルポートと、  
外部クロックを受信して前記パラレル通信データの前記受信を同期化するように構成さ  
れた外部クロック入力部と、  
前記物理インタフェースエラー検出ビットを使用して前記パラレル通信データが正しく  
ないビットをエラーとして含むか否かを判断するように構成された物理インタフェースエ  
ラー検出器と、  
物理インタフェースにおいて前記エラーを訂正するように構成されたエラー訂正器と、  
を含むことを特徴とする物理インタフェース。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、一般的に、電子装置間の確実なデータ通信を提供するための機器及び方法に  
関するものであり、より具体的には、電子装置間の物理的媒体上のデータビットのポイン  
トツーポイント伝送中に物理インタフェースでエラーを効率的に検出するように構成され  
た物理インタフェースに関する。本発明の一実施形態では、データビットのポイントツ  
ーポイント伝送は、シリアルデータリンクに沿った低振幅高周波数信号伝達を使用する。

【背景技術】

【０００２】

集積回路（ＩＣ）のような電子装置間のポイントツーポイント通信は、特にデータ転送  
の同期化に専用同期クロックラインが使用される時には一般的に確実であると考えられて  
いる。特に、多くのシステム設計担当者は、ＩＣ間の通信が、ＩＣ間で交換されるデー  
タビットが同じデータビットであるという広く知られた仮定に従うと予想している。例え  
ば、殆どの従来のプロセッサＩＣは、メモリＩＣに送信されたアドレスビットが正しく受信  
したか、又はプロセッサＩＣによって受信したデータがメモリＩＣによって送られたもの  
と同じであったか否かに関係なく、メモリＩＣからフェッチされたデータに作用するよう  
に設計されている。しかし、データビットを高速データ転送速度でかつ振幅を落として（  
すなわち、送信電力を落として）送信する従来の物理インタフェースでは、データ信号の  
強度に及ぼす背景ノイズの影響のためにＳＮ比が小さくなり、これは、確実なポイントツ  
ーポイント通信を妨げ兼ねないエラーの増加を示すものである。更に、多くのシステム設  
計担当者は、クロックが符号化されたシリアル化データに組み込まれる非同期クロッキ  
ング技術を使用してシリアルデータリンク上でデータをシリアル搬送する物理インタフェ  
ースを実施している。シリアルデータリンクを有する高速シリアル通信技術の例としては、  
「シリアルＡＴＡ（ＳＡＴＡ）（登録商標）」、「遷移最小化差動信号伝達（ＴＭＤＳ）」

」、「P C I E x p r e s s」、及び「I n f i n i B a n d（登録商標）」などがある。シリアルデータ通信に対して準備された従来の物理インタフェースは、ノイズ及び電力の両方を低減するが、少なくとも1つの欠点は、従来のシリアルデータリンクは、搬送中にデータ破損を受けやすいことである。

【0003】

コンピュータネットワーク分野において、「開放型システム相互接続（O S I）」モデルは、2つの端末局（又はコンピュータ装置）間のエラーを検出して訂正する一般的な手法を説明するものである。この手法によれば、データリンク層は、送信されたビットのフレームに基づいてエラー検出コード（例えば、巡回冗長コード、「C R C」）を計算する。次に、それは、連続ビットとしてエラー検出コードをフレームに添付し、次に、エラー検出コードとアプリケーションデータビットの両方をライン符号化（又はコード化）のために物理インタフェースに渡す。ライン符号化の一例は、8つのアプリケーションデータビットを10個の符号化ビットに変換するものである。この種のライン符号化技術は、一般的に8 B / 10 Bとして公知である。ライン符号化が行われた後、シリアル物理インタフェースは、通信媒体上で符号化ビットを別のシリアル物理インタフェースに追い出し、これが、クロックを回復してこれらのビットを復号化する。次に、受信器のデータリンク層は、エラー検出コードの連続ビットを使用してエラー検出及び訂正技術を適用する。

10

【0004】

データリンク層は、機能的ではあるが、これらのエラー検出及び訂正処理を物理インタフェースを超えて実行し、それによってエラー検出及び/又は解決を遅延させる。また、従来の物理インタフェースは、一般的に、エラー検出及び訂正を標準的方法で容易にするのに十分に適するものではない。例えば、従来の物理インタフェースは、集積回路間の同期及び並行データ通信を行うように最適化されている。従って、それらは、シリアルデータリンク上のような非同期及びシリアルデータ通信に対して適時かつ確実なデータ転送を行うように容易に適応させることができない。更に、従来の物理インタフェースは、一般的に、少なくとも（1）アプリケーションが特化エラー処理技術に参加する必要がある透明なエラー検出及び回復技術と、（2）エラー検出コードがシリアルデータリンク上で転送される速度を上げるためのエラー検出コードの効率的な送信とを提供することを助け

20

30

【0005】

【特許文献1】米国特許第6,747,580号

【発明の開示】

【発明が解決しようとする課題】

【0006】

以上に鑑みて、上述の欠点を最小にし、かつ少なくともエラー検出及び任意的なエラー・リカバリーを準備する物理インタフェース、機器、及び方法を提供することが望ましいと考えられる。

【課題を解決するための手段】

【0007】

集積回路（I C）間のデータ通信の送信及び/又は受信中の物理インタフェースにおけるエラーを検出するための機器、システム、及び方法を開示する。本発明の様々な実施形態によれば、物理インタフェースは、シリアル又はパラレルデータリンク上のいずれかの通信を容易にすることができ、又はそれは、組込非同期クロック又は外部同期クロックのいずれかを使用するように構成することができる。

40

【0008】

一実施形態では、機器は、受信物理インタフェースとして又はその内部で作動するように構成される。機器はまた、符号化データビットの部分集合を復号化して復号化データビットを生成するように構成された復号器を含む。それはまた、復号化データビットから物理インタフェースエラー検出ビットを抽出するように構成された物理インタフェース（P

50

I) エラー検出ビット抽出器を含む。従って、機器は、物理インタフェースエラー検出ビットを使用して、符号化データビットが少なくとも1つの誤ったデータビットをエラーとして含むか否かを判断する。一部の実施形態では、機器は、物理層内で作動するように構成されたエラー検出器を含む。少なくとも1つの実施形態では、機器は、例えば  $NB/(N+1)$  Bライン符号器内でエラー検出コードを効率的に送信する。符号化データビットは、実質的に直流(DC)均衡状態とすることができる。更に、符号化データビットは、放出放射線のスペクトル特性を形作るように構成することができる。また、これらの符号化データビットは、受信装置でクロックを回復するのに使用することができる組込非同期クロックを含むこともできる。一部の事例においては、機器は、物理層内で作動してエラーを検出するように構成されたエラー検出器を含む。エラー検出器は、「開放型システム相互接続(OSI)」7層基準モデルのデータリンク層又はあらゆる上部層におけるよりも早期にエラー検出を実行する。一部の 경우에는、物理インタフェースエラー検出ビットは、パリティビットであり、一方、他の事例では、物理インタフェースエラー検出ビットは、エラー検出コードの一部である。例えば、エラー検出コードは、巡回冗長検査(CRC)チェックサムとすることができる。この場合のエラー検出器は、物理インタフェースエラー検出ビット、並びに符号化データビットの他の部分集合(すなわち、他のコード語)からの他の物理インタフェースエラー検出ビットを蓄積してエラー検出コードを再構成するように構成される。エラー検出器は、物理インタフェースエラー検出ビット及び他の物理インタフェースエラー検出ビットからCRC検査を発生させるように構成することができることに注意されたい。次に、エラー検出器は、CRC検査をエラー検出コードと比較して、エラーが発生したか否かを判断することができる。

#### 【0009】

様々な実施形態では、物理インタフェースエラー検出ビットは、符号化データビットの部分集合と同時に機器で受信され、他の物理インタフェースエラー検出ビットの各々も、符号化データビットの他の部分集合の各々と同時に機器で受信され、それによってそうでなければ符号化データビットに追随するように添付されるエラー検出コードを機器が受信する時間を短縮する。少なくとも1つの実施形態では、機器はまた、エラーを訂正するように構成されたエラー訂正器を含む。エラー訂正器は、符号化データビットの部分集合の再送信を要求するか、又はエラーが検出されたことをアプリケーション特有のモジュールに信号で知らせ、それによってアプリケーション特有のモジュールがエラーを解決することができるように構成することができる。あるいは、エラー訂正器は、1つ又はそれよりも多くのICを再初期化するなどのデフォルト対策を取ることができる。符号化データビットの部分集合の1つは、 $N+m$ 個のデータビットを含むことができ、これを復号化して、復号化データビットとしてN個のアプリケーションデータビットを取得することができることに注意されたい。機器は、 $N+2$ 個のデータビットとして符号化データビットを受信し、かつN個のアプリケーションデータビットを出力として供給するように構成することができることに注意されたい。例えば、Nが8の時、機器は、 $8B/10B$ 復号器として構成され、それは、9個の復号化データビット、すなわち、8個のアプリケーションデータビット及びエラー検出のための1つの物理インタフェースエラー検出ビットを提供する。

#### 【0010】

本発明の別の実施形態では、集積回路(IC)間のデータ送信のために物理インタフェースにおいてエラー検出コードを発生させる機器を開示する。一実施形態では、機器は、送信物理インタフェースとして又はその内部で作動するように構成される。機器は、Nビットから $N+2$ (Nビット/ $N+2$ )ビットの物理層(PHY)符号器を含むことができる。この符号器は、物理インタフェースエラー検出ビットをN個のアプリケーションデータビットと共に挿入して $N+1$ 個の未符号化データビットを形成し、 $N+1$ 個の未符号化データビットを符号化して $N+2$ 個の符号化データビットを生成するように構成される。機器はまた、物理インタフェースエラー検出ビットを含む、エラー検出コードを構成するいくつかのビットを発生させるように構成されたエラー検出コード発生器を含む。Nは、

あらゆる整数のデータビットを表すことができる。機器は、一部の場合では、 $N + 2$  個の符号化データビットをシリアル化して  $N + 2$  個のシリアル符号化データビットを形成し、それによって単一のエラー検出データビットを  $N + 2$  個のシリアル符号化データビットと交互配置するように構成されたシリアルライザを含むことができる。少なくとも 1 つの実施形態では、機器は、 $N$  個のデータビットを含むデータビットの群を記憶する記憶装置を含む。エラー検出コード発生器は、データビットの群を検査してエラー検出コードを発生させるように構成される。特定の実施形態では、機器は、送信器 (TX) エラー訂正器を含み、これは、送信器エラー訂正器が再送信の要求を受信した時にデータビットの群の 1 つ又はそれよりも多くの部分集合を再送信するように構成されている。

#### 【0011】

10

本発明の更に別の実施形態では、物理インタフェースは、第 1 の基板上に集積回路 (IC) として形成され、第 2 の基板上に形成されたコア IC と交換されたデータにおける送信エラーを検出する。コア IC は、メモリチップの記憶機能のようなアプリケーション特有の機能を実施する論理を収容する別の IC である。物理インタフェースは、着信符号化データビットを受信するように構成された入力ポートの第 1 の部分集合と、着信復号化データビットをコア IC に送信するように構成された出力ポートの第 1 の部分集合とを含む複数の入力ポート及び出力ポートを含む。物理インタフェースはまた、複数の入力ポート及び出力ポートの間に結合された 1 つ又はそれよりも多くのエラー・リカバリー・モジュールを含む。エラー・リカバリー・モジュールの少なくとも 1 つは、入力ポートの第 1 の部分集合の少なくとも 1 つと出力ポートの第 1 の部分集合の少なくとも 1 つとの間に結合される。エラー・リカバリー・モジュールは、任意的に、着信符号化データビット内のエラーを検出してエラーを訂正するアクションを開始するように構成された物理層 (PHY) 復号器を含むことができる。

20

#### 【0012】

本発明の更に別の実施形態では、少なくとも 2 つの集積回路 (IC) 間の高速シリアル通信リンクを開示する。このリンクは、データを伝達するための物理的媒体と、物理的媒体の第 1 の端部に結合された送信装置と、物理的媒体の第 2 の端部に結合された受信装置とを含む。それはまた、物理インタフェースエラー検出ビットを使用してエラーを判断するように構成されたエラー検出器を含む。送信装置は、物理インタフェースエラー検出ビットを符号化した符号化データビットを発生させる  $N$  ビット /  $N + 2$  ビット符号器を含み、受信装置は、符号化データビットを復号化する  $N + 2$  ビット /  $N$  ビット復号器と、物理インタフェースエラー検出ビットを使用してエラーを判断するエラー検出器とを含むことに注意されたい。また、送信装置及び受信装置は、遷移最小化差動信号伝達 (TMDS) 装置、高品位マルチメディアインタフェース (HDMI) 装置、マルチギガビット「イーサネット」装置 (例えば、10 Gbps「イーサネット」又は「10 GbE」装置)、及び / 又はシリアル高度技術アタッチメント (SATA) 装置のいずれかの 1 つ又はそれよりも多くとすることができるとに注意されたい。また、送信装置及び受信装置は、「デジタル視覚インタフェース (DVI)」及び / 又は「低電圧差動信号伝達 (LVDS)」並びにあらゆる他の同等の技術を実施することができるとに注意されたい。

30

本発明は、添付図面と共に以下の詳細説明に関連してより完全に評価される。

40

同じ参照番号は、図面のいくつかの図を通して対応する部分を指している。参照番号の大半は、その参照番号を最初に導入する図を全体的に特定する最も左の 1 つ又は 2 つの桁を含むことに注意されたい。

#### 【発明を実施するための最良の形態】

#### 【0013】

図 1 A は、本発明の少なくとも 1 つの特定の実施形態によりエラー・リカバリー・モジュールを含んでライン符号化 / 復号化に関連してエラーを検出する物理インタフェースを示すブロック図 100 である。物理インタフェース 102 及び 122 は、装置 101 及び 121 の間の物理的接続性を提供するものであり、その各々は、メモリチップのような集積回路か、又は「イーサネット」ラインカード又はネットワーク化コンピュータのよう

50



な電子装置とすることができる。物理インタフェース102及び122は、物理的媒体110を通じてデータビットの送受信を容易にするために、それぞれ、ライン符号器(LE)106とライン復号器(LD)126を含む。一実施形態では、ライン符号器(LE)106は、ライン符号化(「ラインコード化」とも呼ばれる)を提供するものであり、これは、銅線、光ファイバ、無線波のような特定の種類の媒体上の符号化データビットの送信を最適化するための特定のコード(又はコード語)へのバイナリ(すなわち、データビット)で表されたデジタル情報の変換である。ライン復号化は、特定のコードからデータビットに変換する逆の処理である。ライン符号器及びライン復号器の両方は、特定の種類の媒体と共に最適に作動するように調整される。一部の実施形態において使用されるように、符号化という用語は、コード語へのデータビットの変換、並びにコード語への又はコード語によるデータビットの挿入を意味する。 10

#### 【0014】

物理インタフェース102はまた、送信器(TX)エラー・リカバリー・モジュール104を含み、一方、物理インタフェース122はまた、受信器(RX)エラー・リカバリー・モジュール124を含む。両方のエラー・リカバリー・モジュール104及び124は、エラーを検出し、かつライン符号化又は復号化に関連して又はそれと共に上述のエラーを任意的に訂正するように作動する。図1Aは、一方向の通信を示すが、当業者は、TXエラー・リカバリー・モジュール104及びRXエラー・リカバリー・モジュール124の構造及び/又は機能を組み合わせて送受信機(TX/RX)エラー・リカバリー・モジュール(図示せず)を形成し、物理的媒体110を通じて双方向通信中にエラー検出及び訂正をサポートすることができることを認めるべきである。少なくとも1つの実施形態では、物理的媒体110は、シリアルデータリンクを提供する。データリンクがシリアルデータリンクであるか又はパラレルデータリンクであるかに関係なく、ライン符号器106及びライン復号器126は、ライン符号化/復号化を実施して、(1)データストリーム内に組み込まれたクロックを回復するために適切な回数の遷移がデータストリーム内にあることを保証し、(2)d-c均衡を維持し、かつ(3)データリンクのスペクトル成形を提供して電磁気放出を低減することのうちの1つ又はそれよりも多くを行う。 20

