

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6411480号  
(P6411480)

(45) 発行日 平成30年10月24日 (2018.10.24)

(24) 登録日 平成30年10月5日 (2018.10.5)

(51) Int. Cl.	F I
<b>H04L 1/00 (2006.01)</b>	H04L 1/00 A
<b>H04L 25/02 (2006.01)</b>	H04L 25/02 V

請求項の数 15 (全 39 頁)

(21) 出願番号	特願2016-521313 (P2016-521313)	(73) 特許権者	507364838
(86) (22) 出願日	平成26年10月9日 (2014.10.9)		クアルコム、インコーポレイテッド
(65) 公表番号	特表2016-539533 (P2016-539533A)		アメリカ合衆国 カリフォルニア 921
(43) 公表日	平成28年12月15日 (2016.12.15)		21 サン ディエゴ モアハウス ドラ
(86) 国際出願番号	PCT/US2014/059981		イブ 5775
(87) 国際公開番号	W02015/054548	(74) 代理人	100108453
(87) 国際公開日	平成27年4月16日 (2015.4.16)		弁理士 村山 靖彦
審査請求日	平成29年9月25日 (2017.9.25)	(74) 代理人	100163522
(31) 優先権主張番号	61/889,030		弁理士 黒田 晋平
(32) 優先日	平成25年10月9日 (2013.10.9)	(72) 発明者	仙石 祥一郎
(33) 優先権主張国	米国 (US)		アメリカ合衆国・カリフォルニア・921
(31) 優先権主張番号	61/946,647		21-1714・サン・ディエゴ・モアハ
(32) 優先日	平成26年2月28日 (2014.2.28)		ウス・ドライブ・5775
(33) 優先権主張国	米国 (US)	審査官	阿部 弘

最終頁に続く

(54) 【発明の名称】 C C I e プロトコルを介したエラー検出能力

(57) 【特許請求の範囲】

【請求項 1】

2ラインバスである共有バスと、

前記共有バスに結合されたスレーブデバイスと、

前記共有バスに結合され、前記共有バス上の通信を管理するように適合されたマスタデ  
バイスと

を備え、

前記共有バスを介した前記スレーブデバイスまたは前記マスタデバイスの少なくとも1  
つからの送信が、送信のためにシンボルに次にトランスコーディングされる3進数に符号  
化される複数のビットであり、前記複数のビットにおける3桁の最下位ビットまたは最下  
位のもののいずれかが、前記送信のエラー検出のために使用され、前記トランスコーディングすることが、送信のための現在の順次シンボルが前の順次シ  
ンボルと現在の遷移番号の関数である一時遷移番号とに基づいて取得される変換関数を使  
用することをさらに含み、

前記遷移番号が前記3進数の個々の桁であり、

前記現在の順次シンボルがトランスコーディングされることで、シンボル状態がシンボ  
ルサイクルごとに変化して、クロック情報をシンボル間遷移内に埋め込むとともに前記2  
ラインバスの両方のラインを使用してシンボルを転送する、デバイス。

【請求項 2】

前記スレーブデバイスが、