#### 【0015】

特定のな実施形態では、エラー・リカバリー・モジュール104及び124は、例えば、ライン復号化の前、最中、又は後のいずれかの時間にエラー検出を提供することができるように、データビットが受信器(例えば、IC又は電子装置)に到着した直後の早期のエラー検出を容易にする。従って、物理インタフェース124内のエラー検出は、エラー検出及び訂正技術を適用することができるまで、装置121が正しくないか又は破損したデータビットで不必要に他の下流側処理を実行するのを防止することができる。有利な態様においては、エラー・リカバリー・モジュール104及び124は、コンピュータリソースを保存しなければ破損データビットを処理する場合があるコンピュータリソースを保存し、また、本発明の様々な実施形態に従ってエラーを検出及び/又は訂正するのに必要とされる時間を短縮することにより、装置101及び121間の通信を容易にすることができる。また、本発明の様々な特徴により、エラーを検出する時間を短縮することができる。例えば、エラー検出ビットは、従来のように、送信されるデータビットの群の終わりにではなく、アプリケーションデータビットと交互に織り込んで転送することができる。少なくとも1つの実施形態では、ライン符号器106は、シリアルデータリンクに対してデータストリームを符号化するためにNB/(N+2)Bライン符号化を使用する。更に、ライン符号器106は、アプリケーションデータビットが物理的媒体110上で通信される速度に影響を及ぼすことなく物理インタフェース(PI)エラー訂正ビットをその符号化データストリーム内に組み込むように作動することができる。有利な態様においては、物理インタフェース102及び122は、全体的又は部分的を問わず、物理層(又は「PHY」)において、エラーを解決するために装置101及び121を必要とすることなく、下位でのエラー検出及び/又は訂正を提供することができる。従って、装置101及び121の作動に対して透明であるように、アプリケーションデータビットと共にPIエ 30 40 50

ラー訂正情報を伝送することができる。言うまでもなく、装置 101 及び 121 の場合のエラー検出及び / 又は訂正は、物理インタフェース 102 及び 122 を超えた回路及び / 又は層での他のエラー・リカバリー技術によって補うことができる。従って、アプリケーションデータビットは、上部層エラー・リカバリー機構のための上部層 (UL) エラー検出ビットを含むことができる。エラー検出及び / 又は訂正は、物理層で行われるので、エラー検出は、有利な態様においては、物理インタフェース 102 及び 122 を超えて行われる場合よりも迅速に行うことができる。

#### 【0016】

図 1B は、本発明の 1 つ又はそれよりも多くの実施形態によりデータリンク層ではなく物理層でエラー検出を提供するように実行される図 1A のエラー・リカバリー・モジュール 104 及び 124 を示している。「国際標準化機構 (ISO) 開放型システム相互接続基準モデル (OSI モデル)」では、物理層 (例えば、下位層 162) 及びデータリンク層 176 を含む他の上部層 160 の機能性が説明されている。図 1B は、装置 101 のための OSI モデル 150 と、装置 121 のための OSI モデル 152 とを示している。特に、OSI モデル 150 及び 152 の各々は、アプリケーション層 170、プレゼンテーション層 172、ネットワーク層 174、データリンク層 176、及び物理層 178 の 1 つ又はそれよりも多くを含み、その全ては、公知でありかつここで詳細に説明する必要がない機能性を有する。OSI モデル 150 によれば、送信 IC 又は装置 101 (すなわち、「送信器」) のデータリンク層 176 は、一般的に、下位層 162 においてライン符号器による符号化の前に上位エラー検出コードを発生させ、一方、受信 IC 又は装置 121 (すなわち、「受信器」) のデータリンク層 176 は、通常、下位層 162 ライン復号器がライン符号化ビットを復号化した後にエラーを検出して訂正する。特に、受信データリンク層 176 は、通常、復号化データビットのビットストリームをフレームに組み立てて、次に、フレームのコンテンツに基づいて CRC 検査のような上位エラー検出コードを計算する。一例においては、フレーマー (FR) 193 は、ビットの連続した組をフレームに組み立てる。特に、フレーマー 193 は、フレームのフィールドの 1 つにおいて「フレーム検査シーケンス (FCS)」コードを確立する。FCS は、例えば、巡回冗長検査 (CRC) チェックサムを使用してエラーを検出するのに使用される。FCS は、通常、CRC 検査と突き合わせてエラーが存在するか否かを判断する。説明のために、「イーサネット」フレームが、ヘッダビット及び 1500 バイトまでのペイロードを含み、CRC 検査が 32 個の連続ビットとしてペイロード端部に添付されていると考える。一般的に、CRC 検査は、物理層 178 よりも上にある 1 つ又はそれよりも多くの上部層 160 内に添付される。図示のように、物理層 178 は、少なくとも一部の実施形態では、下位層 162 が上部層 160 内で採用されたあらゆる上位エラー・リカバリー機構に関係なく下位エラー検出及び / 又は訂正 (エラー・リカバリー) を提供することを可能にするために、下位エラー・リカバリー機能 (Err Rec) 190 及び 192 を含む。少なくとも 1 つの実施形態では、下位エラー・リカバリー機能 (Err Rec) 190 及び 192 は、全体的又は部分的を問わず、エラーが発生したか否かを判断するために、バッファ (例えば、FIFO バッファ) 内にアプリケーションデータビットが一時的に記憶される前に又は記憶中に作動する。エラーが存在しない場合、アプリケーションデータビットは、ビットの部分集合への組立てのために上部層に放出される。ビットの部分集合は、「イーサネット」フレーム、メモリ装置のアドレスなどを表すことができる。特定のな実施形態では、物理層 178 とデータリンク層 176 の間の境界は、アプリケーションデータビットが組立てのためにバッファを出て上部層に向かう点となっている。

#### 【0017】

どの層が従来のエラー検出及び訂正を実行するかに関係なく、従来の上位エラー検出コードは、一般的に、ペイロードと類似の方法で送信される。しかし、本発明の様々な実施形態によれば、下位エラー検出コード及びペイロードは、パラレルのような異なる方法でライン復号化することができる。本明細書で使用される時、ペイロードという用語は、一般的に、アプリケーション層によって使用されるプログラム命令及び / 又はプログラムデ

ータを表すビットの集まり（すなわち、アプリケーションデータビット）を意味する。アプリケーションデータビットは、アプリケーションデータを構成し、従って、データビット転送を制御する制御コードを含まない。それに反して、検出コードは、データビット送信の信頼性を制御するのに使用されるので制御コードである。従って、様々な実施形態によれば、アプリケーションデータビットとエラー検出ビットの両方の一部分は、同時に又はほぼ同時にライン復号化される（かつライン符号化される）（特にアプリケーションデータビットとエラー検出ビットの両方の一部分を共に符号化してコード語を形成する場合）。一実施形態では、ライン復号器は、コード語を変換して復号化データビットを形成するように作動し、復号化データビットは、アプリケーションデータビットと物理インタフェース（PI）エラー検出ビットの両方を含むことができる。従来の上位エラー検出コードは、連続ビットの群（例えば、「イーサネット」フレームのためのCRC検査のビット）として送信され、通常、ライン復号器を通じてアプリケーションデータビットと類似の方法で上部層に渡されることに注意されたい。その結果、従来の上位エラー検出技術では、一般的に、コンピュータリソースを浪費して上位層でのエラー・リカバリーのために制御コードをライン復号化し、一方、本発明の少なくとも特定のな実施形態によれば、下位エラー検出コードは、物理層へ以外にはどこにも伝達される必要がない。

10

20

30

40

50

#### 【0018】

図2Aは、本発明の一実施形態によりエラー・リカバリー・モジュールを実行して組込クロック及びクロック・リカバリーを使用する物理インタフェースで確実なデータ転送を提供する回路のブロック図である。図示の例においては、電子装置200は、それぞれ、RXエラー・リカバリー・モジュール（ERM）212及びRXエラー・リカバリー・モジュール（ERM）232から成る物理インタフェース210及び230を含む。尚、図2AではTXエラー・リカバリー・モジュールを含む物理インタフェースが割愛されているが、当業者は、このようなTXエラー・リカバリー・モジュールも実行することができるが機能性が逆であることを認めるべきである。一部の 경우에는、物理インタフェース210及び230は、例えば、入力/出力（I/O）リング202内の回路として形成される。I/Oリング202は、結合ワイヤのような導体を、信号をコア回路204に送信する集積回路のような外部信号発生源（図示せず）に結合する入力端子及び出力端子を含む。RXエラー・リカバリー・モジュール212及びRXエラー・リカバリー・モジュール232は、コア回路に信号を送出するためにコア回路204に結合される。一部の形態で使用される時、コア回路という用語は、1つ又はそれよりも多くのコア機能を実行し、かつ物理インタフェース210及び230を通じて入力を受信又は入力を送出するように構成することができるあらゆる種類の回路を意味する。一部の形態では、コア回路204は、知的財産（IP）コアから形成することができ、知的財産（IP）コアは、ハードウェア説明言語（HDL）コード（例えば、「VHDL」又は「Verilog」）で又はネットリストファイルとして表すことができる。従って、コア回路204は、携帯式であるので、物理インタフェース210及び230によって行われる下位層エラー検出及び訂正から恩典を受ける殆どの設計に容易に挿入することができる。一例として、コア回路204は、動的ランダムアクセスメモリ（DRAM）、マイクロコントローラ又はプロセッサ回路、又は物理インタフェースでのエラー検出及び訂正の実行から恩典を受けるようなあらゆる種類の回路のようないずれかの回路であると考えられる。有利な態様においては、下位エラー検出及び訂正は、コア回路204に対して透明である方法で作動する（すなわち、物理インタフェースエラー・リカバリーでは、エラー検出及び/又は訂正を提供するコア回路204のリソースが必要でなくてもよい）。

#### 【0019】

物理インタフェース210は、 $N + m$ 個の符号化ビットを受信してライン復号器エラー検出器（LD/ED）214に入れ込むように構成されたシリアル物理インタフェースであり、ライン復号器エラー検出器（LD/ED）214は、 $N + m$ 個の符号化ビットの全て又は一部を復号化して $N$ 個の復号化アプリケーションデータビットを形成するように構成される。「 $m$ 」の値は、符号化データビット数とアプリケーションデータビット数に基

づくビット数の差を表している。例えば、8 B / 10 B ライン符号器は、2 という m を有し、2 という m は、差である。ライン復号器エラー検出器 214 は、符号化ビットをライン復号化し、次に、エラーが検出可能である場合はエラーを検出するように作動する。ライン復号器エラー検出器 214 がエラーを検出した場合、エラー訂正器 (EC) 216 にエラー・リカバリー対策を講じるように指示する。一実施形態では、エラー訂正器 216 は、エラー表示信号 218 を外部 IC 又は装置に送信して、 $N + m$  個の符号化ビットを物理インタフェース 210 に再送信することなどによりエラーを解決することができるように外部 IC 又は装置に警告するように構成される。有利な態様においては、物理インタフェース 210 及び 230 は、コア回路 204 を修正することなく、エラー検出及び訂正をコア回路 204 に対して行うことにより、エラー検出又は訂正をコア 204 に統合するために必要とされるエンジニアリングリソースが保存される。別の実施形態では、エラー訂正器 216 は、エラーを訂正することができるように、エラー表示信号 220 をコア回路 204 に送信するように構成される。この場合、コア回路 204 は、アプリケーション特有の論理 244 を含んであらゆる検出エラーを訂正する。有利な態様においては、従って、コア回路 204 (又は他の場所) におけるアプリケーション特有の論理 244 によるエラー処理により、エラー検出及び訂正が可能になり、エラー訂正を容易にする送信 IC 又は装置 (図示せず) が不要になる。それによって、特に、半導体パッケージに封入された場合、電子装置 200 が必要とする信号導体及び / 又はピンの数が最小限に抑えられる。更に別の実施形態では、物理インタフェース 210 のエラー訂正器 216 は、いずれかのエラー表示信号 218 又は 220 を送ることなくエラーを訂正することができる。例えば、エラー訂正器は、自動的にエラーを訂正する順方向のエラー訂正器技術を実行することができる。

10

20

#### 【0020】

次に、外部 IC が、例えば DRAM コントローラ (図示せず) であり、コア回路 204 が、DRAM メモリであると考え。そこで、物理インタフェース 230 は、チップ間インタフェースとなって、エラー訂正後のアドレス及びデータを DRAM コントローラ及び DRAM メモリ (すなわち、コア回路 204) 間で交換することができ、その 1 つ又はそれよりも多くは、エラー検出及び訂正を提供するようには修正されない (機能的又は構造的に)。一部の実施形態では、物理インタフェース 210 及び 230 は、第 1 の基板上に形成され、第 1 の基板は、入力 / 出力 (I/O) リング 202 を含む。コア回路 204 は、第 2 の基板上に形成される。第 1 及び第 2 の基板が別々のものである場合、物理インタフェース 210 及び 230 は、コア回路 204 から分離してパッケージ化することができる。しかし、同じ基板上に形成された場合、システムオンチップ (SOC) の一部として形成することができ、かつ、単一のパッケージによって共に封入することができる。代替的な実施形態では、物理インタフェース 230 は、 $N + m$  個の符号化ビットを受信して RX エラー・リカバリー・モジュール 232 に入れ込み、次に、 $N + m$  個の符号化ビットの一部又は全てを復号化して  $N$  個の復号化アプリケーションデータビットを形成するように構成されたパラレル物理インタフェースである。図 2 A では、説明する内容を簡素化するために、RX エラー・リカバリー・モジュール 232 のライン復号器エラー検出器及びエラー訂正器の詳細は割愛されている。図 2 A は、着信データビット (すなわち、コア 204 のために着信) を着信方向に供給する物理インタフェース 210 及び 230 を示すものであることに注意されたい。上述と同じ物理インタフェースは、発信方向 298 のコア 204 からの発信データビット送信 (図示せず) を提供することができる。シリアル物理インタフェース 210 及びパラレル物理インタフェース 230 のいずれも、着信データビットをライン復号化し、特に、その中の組込クロックを回復するように構成されることに注意されたい。他の実施形態では、シリアル物理インタフェース 210 又はパラレル物理インタフェース 230 のいずれか又は両方は、組込クロックではなく外部クロックを受信するように構成することができる。

30

40

#### 【0021】

図 2 B は、本発明の一実施形態によりエラー・リカバリー・モジュールを実行して確實

50

なデータ転送を外部クロックを使用する物理インタフェースにおいて行う図 2 A の回路の別のブロック図である。図 2 A のシリアル物理インタフェース 2 1 0 又はパラレル物理インタフェース 2 3 0 とは異なって、図 2 B の電子装置 2 0 0 は、外部クロック (Ext. CLK) 2 7 4 を受信するように構成されたシリアル物理インタフェース 2 5 0 を含む。また、電子装置 2 0 0 は、外部クロック (Ext. CLK) 2 5 4 を受信するように構成されたパラレル物理インタフェース 2 7 0 を含む。両方の物理インタフェース 2 5 0 及び 2 7 0 は、外部クロック信号を受信することができるので、これらのインタフェースは、組込クロック技術を採用する必要がなく、従って、クロック・リカバリー回路を含む必要がない。従って、エラー・リカバリー・モジュール 2 5 2 及び 2 7 2 は、ライン復号器又はライン復号化処理の後に配置する必要がない。図 2 B の要素は、各々、図 2 A の類似の付番を有する要素と同等の機能性及び / 又は構造を有することができることに注意されたい。

10

#### 【0022】

図 3 は、本発明の特定のな実施形態によりエラーを検出しかつ任意的に訂正するように構成された物理インタフェースを示すブロック図である。物理インタフェース 3 0 0 は、物理層 (PHY) ライン復号器 3 0 2 とエラー・リカバリー・モジュール 3 1 0 とを含み、エラー・リカバリー・モジュール 3 1 0 は、エラー検出ビット抽出器 3 1 2、エラー検出器 3 1 4、及びエラー訂正器 3 1 6 から成る。エラー検出ビット抽出器 3 1 2 は、少なくとも PHY ライン復号器 3 0 2 と機能的に協働し、従って、図 3 においては、PHY ライン復号器 3 0 2 内にあるように示されている。この例においては、PHY ライン復号器 3 0 2 は、少なくとも、ビットストリームの  $N + m$  個の符号化ビット (すなわち、符号又はコード語) を受信し、かつ上述のビットの全て又は一部を復号化して  $N$  個の復号化アプリケーションデータビットを形成するように構成される。例えば、PHY ライン復号器 3 0 2 は、10 個、14 個、20 個、66 個、又は 130 個の符号化アプリケーションデータビット (すなわち、「 $N + 2$ 」ビット) を受信するように作動することができ、次に、それぞれ、上述のビットを復号化して 8 個、12 個、18 個、64 個、又は 128 個の復号化アプリケーションデータビット (すなわち、「 $N$ 」ビット) を形成することができる。少なくともクロック・リカバリー及び DC 均衡化には、2 つのビットが使用されてきた。様々な実施形態では、PHY ライン復号器 3 0 2 は、あらゆる数の符号化データビットを受信し、次に、上述のビットをあらゆる他の数のアプリケーションデータビットに復号化するように作動することができる。例えば、PHY ライン復号器 3 0 2 は、80 ビットの符号化データを復号化して 64 個の復号化アプリケーションデータビットを形成することができる。

20

30

#### 【0023】

特定のな実施形態では、PHY ライン復号器 3 0 2 は、 $N + m$  個のビットの部分集合である  $N + m - 1$  個の符号化ビットを復号化して、 $N$  個の復号化アプリケーションデータビットと  $E$  個のエラー検出ビットを生成する。次に、ビット抽出器 3 1 2 は、下位エラー検出のために 1 つ又はそれよりも多くの物理インタフェースエラー検出ビットを抽出する。例えば、エラー検出ビット抽出器 3 1 2 は、単一のビットを  $N + m$  個 (例えば、10 個のビット) という数量から抽出して  $N + m - 1$  個のビット (例えば、9 個のビット) 復号化ビットを生成することができるが、 $N + m - 1$  個のビット (例えば、9 ビット) 復号化ビットは、 $N$  個の復号化アプリケーションデータビットと  $E$  個の復号化エラー検出ビットとを含む。従って、PHY ライン復号器 3 0 2 は、 $N + m$  個のビットを受信し、かつライン復号化を提供するように構成され、それによって  $N$  個のアプリケーションデータビットと少なくとも 1 つのエラー検出ビットが得られる。有利な態様においては、PHY ライン復号器 3 0 2 は、従来の 10 B / 8 B 復号器を補うか又は従来の 10 B / 8 B 復号器に取って代わるように実行することができるが、従来の 10 B / 8 B 復号器は、符号化ビットに対しては 10 ビットという一般的に使用される符号サイズを使用し、復号化アプリケーションデータビットは、8 ビットというビットサイズを有する。本発明の様々な実施形態では、PHY ライン復号器 3 0 2 は、9 番目のビット (すなわち、10 個の符号化アプリケ

40

50

ーションデータビットと8個の復号化アプリケーションデータビットの間の差の2ビットの一方)を検査してエラーを検出する。適切なPHYライン復号器302は、「NB/(N+1)Bブロックコードに従ってデータを符号化又は復号化する方法及び機器、及びこのようなブロックコードを判断する方法」という名称の米国特許第6,747,580号で説明されているようなN+1ビット/Nビット復号器であり、この特許の内容全体は、引用により全ての目的に対して組み込まれている。尚、図3は、PHYライン復号器302を示すが、当業者は、図3及び本明細書の他の場所で説明するのと逆の方法で作動するようにN個のアプリケーションデータビットとE個のエラー検出ビットとをN+m個の符号化ビットに符号化するPHYライン符号器(すなわち、Nビット/N+mビット符号器)を構成することができることを認めるべきである。

10

#### 【0024】

抽出後、エラー検出ビット抽出器312は、次に、物理インタフェース(PT)エラー検出ビット(EDB)313をエラー検出器314に送り、エラー検出器314は、そのビットを使用してN+m個の符号化データビットが1つ又はそれよりも多くの誤ったデータビットを含むか否かを判断する。エラー検出器314は、いずれかの数のエラー検出技術及びコードに従ってエラーを検出するように構成される。一実施形態では、エラー検出器314は、偶数又は奇数パリティチェッカーとして構成され、物理インタフェースエラー検出ビットは、パリティビット(すなわち、偶数又は奇数のパリティビット)である。有利な態様においては、物理インタフェースエラー検出ビットは、復号化されるアプリケーションデータビットの部分集合と同時にPHYライン復号器302で受信される。それによって、符号化データビットの群がN+m個のビットよりも遥かに大きいサイズを有する可能性がある場合は、エラー検出コードをその群の端部に添える従来のエラー検出技術と比較すると、PHYライン復号器302がエラー検出コードを受信するのに必要とされる時間を低減することができる。別の実施形態では、物理インタフェースエラー検出ビットは、CRC検査の一部(例えば、1ビット)である。この場合、送信物理インタフェース(図示せず)におけるエラー検出コード発生器は、物理インタフェースエラー検出ビットをCRC検査の一部として発生させるように構成される。図6で以下に説明するように、エラー検出コード発生器(又は均等物)は、CRC検査のビットをPHYライン符号器(図示せず)に供給することができる。PHYライン符号器は、物理インタフェースエラー検出ビットをN個の未符号化アプリケーションデータビットと共に挿入してN+2個の符号化データビットを生成する。その結果、エラー検出器314は、N+m個の復号化データビットの各部分集合からの物理インタフェースエラー検出ビットを蓄積するか又は収集し、CRC検査をエラー検出コードとして再構成するように作動する。例えば、PHYライン復号器302によって復号化される10個の符号化ビットの各組に対して、CRC検査の1ビットが8個の復号化アプリケーションデータビットの各組に対して物理インタフェースエラー検出ビットとして供給されたと考える。16ビットのCRC検査が実行された場合、8個の復号化アプリケーションデータビットの16組毎に(すなわち、128ビット毎に)、エラー検出器314CRCは、160ビットCRC検査を再構成することができる。次に、エラー検出器314は、上述の128ビットからCRC検査を発生させ、次に、CRC検査を再構成エラー検出コードと突き合わせてエラーが発生したか否かを判断する。エラー検出器314は、順方向エラー訂正のような他のエラー検出技術を実行することができる。順方向エラー訂正技術の例としては、「Reed-Solomon」コード、「Hamming」コード、及び「Bose-Chaudhuri-Hocquenghem(BCH)コードがある。

20

30

40

#### 【0025】

エラー検出器314がデータビット送信中にエラーが発生したと判断した状態で、このようなエラーが発生したことをエラー訂正器316に知らせる。一部の場合では、エラー訂正器316は、エラーが検出された復号化アプリケーションデータビットの再送信を要求するように作動する。先の例に続いて、16ビットCRC検査の比較でエラーが128個の復号化アプリケーションデータビット内に存在することが分った場合、エラー訂正器

50

316は、送信物理インタフェースが上述の128ビットを再度送るように要求することができる。他の場合には、エラー訂正器316は、アプリケーション特有の回路を起動させて、ユーザによって構成されたエラー処理を提供することができる。少なくとも1つの事例においては、エラー訂正器316は、上述のコア回路のような関連ICを再初期化するなどの何らかの種類のデフォルトエラー訂正措置を講じることができる。

#### 【0026】

図4は、本発明の一実施形態によりエラー検出ビット挿入及び抽出を備えた送信器におけるライン符号化及び受信器におけるライン復号化を示す機能ブロック図である。図400に示すように、送信器物理インタフェース(TX)410は、例えば、PHYライン符号器412と、エラー検出コード発生器416と、エラービット挿入器418と、エラー検出コード発生中にアプリケーションデータビットを維持する任意的なバッファ421とを含む。この例においては、エラー検出コード発生器416は、16ビットCRC検査のようなバッファ421におけるアプリケーションデータビットに基づいて、エラー検出コードを発生させる。次に、エラービット挿入器418は、少なくとも1つのエラー検出ビット(EDB)414を16ビットCRC検査から選択し、次に、バッファ421からのN個のアプリケーションデータビットの発信組と共に挿入して9ビットを形成する。次に、PHYライン符号器412は、9ビットを10個の符号化ビット(又は記号)に符号化する。一部の実施形態では、PHYライン符号器412は、10個の符号化ビットに関するゼロDCオフセット及び/又は任意的なスペクトル特性を維持すると共に、上述の10個の符号化ビット内に非同期クロックを組み込むような方法でエラー検出ビット414を符号化する。特定の実施形態では、PHYライン符号器412は、米国特許第6,747,580号に定めるコード化技術に従って作動することにより、NビットのアプリケーションデータをN+1個の符号化ビットに符号化する。例えば、Nが数字9であると考え、従って、PHYライン符号器412は、8B/10Bコード化技術に関連したデータ転送速度に影響を与えることなく、8個のアプリケーションデータビットと1個のエラービット(すなわち、9ビット)とを10個の符号化ビットに符号化することができる。図4において、Nは、あらゆる負以外の整数を表すことに注意されたい。

#### 【0027】

次に、送信器物理インタフェース410は、物理的媒体420を通じて符号化ビットを受信物理インタフェース(RX)440のエラービット抽出器422に送る。受信物理インタフェース(RX)440は、PHYライン復号器428と、エラービット抽出器422と、エラー検出コード再構成器426と、エラー検出器430と、比較器434と、エラーが検出されていないという復号化アプリケーションデータビット未決確認を維持することができるバッファ431とを含む。PHYライン復号器428は、10個の符号化データビットを9個の復号化ビットに復号化し、9個の復号化ビットは、8個の復号化アプリケーションデータビットと1個の復号化エラー検出ビット(EDB)424を含む。様々な実施形態では、PHYライン復号器428は、クロックデータリカバリ回路(図示せず)を含んで符号化アプリケーションデータビットと共に物理的媒体420で送信された組込クロックを回復する。エラービット抽出器422は、エラー検出ビット424を9個の復号化ビットから抽出してエラー検出コード再構成器426に供給し、エラー検出コード再構成器426は、他の物理インタフェースエラー検出ビットと共にエラー検出ビットを蓄積する。次に、エラー検出コード再構成器426は、元々エラー検出コード発生器416が発生させたエラー検出コードを再構成する。次に、エラー検出器430は、バッファ431におけるN個の復号化アプリケーションデータビットの群に基づいてCRC検査432を発生させる。比較器434は、CRC検査432がエラー検出コード426に適応するか否かに関して比較するように作動することができる。適応した場合には、エラーがなく、適応しなかった場合には、エラーがある。

#### 【0028】

図5は、本発明の一実施形態に従ってセルとしてデータビットの群をライン符号化するフローの一例を示している。一部の実施例においては、送信器物理インタフェースは、フ

10

20

30

40

50

ロー 5 5 0 に示すようにデータを送信することができ、それによってデータビットの群 5 6 0 は、セル（並びにフレームなど）と呼ぶことができる。セル 5 6 0 は、例えば、あらゆる数の制御ビット 5 6 1（例えば、X 個のビット）と、各々が 6 4 ビットの 2 個のアプリケーションデータ語 5 6 3 とを含むことができる。物理層内で P H Y ライン符号器の上方にある上部層装置（図示せず）は、セル 5 6 0 が P H Y ライン符号器によって使用されるように、個別の組 5 6 2 のコード化されていない（すなわち、予めコード化されているか又はコード化されていない）ビットに分解するように構成することができる。各組 5 6 2 は、N 個のビット（例えば、8 ビット）を有する。エラー検出コード発生器（図示せず）は、例えば、セル 5 6 0 に基づいてエラー検出コード 5 7 0 を発生させ、エラー検出コードを 1 2 ビットの C R C として形成する。次に、P H Y ライン符号器は、物理インタフェースエラー検出ビット（E D B）5 7 2 を組 5 6 2 内又は組 5 6 2 の近くでコード化されていないビットストリームに挿入し、N + E 個のビット（例えば、N は、8 ビットとすることができ、E は、1 ビットとすることができ）の組 5 7 4 を形成することができる。次に、P H Y ライン符号器は、シリアルデータリンクのようなデータリンク上で送信されるように、N + E 個のビットの各組 5 7 2 を 1 0 ビットのコード語 5 8 0（例えば、符号化データビット）に変換することができる。図示していないが、受信物理インタフェースでの P H Y ライン復号器は、類似であるが逆に作動することになる。

10

#### 【 0 0 2 9 】

図 6 は、本発明の少なくとも 1 つの特定のな実施形態により少なくとも再送信データビットによるエラー検出及び訂正を提供する物理インタフェースを示すブロック図 6 0 0 である。物理インタフェース 6 0 2 及び 6 5 2 は、高速シリアル通信リンク 6 5 0 を通じて物理的接続性を提供する。物理インタフェース 6 0 2 は、送信されるデータビットを記憶するために先入れ先出しバッファ（F I F O）6 0 4 のような記憶装置を含む。F I F O 6 0 4 は、例えば、データビットの群にわたってエラー検出コードを計算するために記憶データビットを維持する。一部の場合では、F I F O 6 0 4 は、エラーが発生した場合に再送信されるようにデータビットを記憶する。また、物理インタフェース 6 0 2 は、P H Y ライン符号器（P H Y 符号器）6 1 0 と、パラレルデータビットをシリアル化するシリアライザ（S E R）6 1 2 と、エラー検出コード発生器 6 1 6 と、送信器（T X）エラー訂正器 6 1 8 とを含む。特定の時間での F I F O 6 0 4 内のデータビットの群のコンテンツに基づいて、エラー検出コード発生器 6 1 6 は、C R C 検査のようなエラー検出コードを発生させる。エラー検出ビット挿入器 6 1 4 は、C R C 検査を分解し、N 個の未符号化ビット 6 0 6 と共に物理インタフェースエラー検出ビット（E D B）として少なくとも 1 つのビット（例えば、1 つの E ビット）を挿入する。作動においては、P H Y ライン符号器 6 1 0 は、N + E 個のビットを符号化して N + 2 個の符号化データビット 6 0 8 を形成する。次に、シリアライザ 6 1 2 は、リンク 6 5 0 での送信の前に N + 2 個のデータビット 6 0 8 をシリアル化する。受信物理インタフェース 6 5 2 は、デシリアライザ 6 5 4 を含んで N + 2 個のデータビットをパラレルデータビットに変換する。P H Y ライン復号器（P H Y 復号器）6 5 6 は、符号化ビットストリームを復号化して復号化データビットを F I F O 6 5 8 に記憶する。エラー検出ビット（E D B）抽出器 6 6 0 は、少なくとも 1 ビットを物理インタフェースエラー検出ビット（E D B）6 6 2 として抽出してエラー検出器 6 6 6 に送る。エラーコード再構成が完全である場合、エラー検出器 6 6 6 は、エラーが発生したか否かを判断するように作動する。エラー検出器 6 6 6 がエラーなしと判断した場合、更に別の処理のために復号化データビットを放出するように F I F O 6 5 8 に指示する。しかし、エラー検出器 6 6 6 がエラーを検出した場合、受信器（R X）エラー訂正器 6 6 8 に対してエラーの存在を表示する。一例においては、R X エラー訂正器 6 6 8 は、T X エラー訂正器 6 1 8 にリンク 6 5 0 を通じてコンテンツ又はその一部を再送信するように指示する要求 6 6 9 を送る。別の例においては、R X エラー訂正器 6 6 8 は、エラーを処理する要求 6 7 1 をアプリケーション特有の回路（図示せず）に送ることができる。更に別の例においては、受信器エラー訂正器 6 6 8 は、順方向エラー訂正技術などを使用してエラーを訂正することができる。

20

30

40

50



## 【0030】

図7は、本発明の少なくとも1つ特定のな実施形態によりライン復号化なしにエラー検出及び訂正を実行する代替的受信器を示している。受信器(RX)700においては、送信器物理インタフェース(図示せず)は、リンク702を通じてデータビットを送る。エラービット抽出器704は、エラー検出データビット(EDB)706をデータビットから抽出してエラー検出コード708を再構成する。図4の図400とは異なって、受信器700は、受信器440のようにPHYライン復号器を使用しない。従って、受信器700は、クロック信号を組込クロックから回復する必要がない。図示のように、受信器700は、外部クロック701を例えば送信装置(図示せず)から受信するように構成される。リンク702での送信の前に、例えばエラー検出ビットをアプリケーションデータビットに添えるのは、この送信装置である。受信器700は、FIFO720内のアプリケーションデータビットの群に基づいて、検出コード、すなわち、この場合はCRC検査722を計算するために、エラー検出器780を含む。次に、比較器724(例えば、エラー検出器720において)は、送信エラーが発生したか否かを判断する。エラーが存在する場合、エラー訂正器780は、例えば、送信装置にデータビットを再送信するように要求するか又はエラーを訂正すべきであることをアプリケーション特有の論理に知らせることによって是正措置を講じることができる。

## 【0031】

図8は、本発明の一実施形態によるエラー訂正器のブロック図を示している。エラー訂正器800は、この例においては、エラー訂正器コントローラ822を含んでエラーを解決することができる1つ又はそれよりも多くの機能を制御し、特定の機能は、特定の用途に対してユーザによってプログラムされる。例えば、ビデオメモリ用途でのピクセルデータの損失は、他の用途におけるデータ損失ほど重要なものではない。従って、エラー訂正器コントローラ822は、以下のモジュールのいずれかを選択してエラー訂正を実行することができる。再送信要求モジュール824は、エラーを検出するとデータビットを再度送るために送信器物理インタフェース行きの要求を発生させる。アプリケーション特有のリカバリーモジュール814は、設計担当者によって指定されるような特注のエラー処理ルーチンを開始するようにアプリケーション特有の回路に知らせる。しかし、エラー訂正器コントローラ822は、エラーを検出すると、ICを再度初期化するなどのデフォルトエラー・リカバリー技術を選択することができる。それによって両方の物理インタフェースは、既知の状態に再度初期化されることになる。様々な代替的な実施形態では、エラー訂正器800は、周波数セクタ830と、振幅セクタ840と、コード化セクタ850とを含むことができる。物理インタフェース間の通信リンク上の信号が行き詰った場合、周波数セクタ830は、リンク上のデータ転送速度を調節してエラー率を低減することができる。代替的に、振幅セクタ840は、エラー率が低減されるまで信号振幅を調整することができる。コード化セクタ850を使用すると、リンクコード化を例えば18B/20Bから8B/10Bに変更して、エラーが検出される時間を短縮することができる。例えば、32ビットのCRCがエラー・リカバリーに実行され、一度にCRCの1ビットが送られると考える。18B/20Bでは、32ビットのCRC(すなわち、 $8 \times 32 = 576$ ビット)を形成するのに576個のアプリケーションデータビット(8ビット)が必要であり、一方、8B/10Bでは、256個のアプリケーションデータビットが必要である。従って、コード化セクタ850は、通信チャンネルでエラーコード(例えば、CRC)が送信される率を上げることができる。エラー訂正器コントローラ822は、これらのセクタを個別に又は組み合わせて選択することができる。

## 【0032】

説明を目的とした以上の説明では、本発明の徹底的な理解が得られるように特定の名称を使用した。しかし、本発明を実施するためには特定の詳細は必要ではないことが当業者には明らかであろう。実際に、以上の説明は、本発明の何らかの特徴又は態様を何らかの実施形態に限定するものとして読み取るべきではなく、むしろ、一実施形態の特徴及び態様は、他の実施形態と容易に交換することができる。例えば、様々な実施形態の以上の説

10

20

30

40

50

明は、物理層ライン復号器に関連するが、説明した内容は、物理層ライン符号器及び全ての種類の通信リンク、並びに「ギガビットイーサネット」、HDMI、TMDS、DVI、及びSATAのような多くの種類の通信プロトコルに適用可能である。また、様々な実施形態は、電子装置間のポイントツーポイント通信、並びに1つの装置から多数の装置までのブロードキャストに適用可能である。以上の説明した内容の一部は、シリアルデータ通信に関連するものであるが、本発明の様々な実施形態をパラレルデータ通信にも適用することができる。

#### 【0033】

すなわち、本発明の特定のな実施形態の以上の説明は、例示及び説明を目的として示されたものである。網羅的であること又は本発明を開示した形態に正確に限定することを意図したものではない。明らかに、多くの変更及び変形が先の教示内容に鑑みて可能である。実施形態は、本発明の原理及びその実際的な用途を最も良く説明するために選択しかつ説明したものである。その結果、当業者は、考えている特定の用途に適するような様々な修正を行って本発明及び様々な実施形態を最良に利用することができる。明白なことであるが、本明細書で説明する全ての利点が本発明の各実施形態によって達成される必要はない。むしろ、どの特定のな実施形態も上述の利点の1つ又はそれよりも多くをもたらすことができる。特許請求の範囲及びその均等物が、本発明の範囲を規定するものとする。

#### 【図面の簡単な説明】

#### 【0034】

【図1A】本発明の少なくとも1つの特定のな実施形態によりエラー・リカバリー・モジュールを含んでエラーを検出する物理インタフェースを示すブロック図である。

【図1B】本発明の1つ又はそれよりも多くの実施形態によりデータリンク層ではなく物理層でエラー検出を提供するように実施されるエラー・リカバリー・モジュールを示す図である。

【図2A】本発明の一実施形態により組込クロック及びクロック・リカバリーを使用する物理インタフェースで確実なデータ転送を提供するためにエラー・リカバリー・モジュールを実施する回路のブロック図である。

【図2B】本発明の一実施形態により外部クロックを使用する物理インタフェースで確実なデータ転送を提供するためにエラー・リカバリー・モジュールを実施する図2Aの回路の別のブロック図である。

【図3】本発明の特定のな実施形態によりエラーを検出して任意的に訂正するように構成された物理インタフェースを示すブロック図である。

【図4】本発明の一実施形態によりエラー検出ビット挿入及び抽出を備えたライン符号化及びライン復号化を示す機能ブロック図である。

【図5】本発明の一実施形態に従ってセルとしてデータビットの群をライン符号化する例示的なフローを示す図である。

【図6】本発明の少なくとも1つの特定のな実施形態によりデータビットを少なくとも再送信することによってエラー検出及び訂正を提供する物理インタフェースを示すブロック図である。

【図7】本発明の少なくとも1つの特定のな実施形態によりライン復号化なしにエラー検出及び訂正を実施する代替的受信器を示す図である。

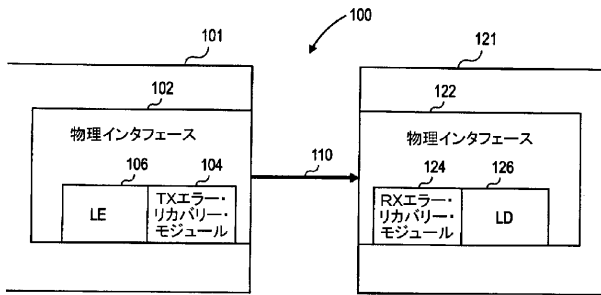
【図8】本発明の一実施形態によるエラー訂正器のブロック図である。

#### 【符号の説明】

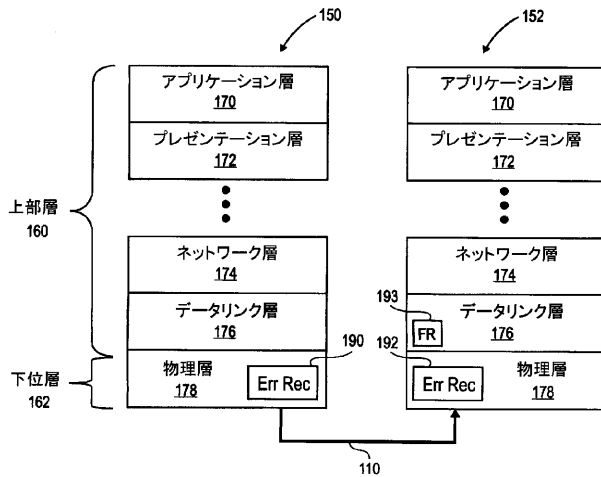
#### 【0035】

- 100 物理インタフェースを示すブロック図
- 101、121 電子装置
- 106 ライン符号器
- 110 物理的媒体
- 126 ライン復号器

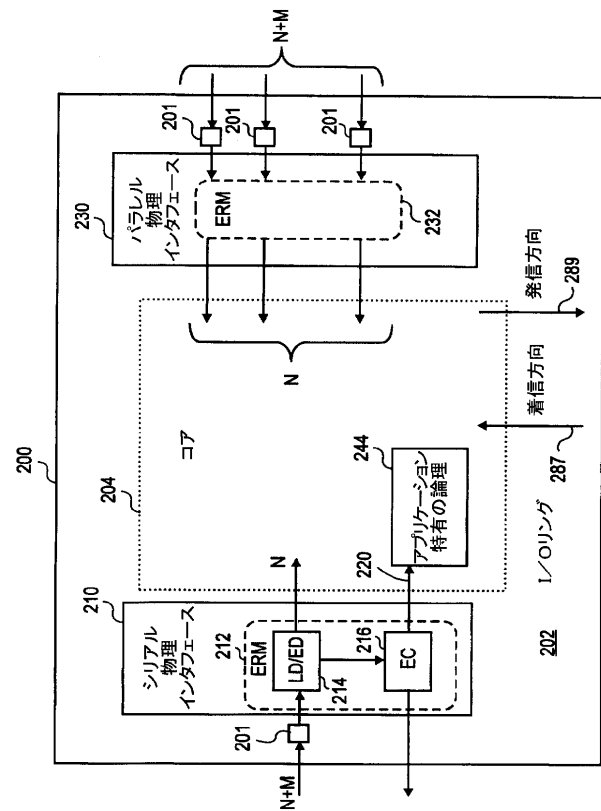
【図 1 A】



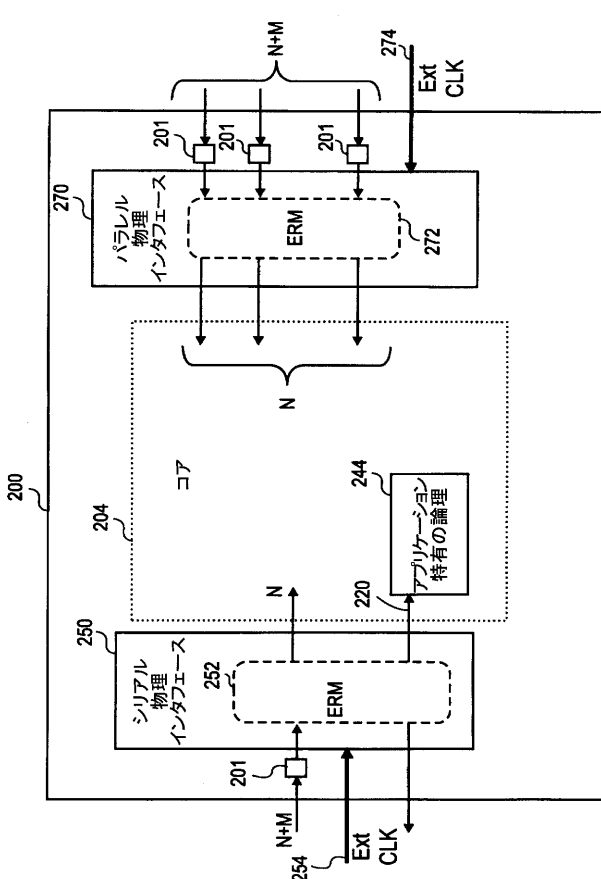
【図 1 B】



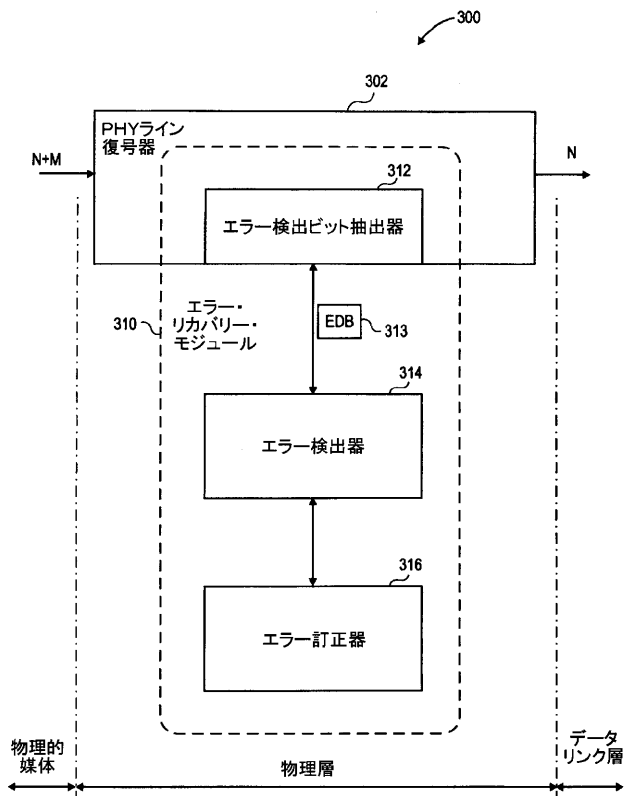
【図 2 A】



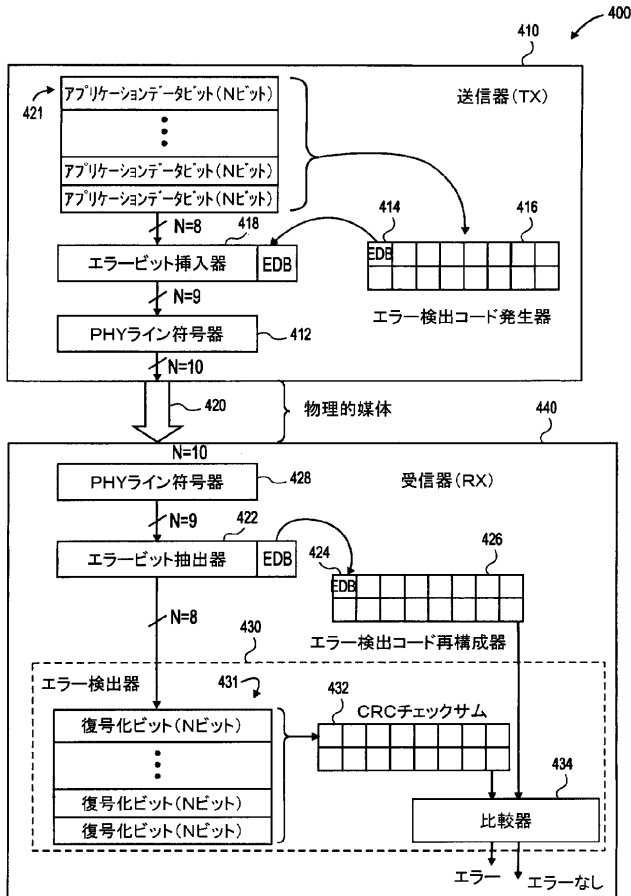
【図 2 B】



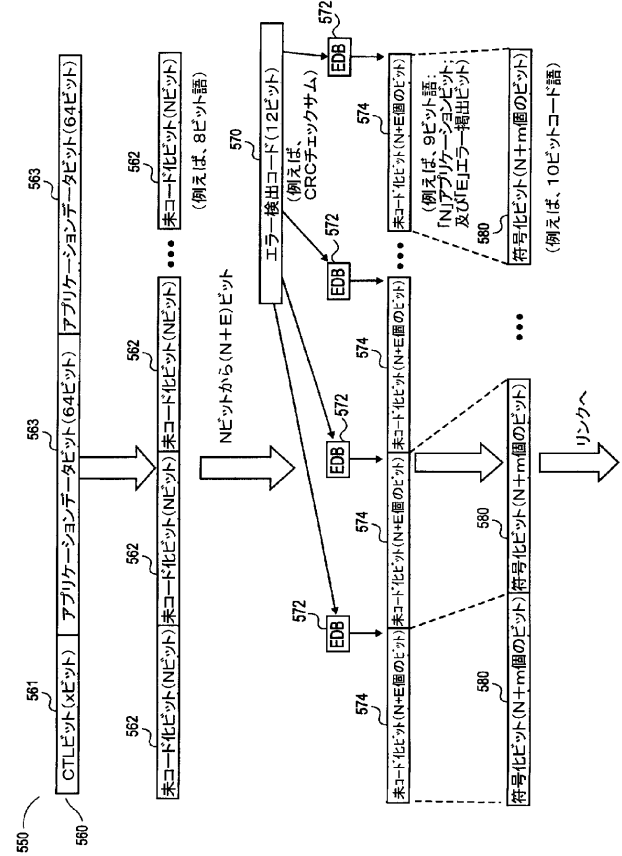
【図 3】



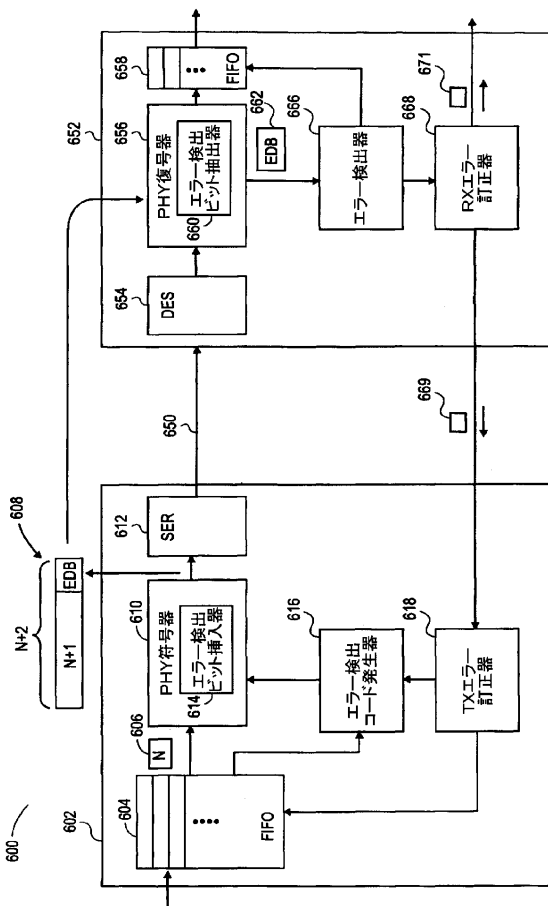
【図 4】



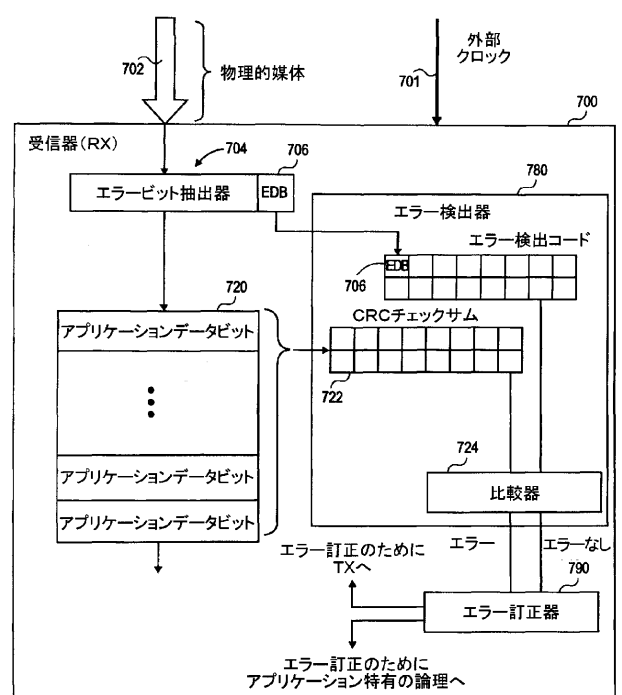
【図 5】



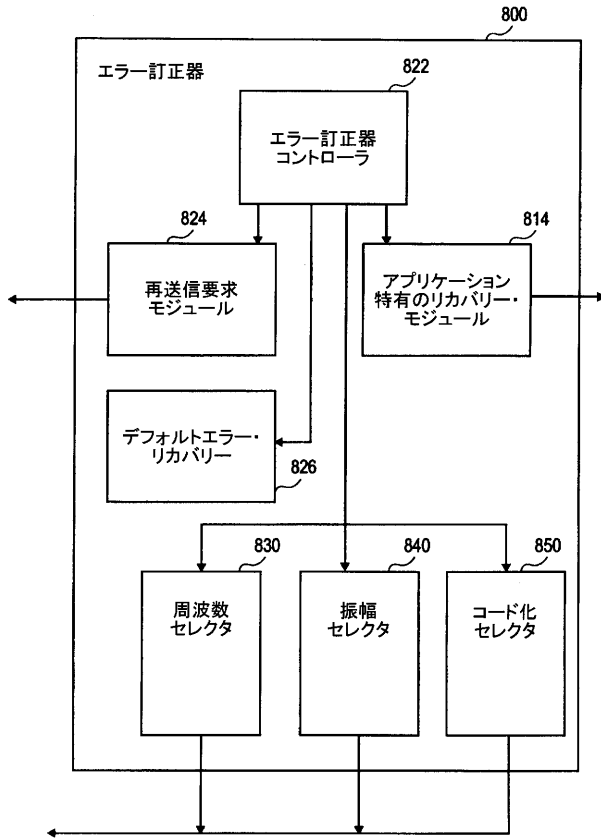
【図 6】



【図 7】



【 図 8 】



---

フロントページの続き

(72)発明者 ブライアン ケイ シュミット

アメリカ合衆国 カリフォルニア州 9 4 0 4 0 マウンテン ヴィュー ケンゾー コート 3  
3 7 0

(72)発明者 ローレンス ルウェリン ブッチャー

アメリカ合衆国 カリフォルニア州 9 4 0 4 3 マウンテン ヴィュー オルディーン アベニ  
ュー 3 3 2

F ターム(参考) 5K014 BA06

5K030 GA12 HA08 HB12 KA03 LA01 MB05

【 外国語明細書 】

## **ERROR DETECTION IN PHYSICAL INTERFACES FOR POINT-TO-POINT COMMUNICATIONS BETWEEN INTEGRATED CIRCUITS**

### **BRIEF DESCRIPTION OF THE INVENTION**

This invention relates generally to an apparatus and method for providing reliable data communications between electronic devices, and more particularly, to a physical interface configured to efficiently detect errors at the physical interface during point-to-point transmission of data bits over a physical medium between electronic devices. In one embodiment, the point-to-point transmission of data bits employs low-amplitude, high-frequency signaling along a serial data link.

### **BACKGROUND OF THE INVENTION**

Point-to-point communication between electronic devices, such as integrated circuits ("ICs"), is generally considered to be reliable, especially when a dedicated synchronous clock line is used to synchronize data transfers. In particular, many system designers expect communications between ICs to conform to a prevailing assumption that data bits exchanged between ICs are the same data bits. For example, most conventional processor ICs are designed to act upon data fetched from a memory IC regardless of whether either the address bits transmitted to the memory IC were correctly received, or the data received by the processor IC were the same as sent by the memory IC. But with traditional physical interfaces both transmitting data bits at faster data rates and at decreasing amplitudes (i.e., decreasing transmission power), the effects of background noise on the strength of the data signals causes the signal-to-noise ratios to decrease, which is indicative of an increase in errors that could thwart reliable point-to-point communications. Further, many system designers are

implementing physical interfaces that serially transport data over a serial data link using asynchronous clocking techniques, whereby the clock is embedded in encoded serialized data. Examples of high-speed serial communications technologies having serial data links include Serial ATA ("SATA")®, Transition Minimized Differential Signaling ("TMDS"), PCI Express, InfiniBand®, and the like. While conventional physical interfaces that provide for serial data communications reduce both noise and power, at least one drawback is that conventional serial data links are susceptible to data corruption during transit.

In the field of computer networking, the Open Systems Interconnect ("OSI") model describes a common approach to detect and correct errors between two end stations (or computing devices). According to this approach, a data link layer calculates an error-detecting code (e.g., a cyclic redundancy code, "CRC") based on a frame of transmitted bits. Then, it appends the error-detecting code as contiguous bits to the frame and then passes both the error-detecting code and the application data bits to a physical interface for line encoding (or coding). One example of line encoding translates 8 application data bits into 10 encoded bits. This type of line encoding technique is commonly known as 8B/10B. After line encoding is performed, the serial physical interface drives the encoded bits out over a communications medium to another serial physical interface, which recovers the clock and decodes the bits. The data link layer at the receiver then applies error detection and correction techniques using the contiguous bits of error-detecting code.

While functional, the data link layer performs these error detection and correction processes beyond the physical interface, thereby delaying error detection and/or resolution. Also, conventional physical interfaces are generally not well suited to facilitate error detection and correction in a standardized manner. For example, traditional physical interfaces are optimized to provide synchronous and parallel data communications between integrated circuits. As such, they can not readily be adapted to provide timely and reliable data transfers for asynchronous and serial data communications, such as over a serial data link. Further, conventional physical interfaces are not generally conducive to at least provide: (1) transparent error detection and recovery techniques that do not require an application to participate in specialized error handling techniques, and (2) efficient transmissions of error detection codes to increase the rate that those codes are transferred over the serial data link.



In view of the foregoing, it would be desirable to provide a physical interface, an apparatus and a method that minimize the above-mentioned drawbacks and provides for at least error detection and optional error recovery.

## SUMMARY OF THE INVENTION

An apparatus, system and method for detecting errors in a physical interface during the transmission and/or receipt of data communications between integrated circuits ("ICs") are disclosed. According to various embodiments of the invention, a physical interface can facilitate communications over either serial or parallel data links, or it can be configured to use either an embedded asynchronous clock or an external synchronous clock.

In one embodiment, an apparatus is configured to operate as or within a receiving physical interface. The apparatus includes a decoder configured to decode a subset of encoded data bits to yield decoded data bits. It also includes a physical interface ("PI") error detection bit extractor configured to extract a physical interface error detection bit from the decoded data bits. As such, the apparatus uses the physical interface error detection bit to determine whether the encoded data bits include at least one erroneous data bit as an error. In some embodiments, the apparatus includes an error detector configured to operate within a physical layer. In at least one embodiment, the apparatus efficiently transmits error detection codes within, for example, an  $NB/(N+1)B$  line coder. The encoded data bits can be substantially direct current ("DC")-balanced. Further, the encoded data bits can be configured to shape the spectral properties of emitted radiations. Also, those encoded data bits can also include an embedded asynchronous clock that can be used to recover a clock at a receiving device. In some instances, the apparatus includes an error detector configured to operate within a physical layer to detect errors. The error detector performs error detection earlier than at a data link layer or at any upper layers of the Open Systems Interconnection ("OSI") seven-layer reference model. In some cases, the physical interface error detection bit is a parity bit, while in other instances, the physical interface error detection bit is a portion of an error-detection code. For example, the error-detection code can be a cyclic redundancy check ("CRC") checksum. The error detector in this case is configured to accumulate the physical interface error detection bit as well as other physical interface error detection bits from other subsets of the encoded data bits (i.e., other code

words) to reconstruct the error-detection code. Note that the error detector can be configured to generate a CRC checksum from the physical interface error detection bit and the other physical interface error detection bits. Then, the error decoder can compare the CRC checksum against the error-detection code to determine whether an error has occurred.

In various embodiments, the physical interface error detection bit is received at the apparatus concurrent with the subset of the encoded data bits and each of the other physical interface error detection bits are also received at the apparatus concurrent with each of the other subsets of the encoded data bits, thereby decreasing the time the apparatus receives the error-detection code that otherwise is appended to follow the encoded data bits. In at least one embodiment, the apparatus also includes an error corrector configured to correct the error. The error corrector can be configured to either request a retransmission of the subset of the encoded data bits, or signal an application-specific module that an error has been detected so that the application-specific module can resolve the error. Or, the error corrector can take default measures, such as reinitializing one or more ICs. Note that one of the subsets of the encoded data bits can include  $N+m$  data bits, which can be decoded to obtain  $N$  application data bits as decoded data bits. Also note, that the apparatus can be configured to receive encoded data bits as  $N+2$  data bits and to provide  $N$  application data bits as outputs. For example, when  $N$  is 8, the apparatus is configured as an 8B/10B decoder, it provides nine decoded data bits: eight application data bits and one physical interface error detection bit for error detection.

In another embodiment of the invention, an apparatus for generating error-detection codes in a physical interface for the transmission of data between integrated circuits ("ICs") is disclosed. In one embodiment, the apparatus is configured to operate as or within a transmitting physical interface. The apparatus can include an  $N$  bit-to- $N+2$  (" $N$  bit/ $N+2$ ") bit physical layer ("PHY") encoder. This encoder is configured to insert a physical interface error detection bit with  $N$  application data bits to form  $N+1$  unencoded data bits, and to encode the  $N+1$  unencoded data bits to yield  $N+2$  encoded data bits. The apparatus also includes an error-detection code generator configured to generate a number of bits constituting an error-detection code, which includes the physical interface error detection bit.  $N$  can represent any integer number of data bits. The apparatus, in some cases, can include a serializer configured to serialize the  $N+2$  encoded data bits to form serial  $N+2$  encoded data bits, thereby interleaving single error-detection data bits with the serial  $N+2$  encoded data bits. In at least one embodiment,

the apparatus includes storage to store a group of data bits that include the N data bits. The error-detection code generator is configured to examine the group of data bits to generate an error-detection code. In a specific embodiment, the apparatus includes a transmitter ("TX") error corrector configured to retransmit one or more subsets of the group of data bits when the transmitter error corrector receives a request for retransmission.

In yet another embodiment of the invention, a physical interface is formed as an integrated circuit ("IC") on a first substrate to detect transmission errors in data exchanged with a core IC formed on a second substrate. The core IC is another IC that contains logic for implementing an application-specific function, such as a storage function of a memory chip. The physical interface includes a plurality of input ports and output ports, including a first subset of input ports configured to receive in-bound encoded data bits and a first subset of output ports configured to transmit in-bound decoded data bits to the core IC. The physical interface also includes one or more error recovery modules coupled between the plurality of input ports and output ports. At least one of the error recovery modules is coupled between at least one of the first subset of input ports and at least one of the first subset of output ports. The error recovery module optionally can include a physical layer ("PHY") decoder configured to detect errors in the in-bound encoded data bits and to initiate an action to correct the errors.

In still yet another embodiment of the invention, a high-speed serial communications link between at least two integrated circuits ("ICs") is disclosed. This link includes a physical medium for conveying data, a transmitter device coupled to a first end of the physical medium, and a receiver device coupled to a second end of the physical medium. It also includes an error detector configured to determine an error using the physical interface error detection bits. Note that the transmitter device includes an N bit/N+2 bit encoder that generates encoded data bits with physical interface error detection bits encoded therein, and the receiver device includes an N+2 bit/N bit decoder, which decodes the encoded data bits, and an error detector to determine an error using the physical interface error detection bit. Note that the transmitter and receiver devices can be one or more of any of the following: a transition-minimized differential signaling ("TMDS") device, a high-definition multimedia interface ("HDMI") device, a multi-Gigabit Ethernet device(e.g., a 10 Gbps Ethernet, or "10 GbE," device), and/or a serial advanced technology attachment ("SATA") device. Note, too, that the

transmitter and receiver devices can implement Digital Visual Interface (“DVI”) and/or Low Voltage Differential Signaling (“LVDS”), as well as any other equivalent technologies.

### **BRIEF DESCRIPTION OF THE FIGURES**

The invention is more fully appreciated in connection with the following detailed description taken in conjunction with the accompanying drawings, in which:

FIG. 1A is a block diagram illustrating physical interfaces that include error recovery modules to detect errors, according to at least one specific embodiment of the invention;

FIG. 1B illustrates error recovery modules and implemented to provide error detection at a physical layer rather than at a data link layer, according to one or more embodiments of the invention;

FIG. 2A is a block diagram of a circuit implementing error recovery modules to provide reliable data transfers at physical interfaces that use embedded clocks and clock recovery, according to one embodiment of the invention;

FIG. 2B is another block diagram of the circuit of FIG. 2A implementing error recovery modules to provide reliable data transfers at physical interfaces that use external clocks, according to one embodiment of the invention;

FIG. 3 is a block diagram showing a physical interface configured to detect and optionally correct errors, according to a specific embodiment of the invention;

FIG. 4 is a functional block diagram illustrating line encoding and line decoding with error-detection bit insertion and extraction, according to one embodiment of the invention;

FIG. 5 depicts an exemplary flow for line encoding a group of data bits as a cell in accordance with one embodiment of the invention;

FIG. 6 is a block diagram illustrating physical interfaces that provides for error detection and correction by at least retransmitting data bits, according to at least one specific embodiment of the invention; and

FIG. 7 illustrates an alternative receiver for implementing error detection and correction without line decoding, according to at least one specific embodiment of the invention.

FIG. 8 depicts a block diagram of an error corrector according to one embodiment of the present invention.

Like reference numerals refer to corresponding parts throughout the several views of the drawings. Note that most of the reference numerals include one or two left-most digits that generally identify the figure that first introduces that reference number.

## **DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS**

FIG. 1A is a block diagram 100 illustrating physical interfaces that include error recovery modules to detect errors in association with line encoding/decoding, according to at least one specific embodiment of the invention. Physical interfaces 102 and 122 provide for physical connectivity between devices 101 and 121, each of which can be an integrated circuit, such as a memory chip, or an electronic device, such as an Ethernet line card or a networked computer. Physical interfaces 102 and 122 include a line encoder (“LE”) 106 and a line decoder (“LD”) 126, respectively, to facilitate transmission and receipt of data bits via physical medium 110. In one embodiment, line encoder (“LE”) 106 performs line encoding (also referred to as “line coding”), which is the translation of digital information expressed in binary (i.e., data bits) into a specific code (or code word) to optimize transmission of the encoded data bits over a particular type of medium, such as copper wire, optical fiber, radio waves, and the like. Line decoding is the inverse process of translating from the specific code to data bits. Both line encoders and line decoders are tuned to optimally operate with a specific type of medium. As used in some embodiments, the term encoding refers to the translation of data bits into code words as well as the insertion of data bits into or with the code words.

Physical interface 102 also includes a transmitter (“TX”) error recovery module 104, whereas physical interface 122 also includes a receiver (“RX”) error recovery module 124. Both error recovery modules 104 and 124 operate to detect errors and to optionally correct those errors either in association or in conjunction with line encoding or decoding. While FIG. 1A depicts unidirectional communication, an ordinarily skilled artisan should appreciate that the structures and/or functions of TX error recovery module 104 and RX error recovery module 124 can be combined to form a transceiver (“TX/RX”) error recover module (not shown) to support error detection and correction during bi-directional communications via physical medium 110.

In at least one embodiment, physical medium 110 provides for a serial data link. Regardless of whether the data link is a serial data link or a parallel data link, line encoder 106 and line decoder 126 implement line encoding/decoding to do one or more of the following: (1) ensure that there are an appropriate number of transitions in the data stream to recover a clock embedded in the data stream, (2) maintain a d-c balance, and (3) provide for spectral shaping of the data link to reduce electromagnetic emissions.

In a specific embodiment, error recovery modules 104 and 124 facilitate early error detection shortly after data bits arrive at a receiver (e.g., an IC or electronic device) such that error detection can be performed, for example, any time before, during or after line decoding. As such, error detection within physical interface 124 can prevent device 121 from performing other downstream processes unnecessarily with incorrect or corrupted data bits until error detection and correction techniques can be applied. Advantageously, error recovery modules 104 and 124 can conserve computational resources that otherwise might process corrupt data bits, and can also expedite communications between devices 101 and 121 by reducing the time required to detect and/or correct errors, according to various embodiments of the invention. Various features of the invention can also decrease the time to detect an error. For example, error detection bits can be transported in an interleaved fashion with application data bits rather than at an end of a group of data bits being transmitted, as is customary. In at least one embodiment, line encoder 106 uses a  $NB/(N+2)B$  line encoding for encoding a data stream for a serial data link. Further, line encoder 106 can operate to embed physical interface ("PI") error correction bits in that the encoded data stream without affecting the rate at which application data bits are communicated over physical medium 110. Advantageously, physical interfaces 102 and 122 can implement low-level error detection and/or correction—in whole or in part—at the physical layer (or "PHY") without requiring devices 101 and 121 to resolve errors. As such, PI error correction information can be transmitted along with the application data bits so as to be transparent to the operation of devices 101 and 121. Of course, error detection and/or correction for devices 101 and 121 can be supplemented by other error recovery techniques at circuits and/or layers beyond physical interfaces 102 and 122. As such, the application data bits can include upper-layer ("UL") error detection bits for upper-layer error recovery mechanisms. As error detection and/or correction are performed at physical layers, error detection can

advantageously be performed more quickly than if performed beyond physical interfaces 102 and 122.

FIG. 1B illustrates error recovery modules 104 and 124 of FIG. 1A implemented to provide error detection at a physical layer rather than at a data link layer, according to one or more embodiments of the invention. The International Standard Organization's ("ISO's") Open System Interconnection Reference Model ("OSI Model") describes the functionality of: a physical layer (e.g., as a low-level layer 162) and other upper layers 160, which include a data link layer 176. FIG. 1B shows an OSI model 150 for device 101 and an OSI model 152 for device 121. In particular, each of OSI models 150 and 152 includes one or more of an application layer 170, a presentation layer 172, a network layer 174, a data link layer 176, and a physical layer 178, all of which have functionalities that are well known and thus need not be discussed here in detail. In accordance with the OSI Model 150, data link layer 176 of a transmitting IC or device 101 (i.e., a "transmitter") generally generates upper-level error detection codes prior to encoding by a line encoder in low-level layer 162, whereas data link layer 176 of a receiving IC or device 121 (i.e., a "receiver") normally detects and corrects errors after the line decoder in low-level layer 162 decodes the line-encoded bits. In particular, the receiving data link layer 176 usually assembles a bit stream of the decoded data bits into frames and then computes an upper-level error detection code, such as a CRC check sum, based on the contents of the frame. In one example, a framer ("FR") 193 assembles contiguous sets of bits into a frame. In particular, framer 193 establishes a Frame Check Sequence ("FCS") code in one of the fields of the frame. The FCS is used to detect errors using, for example, a Cyclic Redundancy Check ("CRC") checksum. The FCS is usually compared against the CRC checksum to determine whether an error exists. To illustrate, consider that an Ethernet frame includes header bits and up to 1500 bytes of payload, with a CRC checksum appended as 32 contiguous bits to the end of the payload. Generally, the CRC checksum is appended in one or more upper layers 160 above physical layers 178. As shown, physical layers 178 include low-level error recovery functions ("Err Rec") 190 and 192 for enabling low-level layer 162—at least in some embodiments—to entirely perform low-level error detection and/or correction ("error recovery") regardless of any upper-level error recovery mechanisms employed in upper layers 160. In at least one embodiment, low-level error recovery functions ("Err Rec") 190 and 192 operate prior or while application data bits are temporarily stored in a buffer (e.g., a FIFO buffer)

for determining, in whole or in part, whether an error occurred. If no error exists, the application data bits are released to an upper layer for assemblage into subsets of bits. The subset of bits can represent an Ethernet frame, an address for a memory device, and the like. In a specific embodiment, the demarcation between physical layer 178 and data link layer 176 is a point at which the application data bits exit the buffer to an upper layer for assemblage.

Regardless of which layer implements conventional error detection and correction, traditional upper-level error detection codes are generally transmitted in a fashion similar to the payload. But according to various embodiments of the invention, low-level error detection codes and payload can be line-decoded differently, such as in parallel. As used herein, the term payload generally refers to a collection of bits (i.e., application data bits) that represent program instructions and/or program data used by an application layer. Application data bits constitute application data, and therefore do not contain control codes for controlling data bit transfer. By contrast, error detection codes are control codes as they are used to control the reliability of data bit transmissions. So according to the various embodiments, portions of both application data bits and error detection bits are line-decoded (and line-encoded) at or nearly at the same time (especially if the portions of both application data bits and error detection bits are together encoded to form a code word). In one embodiment, line decoders operate to translate a code word to form decoded data bits, which can include both application data bits and physical interface ("PI") error detection bits. Note that traditional upper-level error detection codes are transmitted in groups of contiguous bits (e.g., the bits of a CRC checksum for an Ethernet frame) and are usually passed through line decoders to upper-layers in a manner similar to application data bits. Consequently, conventional upper-level error detection techniques generally expend computational resources to line-decode control codes for error recovery at upper-layers, whereas in accordance with at least a specific embodiment of the invention, low-level error detection codes need not be communicated anywhere other than to the physical layer.

FIG. 2A is a block diagram of a circuit implementing error recovery modules to provide reliable data transfers at physical interfaces that use embedded clocks and clock recovery, according to one embodiment of the invention. In the example shown, an electronic device 200 includes physical interfaces 210 and 230 that are composed of an RX error recovery module ("ERM") 212 and RX error recovery module ("ERM") 232, respectively. Note that while FIG. 2A omits physical interfaces that include TX error recovery modules, an ordinarily skilled artisan



should appreciate that such TX error recovery modules can also be implemented, but with inverse functionality. In some cases, physical interfaces 210 and 230 are formed as circuits within, for example, an input/output (“I/O”) ring 202. I/O ring 202 includes input and output terminals 201 for coupling conductors, such as bonding wires, to an external source of signals (not shown), such as an integrated circuit transmitting signals to a core circuit 204. RX error recovery module 212 and RX error recovery module 232 each are coupled to a core circuit 204 to deliver signals thereto. As used in some embodiments, the term core circuit refers to any kind of circuit that performs one or more core functions and can be configured to receive inputs or transmits output via physical interfaces 210 and 230. In some embodiments, core circuit 204 can be formed from an intellectual property (“IP”) core, which can be expressed in a Hardware Description Language (“HDL”) code (e.g., VHDL or Verilog) or as a netlist file. Accordingly, core circuit 204 is portable and can be easily inserted into most designs benefiting from low-level layer error detection and correction provided by physical interfaces 210 and 230. As an example, consider that core circuit 204 is any circuit, such as a dynamic random access memory (“DRAM”), a microcontroller or processor circuit, or any type of circuit that would benefit from implementations of error detection and corrections at physical interfaces. Advantageously, low-level error detection and corrections operates in a manner that is transparent to core circuit 204 (i.e., physical interface error recovery need not require resources of core circuit 204 to perform error detection and/or correction).

Physical interface 210 is a serial physical interface configured to serially receive  $N+m$  encoded bits into a line decoder-error detector (“LD/ED”) 214, which is configured to decode some or all of the  $N+m$  encoded bits to form  $N$  decoded application data bits. The value “ $m$ ” represents the difference in the number of bits from the number of encoded data bits and the number of application data bits. For example, an 8B/10B line encoder has an  $m$  of 2, which is the difference. Line decoder-error detector 214 operates to line decode encoded bits and then detect an error, if one is detectable. If line decoder-error detector 214 detects an error, then it instructs an error corrector (“EC”) 216 to take appropriate error recovery measures. In one embodiment, error corrector 218 is configured to transmit an error indication signal 218 to an external IC or device to alert the external IC or device so it can resolve the error, such as by retransmitting the  $N+m$  encoded bits to physical interface 210. Advantageously, physical interfaces 210 and 230 provide error detection and correction for core circuit 204 without

modifying core circuit 204, thereby conserving engineering resources required to integrate either error detection or correction into core 204. In another embodiment, error corrector 218 is configured to transmit an error indication signal 220 into core circuit 204 so that it can correct the error. In this case, core circuit 204 includes application-specific logic 244 to correct any detected errors. Advantageously, error handling by application-specific logic 244 in core circuit 204 (or elsewhere) therefore enables error detection and correction without requiring the transmitting IC or device (not shown) to help correct the error. This minimizes the number of signal conductors and/or pins needed by electronic device 200, especially if it is encapsulated in a semiconductor package. In still yet another embodiment, error corrector 216 of physical interface 210 can correct errors without sending either error indication signals 218 or 220. For example, the error corrector can implement forward error correction techniques, which automatically corrects errors.

Next, consider that the external IC, for example, is a DRAM controller (not shown) and core circuit 204 is a DRAM memory. Then, physical interface 230 can provide a chip-to-chip interface to exchange error-corrected addresses and data between the DRAM controller and the DRAM memory (i.e., core circuit 204) without one or more of them being modified (functionally or structurally) to implement error detection and correction. In some embodiments, physical interfaces 210 and 230 are formed on a first substrate, which contains an input-output ("I/O") ring 202. Core circuit 204 is formed on a second substrate. If the first and second substrates are separate, then physical interfaces 210 and 230 can be packaged separately from core circuit 204. But if they are formed on the same substrate, then they can be formed as part of a system-on-chip ("SOC") and can be encapsulated together by a single package. In alternative embodiments, physical interface 230 is a parallel physical interface configured to receive  $N+m$  encoded bits into RX error recovery module 232 and then decode some or all of the  $N+m$  encoded bits to form  $N$  decoded application data bits. FIG. 2A omits the details of line decoder-error detector and error corrector for RX error recovery module 232 to simplify the discussion. Note that while FIG. 2A depicts physical interfaces 210 and 230 providing in-bound data bits (i.e., in-bound toward core 204) in an in-bound direction 287, those same physical interfaces can provide for out-bound data bit transmissions (not shown) from core 204 in an out-bound direction 289. Note that both serial physical interface 210 and parallel physical interface 230 shown in FIG. 2A are configured to line decode in-bound data bits to, among other things,

recover embedded clocks therein. In other embodiments either serial physical interface 210 or parallel physical interface 230, or both, can be configured to receive an external clock rather than an embedded clock.

FIG. 2B is another block diagram of the circuit of FIG. 2A implementing error recovery modules to provide reliable data transfers at physical interfaces that use external clocks, according to one embodiment of the invention. Unlike serial physical interface 210 or parallel physical interface 230 in FIG. 2A, electronic device 200 of FIG. 2B includes a serial physical interface 250 configured to receive an external clock ("Ext. CLK") 274. Electronic device 200 also includes a parallel physical interface 270 configured to receive an external clock ("Ext. CLK") 254. Since both physical interfaces 250 and 270 can receive external clock signals, these interfaces need not include clock recovery circuits since embedded clocking techniques need not be employed. As such, error recovery modules 252 and 272 need not be disposed after a line decoder or a line decoding process. Note that elements in FIG. 2B each can have an equivalent functionality and/or structure as elements similarly-numbered in FIG. 2A.

FIG. 3 is a block diagram showing a physical interface configured to detect and optionally correct errors, according to a specific embodiment of the invention. A physical interface 300 includes a physical layer ("PHY") line decoder 302 and an error recovery module 310, which is composed of an error detection bit extractor 312, an error detector 314 and an error corrector 316. Error detection bit extractor 312 is at least functionally cooperative with the line decoding process of PHY line decoder 302 and, as such, it is shown in FIG. 3 to reside within PHY line decoder 302. In this example, PHY line decoder 302 is at least configured to receive  $N+m$  encoded bits (i.e., symbols or code words) of a bit stream and to decode all or some of those bits to form  $N$  decoded application data bits. For example, PHY line decoder 302 can operate to receive 10, 14, 20, 66, or 130 encoded application data bits (or " $N+2$ " bits), and then can respectively decode those bits to form 8, 12, 18, 64 or 128 decoded application data bits (or " $N$ " bits). The two bits are traditionally used for at least clock recovery and DC-balancing. In various embodiments, PHY line decoder 302 can operate to receive any number of encoded data bits and then decode those bits into any other number of application data bits. For example, PHY line decoder 302 can decode 80 bits of encoded data to form 64 decoded application data bits.

In a specific embodiment, PHY line decoder 302 decodes the  $N+m-1$  encoded bits, which is a subset of the  $N+m$  bits, to yield  $N$  decoded application data bits and  $E$  error detection bits. Then, error detection bit extractor 312 extracts one or more physical interface error detection bits for low-level error detection purposes. For example, error detection bit extractor 312 can extract a single bit from the quantity of  $N+m$  (e.g., 10 bits) to yield  $N+m-1$  bits (e.g., 9 bits) decoded bits, which include  $N$  decoded application data bits and  $E$  decoded error detection bits. As such, PHY line decoder 302 is configured to receive  $N+m$  bits, and to perform line decoding, which yields  $N$  application data bits and at least one error detection bit. Advantageously, PHY line decoder 302 can be implemented to supplement or replace a conventional 10B/8B decoder, which uses a commonly-used symbol size of 10 bits for the encoded bits, with the decoded application data bits having a bit size of 8 bits. In various embodiments of the invention, PHY line decoder 302 examines a 9<sup>th</sup> bit (i.e., one of the two bits of the difference between 10 encoded and 8 decoded application data bits) to detect errors. A suitable PHY line decoder 302 is an  $N+1$  bit/ $N$  bit decoder as described in U.S. Patent No. 6,747,580 titled "Method and Apparatus for Encoding or Decoding Data in accordance with an  $NB/(N+1)B$  Block Code and Method for Determining such a Block Code," which is incorporated by reference in its entirety for all purposes. Note that while FIG. 3 depicts a PHY line decoder 302, one having ordinary skill in the art should appreciate that a PHY line encoder that encodes  $N$  application data bits and  $E$  error detection bits into  $N+m$  encoded bits (i.e., an  $N$  bit/ $N+m$  bit encoder) can be constructed to operate in an inverse manner described in FIG. 3 and elsewhere herein.

After extraction, error detection bit extractor 312 then sends the physical interface ("PI") error detection bit ("EDB") 313 to error detector 314, which uses that bit to determine whether the  $N+m$  encoded data bits include one or more erroneous data bits. Error detector 314 is configured to detect errors in accordance with any number of error detection techniques and codes. In one embodiment, error detector 314 is configured as either an even or odd parity checker, with the physical interface error detection bit being a parity bit (i.e., either an even or odd parity bit). Advantageously, the physical interface error detection bits are received at PHY line decoder 302 concurrent with subsets of application data bits that undergo decoding. This can decrease the amount of time required for PHY line decoder 302 to receive the error-detection code in comparison to conventional error detection techniques that append an error detection

code to the end of a group of encoded data bits, where the group can have a bit size much greater than  $N+m$  bits. In another embodiment, the physical interface error detection bit is a portion (e.g., one bit) of a CRC checksum. In this case, an error-detection code generator in a transmitting physical interface (not shown) is configured to generate the physical interface error detection bit as part of a CRC checksum. As is discussed below in FIG. 6, an error-detection code generator (or equivalent) can provide the bits of the CRC checksum to a PHY line encoder (not shown), which inserts the physical interface error detection bit with  $N$  unencoded application data bits to yield  $N+2$  encoded data bits. Consequently, error detector 314 operates to accumulate or collect physical interface error detection bits from each subset of decoded  $N+m$  data bits to reconstruct the CRC checksum as an error-detection code. For example, consider that for each set of 10 encoded bits being decoded by PHY line decoder 302, one bit of a CRC checksum is provided as a physical interface error detection bit for each set of 8 decoded application data bits. If a 16-bit CRC checksum is implemented, then for every 16 sets of 8 decoded application data bits (i.e., for every 128 bits) error detector 314 CRC can reconstruct the 16-bit CRC checksum. Then, error detector 314 generates a CRC checksum from those 128 bits and then compares that CRC checksum against the reconstructed error-detection code to determine whether an error has occurred. Error detector 314 can implement other error detection techniques, such as forward error correction. Examples of forward error correction techniques include Reed-Solomon codes, Hamming codes, and Bose-Chaudhuri-Hocquenghem ("BCH") codes.

Once error detector 314 determines that an error has occurred during data bit transmission, then it signals to error corrector 316 that such an error has occurred. In some cases, error corrector 316 operates to request a retransmission of the decoded application data bits for which an error has been detected. Continuing with the previous example, if the 16-bit CRC checksum comparison indicates that an error exists in the 128 decoded application data bits, then error corrector 316 can request that the transmitting physical interface resend those 128 bits. In other cases, error corrector 316 can activate an application-specific circuit to perform user-defined error handling. In at least one instance, error corrector 316 can take some sort of default error-correcting action, such as reinitializing an associated IC, like a core circuit described above.

FIG. 4 is a functional block diagram illustrating line encoding in a transmitter and line decoding in a receiver with error-detection bit insertion and extraction, according to one

embodiment of the invention. As shown in diagram 400, transmitter physical interface ("TX") 410 includes a PHY line encoder 412, an error detection code generator 416, an error bit inserter 418 and an optional buffer 421 for maintaining application data bits during generation of error detection codes, for example. In this example, error detection code generator 416 generates an error detection code based on application data bits in buffer 421, such as a 16-bit CRC checksum. Next, an error bit inserter 418 selects at least one error-detection bit ("EDB") 414 from the 16-bit CRC checksum and then inserts it with an out-going set of N application data bits from buffer 421 to form 9 bits. Then, PHY line encoder 412 encodes the 9 bits into 10 encoded bits (or symbols). In some embodiments, PHY line encoder 412 encodes error-detection bit 414 in a manner that embeds an asynchronous clock in the 10 encoded bits as well as maintaining a zero DC offset and/or optimal spectral characteristics for those same encoded bits. In a specific embodiment, PHY line encoder 412 operates in accordance with coding techniques set forth in U.S. Patent 6,747,580 whereby PHY line encoder 412 encodes the N bits of application data into N+1 encoded bits. For example, consider N is the number 9. As such, PHY line encoder 412 can encode 8 application data bits and 1 error bit (i.e., 9 bits) into 10 encoded bits without affecting the data rate associated with 8B/10B coding schemes. Note that in FIG. 4, N can represent any non-negative integer.

Transmitter physical interface 410 then sends the encoded bits via physical medium 420 to error bit extractor 422 of a receiving physical interface ("RX") 440. Receiving physical interface ("RX") 440 includes a PHY line decoder 428, an error bit extractor 422, error detection code reconstructor 426, an error detector 430, a comparator 434, and a buffer 431, which can maintain the decoded application data bits pending confirmation that no errors are detected. PHY line decoder 428 decodes the 10 encoded data bits into 9 decoded bits, which includes 8 decoded application data bits and 1 decoded error detection bit ("EDB") 424. In various embodiments, PHY line decoder 428 includes clock data recovery circuits (not shown) to recover the embedded clock transmitted over physical medium 420 with the encoded application data bits. Error bit extractor 422 extracts error-detection bit 424 from the 9 decoded bits and provide it to error detection code reconstructor 426, which accumulates the error detection bit with other physical interface error detection bits. Next, error detection code reconstructor 426 reconstructs the error detection code, which was originally generated by error detection code generator 416. Next, error detector 430 generates a CRC checksum 432 based on a group of N

decoded application data bits in buffer 431. Then, comparator 434 can operate to compare whether CRC checksum 432 matches error detection code 426. If it does, then there is no error; if it does not, then there is an error.

FIG. 5 depicts one example of a flow for line encoding a group of data bits as a cell in accordance with one embodiment of the invention. In some implementations, a transmitting physical interface can transmit data as shown in flow 550, whereby groups 560 of data bits can be referred to as a cell (as well as a frame or the like). Cell 560, for example, can include any number of control bits 561 (e.g., x number of bits) and two application data words 563 of 64 bits each. An upper-layer device (not shown), which resides above a PHY line encoder in the physical layer, can be configured to break cell 560 into discrete sets 562 of uncoded (i.e., precoded or unencoded) bits for use by a PHY line encoder. Each set 562 has N bits (e.g., 8 bits). An error detection code generator (not shown) generates an error detection code 570 based on cell 560 to form an error detection code as a 12-bit CRC, for example. The PHY line encoder then can insert a physical interface error detection bit ("EDB") 572 into the uncoded bit stream in or near sets 562 to form sets 574 of N+E bits (e.g., N can be 8 bits and E can be 1 bit). The PHY line encoder then can translate each set 572 of N+E bits into 10-bit code words 580 (e.g., encoded data bits) for transmission over a data link, such as a serial data link. Although not shown, a PHY line decoder at a receiving physical interface would operate in a similar, but inverse manner.

FIG. 6 is a block diagram 600 illustrating physical interfaces that provide for error detection and correction by at least retransmitting data bits, according to at least one specific embodiment of the invention. Physical interfaces 602 and 652 provide for physical connectivity through a high-speed serial communications link 650. Physical interface 602 includes storage, such as a first in, first out buffer ("FIFO") 604, for storing data bits that are to be transmitted. FIFO 604 maintains the stored data bits to compute, for example, an error detection code over a group of data bits. In some cases, FIFO 604 stores data bits for retransmission if an error occurs. Physical interface 602 also includes PHY line encoder ("PHY encoder") 610, a serializer ("SER") 612 for serializing parallel data bits, an error-detection code generator 616 and a transmitter ("TX") error corrector 618. Based on the contents of the group of data bits in FIFO 604 at a specific time, error detection code generator 616 generates an error detection code, such as a CRC checksum. Error detection bit inserter 614 decomposes the CRC checksum and inserts

at least one bit (e.g., one E bit) as a physical interface error detection bit ("EDB") with the N unencoded bits 606. In operation, PHY line encoder 610 encodes N+E bits to form N+2 encoded data bits 608. Serializer 612 then serializes N+2 data bits 608 prior to transmission over link 650. Receiving physical interface 652 includes a deserializer 654 to convert N+2 data bits 608 to parallel data bits. PHY line decoder ("PHY decoder") 656 decodes the encoded bits stream and stores the decoded data bits in FIFO 658. Error detection bit ("EDB") extractor 660 extracts at least one bit as physical interface error detection bit ("EDB") 662 and sends it to error detector 666. When error code reconstruction is complete, error detector 666 operates to determine whether an error has occurred. If error detector 666 detects no errors, then it instructs FIFO 658 to release the decoded data bits for further processing. But if error detector 666 detects an error, then it indicates the presence of an error to a receiver ("RX") error corrector 668. In one instance, RX error corrector 668 sends a request 669 to instruct TX error corrector 618 to retransmit the contents, or a portion thereof, via link 650. In another instance, RX error corrector 668 can send a request 671 to an application-specific circuit (not shown) to handle the error. In yet another instance, receiver error corrector 668 can correct the error using forward error correction techniques or the like.

FIG. 7 illustrates an alternative receiver for implementing error detection and correction without line decoding, according to at least one specific embodiment of the invention. In receiver ("RX") 700, a transmitting physical interface (not shown) sends data bits via link 702. Error bit extractor 704 extracts error detection data bits ("EDBs") 706 from the data bits and collects them to reconstruct an error detection code 708. Unlike diagram 400 of FIG. 4, receiver 700 does not use a PHY line decoder as does receiver 440. As such, receiver 700 need not recover a clock signal from an embedded clock. As shown, receiver 700 is configured to receive an external clock 701 from, for example, a transmitting device (not shown). It is the transmitting device that, for example, appends error detection bits to application data bits prior to transmission over link 702. Receiver 700 includes an error detector 780 for computing an error detection code, or in this case, a CRC checksum 722 based on a group of application data bits in FIFO 720. A comparator 724 (e.g., in error detector 720) then determines whether a transmission error has occurred. If an error exists, then an error corrector 780 can take corrective action by, for example, either requesting the transmitting device to retransmit the data bits or signaling application-specific logic that it is to correct the error.



FIG. 8 depicts a block diagram of an error corrector according to one embodiment of the invention. Error corrector 800 in this example includes an error corrector controller 822 to control one or more functions that can resolve errors, with the specific functions being programmed by a user for specific applications. For example, loss of pixel data in video memory application is not as crucial as the loss of data in other applications. As such, error corrector controller 822 can select any of the following modules to implement error correction. Retransmission request module 824 generates a request bound for the transmitting physical interface to resend the data bits upon detection of an error. Application-specific recovery module 814 signals an application-specific circuit to initiate its custom error-handling routines, as prescribed by a designer. But error corrector controller 822 can select a default error recovery scheme, such as reinitializing an IC upon detection of an error. This will cause both physical interfaces to reinitialize to a known state. In various alternative embodiments, error corrector 800 can include a frequency selector 830, an amplitude selector 840, and a coding selector 850. If signals on the communication link between the physical interfaces falter, then frequency selector 830 can adjust the data transfer rate over the link to reduce the error rate. Or, amplitude selector 840 can adjust the signal amplitude until the error rate is curtailed. Coding selector 850 can be used to change the line coding, for example, from 18B/20B to 8B/10B to reduce the time during which errors are detected. For example, consider a CRC of 32-bits is implemented for error recovery, with one bit of the CRC being sent at a time. With 18B/20B, it takes 576 application data bits (8 bits) to form a CRC of 32 bits (i.e.,  $8 * 32 = 576$  bits), whereas with 8B/10B it takes 256 application data bits. As such, coding selector 850 can increase the rate at which errors codes (e.g., CRCs) are transmitted over a communications channel. Error corrector controller 822 can chose these selectors individually or in combination.

The foregoing description, for purposes of explanation, used specific nomenclature to provide a thorough understanding of the invention. However, it will be apparent to one skilled in the art that specific details are not required in order to practice the invention. In fact, this description should not be read to limit any feature or aspect of the invention to any embodiment; rather features and aspects of one embodiment may readily be interchanged with other embodiments. For example, although the above descriptions of the various embodiments relate to physical layer line decoders, the discussion is applicable to physical layer line encoders and all types of communications links, as well as many types of communications protocols, such

as Gigabit Ethernet, HDMI, TMDS, DVI and SATA. Also, the various embodiments are applicable to point-to-point communications between electronic devices as well as broadcasting from one to many. While some of the above discussions relate to serial data communication, the various embodiments of the invention can apply to parallel data communications as well.

Thus, the foregoing descriptions of specific embodiments of the invention are presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed; obviously, many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications; they thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. Notably, not every benefit described herein need be realized by each embodiment of the invention; rather any specific embodiment can provide one or more of the advantages discussed above. It is intended that the following claims and their equivalents define the scope of the invention.

**Claims for foreign applications:**

1. An apparatus for detecting errors in a physical interface that facilitates data communications between integrated circuits ("ICs"), the apparatus comprising:  
a decoder configured to decode a subset of encoded data bits to yield decoded data bits;  
and  
a physical interface ("PI") error detection bit extractor configured to extract a physical interface error detection bit from said decoded data bits, said physical interface error detection bit being used to determine whether said encoded data bits include at least one erroneous data bit as an error.
2. The apparatus of claim 1 further comprising:  
an error detector configured to operate within a physical layer to detect said error,  
wherein said subset of said encoded data bits includes an embedded clock.
3. The apparatus of claim 2 wherein said error detector performs error detection earlier than at a data link layer or at any upper layer of the Open Systems Interconnection ("OSI") seven layer reference model above said physical layer.
4. The apparatus of claim 2 wherein said physical interface error detection bit is a parity bit, said error detector being configured to check a parity of said encoded data bits and to detect an error if said parity is noncompliant.
5. The apparatus of claim 2 wherein said physical interface error detection bit is a portion of an error-detection code and wherein said error-detection code is a cyclic redundancy check ("CRC") checksum, said error detector being configured to accumulate said physical interface error detection bit and other physical interface error detection bits from other subsets of said encoded data bits to reconstruct said error-detection code.
6. The apparatus of claim 1 further comprising:  
an N bit-to-N+2 ("N bit/N+2") bit physical layer ("PHY") encoder configured to:

(1) insert a physical interface error detection bit with N application data bits to form N+1 unencoded data bits, and

(2) encode said N+1 unencoded data bits to yield N+2 encoded data bits, wherein N represents any integer number of data bits; and

an error-detection code generator configured to generate a number of bits constituting an error-detection code that includes said physical interface error detection bit.

7. A system comprising:

a first integrated circuit (IC) coupled through a physical medium to a second IC,

the first IC including a physical interface formed on a first substrate portion of the first IC to detect transmission errors in data exchanged with the second IC formed on a second substrate portion, the system further comprising:

a decoder configured to decode a subset of encoded data bits to yield decoded data bits; and

a physical interface error detection bit extractor configured to extract a physical interface error detection bit from said decoded data bits, said physical interface error detection bit being used to determine whether said encoded data bits include at least one erroneous data bit as an error.

8. The system of claim 7, wherein the physical interface includes:

a plurality of input ports and output ports, including a first subset of input ports configured to receive in-bound encoded data bits and a first subset of output ports configured to transmit in-bound decoded data bits to said another IC; and

one or more error recovery modules coupled between said plurality of input ports and output ports,

wherein a first error recovery module of said one or more error recovery modules is coupled between at least one of said first subset of input ports and at least one of said first subset of output ports.

9. The system of claim 8 wherein said first error recovery module includes a physical layer (“PHY”) decoder configured to detect errors in said in-bound encoded data bits and to initiate an action to correct said errors.

10. The system of claim 9 wherein said plurality of input ports and output ports include a second subset of input ports configured to receive out-bound unencoded data bits from said another IC and a second subset of output ports configured to transmit out-bound encoded data bits, wherein a second error recovery module of said one or more error recovery modules is coupled between at least one of said second subset of input ports and at least one of said second subset of output ports, said second error recovery module including a PHY encoder configured to generate error-detection codes for said out-bound encoded data bits.

11. The system of claim 7, further comprising:

a transmitter device coupled to a first end of said physical medium, said transmitter device includes an  $N$  bit/ $N+2$  bit encoder that generates encoded data bits with physical interface error detection bits encoded therein; and

a receiver device coupled to a second end of said physical medium, said receiver device includes:

an  $N+2$  bit/ $N$  bit decoder that decodes said encoded data bits, and

an error detector configured to determine an error using said physical interface error detection bits.

12. The system of claim 11 wherein said transmitter device is a transition-minimized differential signaling (“TMDS”) transmitter and a high-definition multimedia interface (“HDMI”) transmitter and said receiver device is a TMDS receiver and a HDMI receiver.

13. A method for decoding data bits to at least detect errors at a physical interface, the method comprising:

decoding a subset of an encoded bit stream having an embedded asynchronous clock to yield decoded data bits;

extracting a physical interface error detection bit from said decoded data bits;

determining said encoded bit stream includes an incorrect bit based on at least said physical interface error detection bit; and  
correcting said error at said physical interface.

14. The method of claim 13 wherein determining said subset of said encoded bit stream includes said incorrect bit further comprises:

determining a parity of a subset of said encoded bit stream with said physical interface error detection bit; and

indicating said error is present if said parity does not comply with a predetermined parity.

15. The method of claim 13 wherein determining said subset of said bit stream includes said incorrect bit comprises:

combining said physical interface error detection bit with other physical interface error detection bits to form a first error detection code;

generating a second error detection code based on application data bits from said decoded data bits and other application data bits from other decoded data bits decoded from other subsets of said bit stream;

comparing said first error detection code against said second error detection code; and

indicating said error is present if said second error detection code does not match said first error detection code.

16. An apparatus for generating error-detection codes in a physical interface for the transmission of data communications between integrated circuits ("ICs"), the apparatus comprising:

an N bit-to-N+2 ("N bit/N+2") bit physical layer ("PHY") encoder configured to:

insert a physical interface error detection bit with N application data bits to form N+1 unencoded data bits, and

encode said N+1 unencoded data bits to yield N+2 encoded data bits; and

an error-detection code generator configured to generate a number of bits constituting an error-detection code that includes said physical interface error detection bit,

wherein N represents any integer number of data bits.

17. A physical interface formed as an integrated circuit ("IC") on a first substrate portion to detect transmission errors in data exchanged with another IC formed on a second substrate portion, the physical interface comprising:

a plurality of input ports and output ports, including a first subset of input ports configured to receive in-bound encoded data bits and a first subset of output ports configured to transmit in-bound decoded data bits to said another IC; and

one or more error recovery modules coupled between said plurality of input ports and output ports,

wherein a first error recovery module of said one or more error recovery modules is coupled between at least one of said first subset of input ports and at least one of said first subset of output ports.

18. A high-speed serial communications link between at least two integrated circuits ("ICs") comprising:

a physical medium for conveying communications data;

a transmitter device coupled to a first end of said physical medium, said transmitter device includes an  $N$  bit/ $N+2$  bit encoder that generates encoded data bits with physical interface error detection bits encoded therein; and

a receiver device coupled to a second end of said physical medium, said receiver device includes:

an  $N+2$  bit/ $N$  bit decoder that decodes said encoded data bits, and

an error detector configured to determine an error using said physical interface error detection bits.

19. A physical interface for detecting transmission errors at a physical layer during point-to-point communications via parallel data link, the physical interface comprising:

a parallel port for receiving parallel communications data via said parallel data link, a subset of said parallel communications data including a physical interface error detection bit;

an external clock input configured to receive an external clock to synchronize the reception of said parallel communications data; and

a physical interface error detector configured to use said physical interface error detection bit to determine whether said parallel communications data includes an incorrect bit as an error;  
and  
an error corrector configured to correct said error at said physical interface.



# ABSTRACT

An apparatus, system and method for detecting errors in a physical interface during the transmission and/or receipt of data communications between integrated circuits ("ICs") are disclosed. In one embodiment, an apparatus is configured to operate as or within a receiving physical interface. The apparatus includes a decoder configured to decode a subset of encoded data bits to yield decoded data bits. It also includes a physical interface ("PI") error detection bit extractor configured to extract a physical interface error detection bit from the decoded data bits. As such, the apparatus uses the physical interface error detection bit to determine whether the encoded data bits include at least one erroneous data bit as an error. In some embodiments, the apparatus includes an error detector configured to operate within a physical layer. In at least one embodiment, the apparatus efficiently transmits error detection codes within, for example, an  $NB/(N+1)B$  line coder.

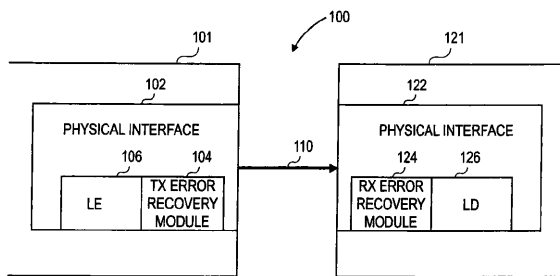


FIG. 1A

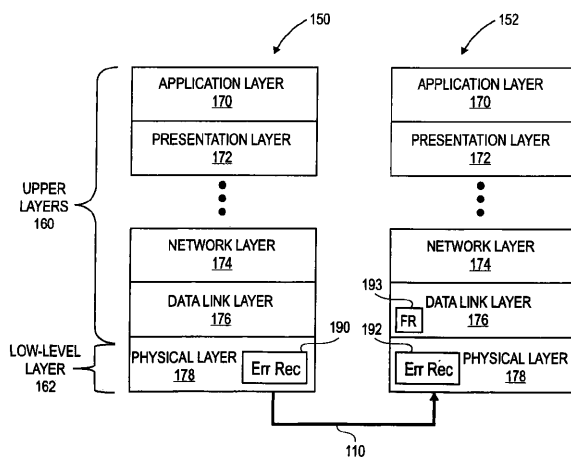


FIG. 1B

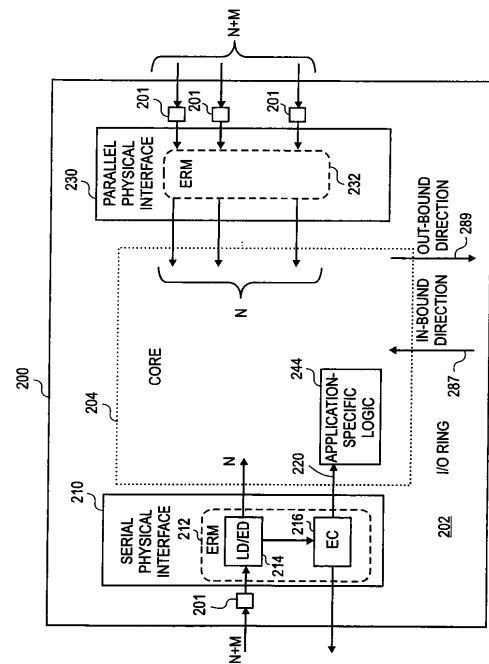


FIG. 2A

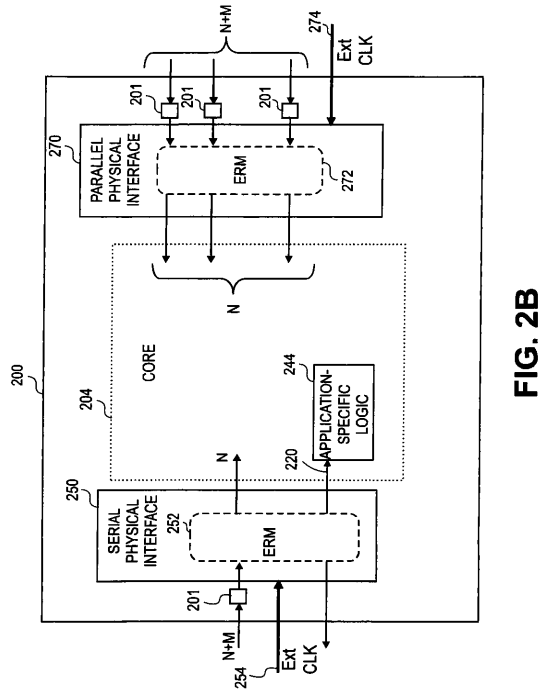


FIG. 2B

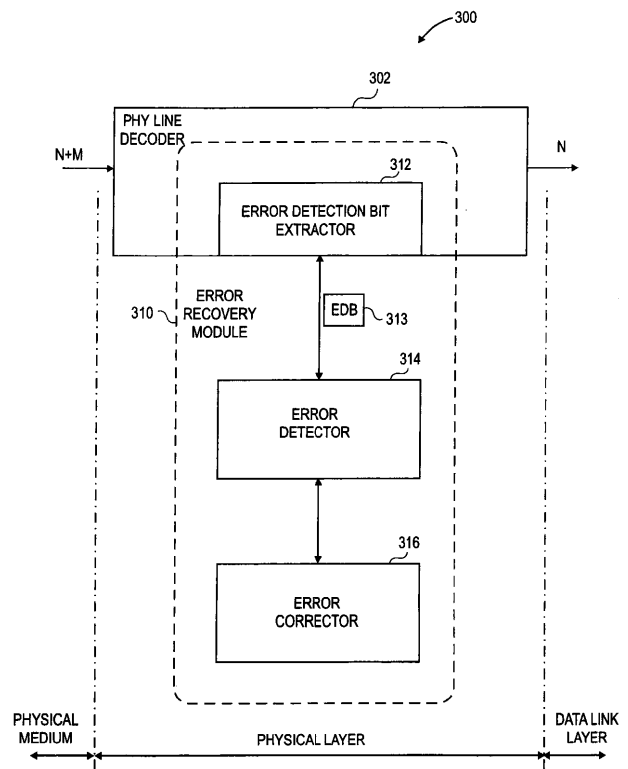


FIG. 3

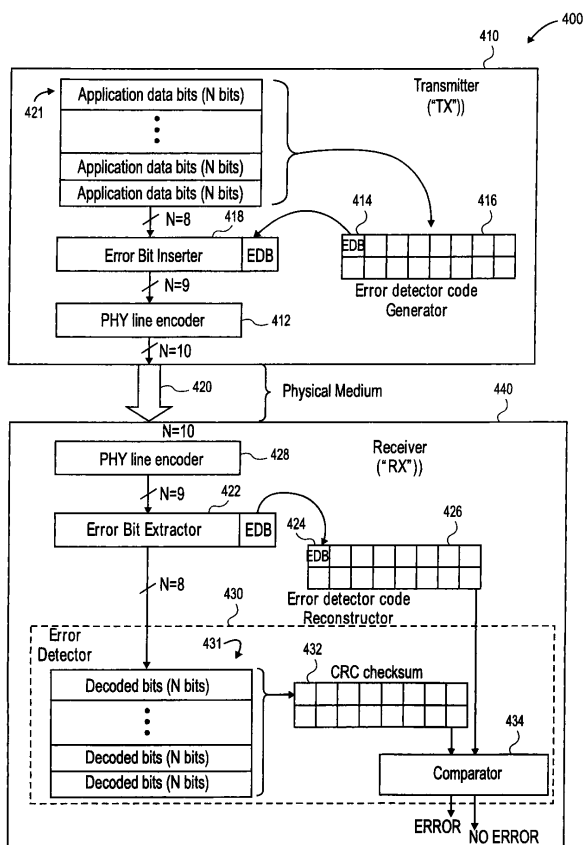


FIG. 4

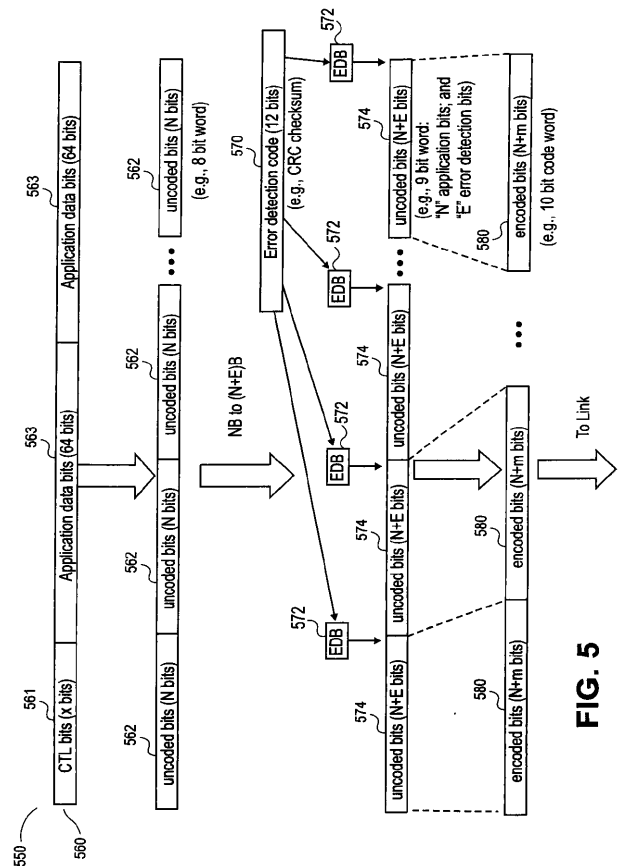


FIG. 5

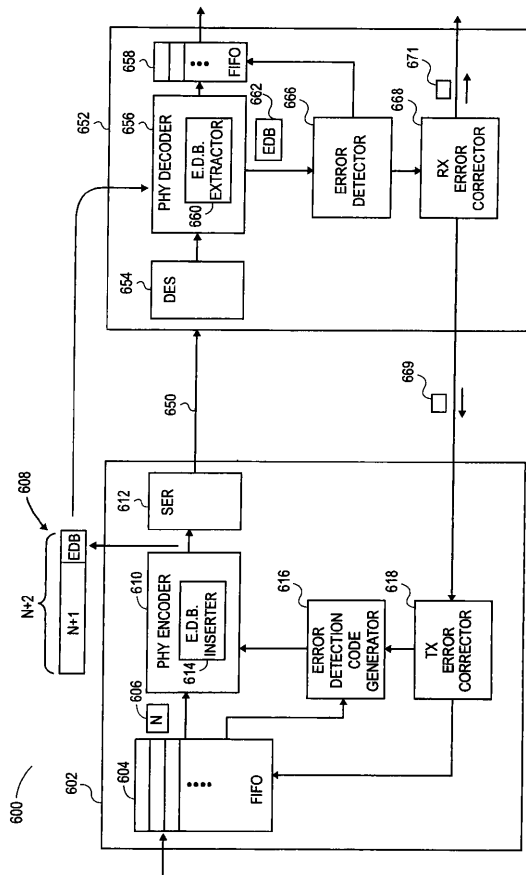


FIG. 6

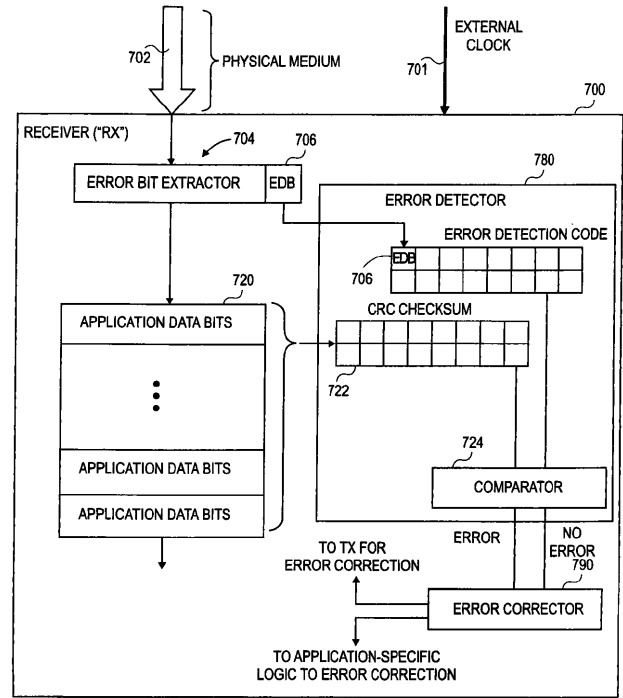


FIG. 7

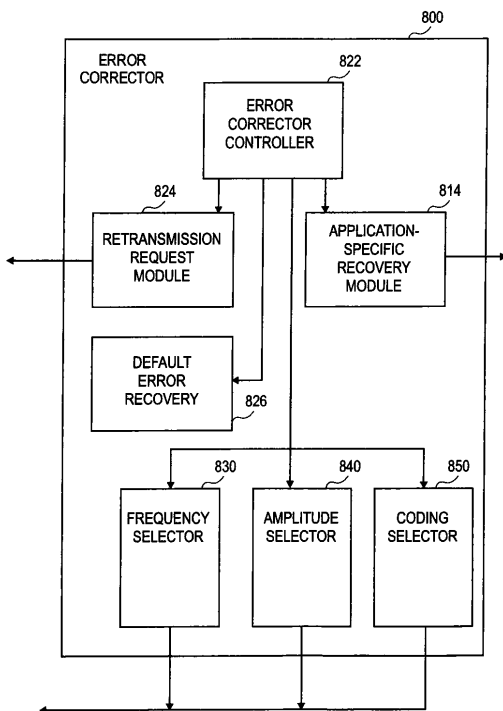


FIG. 8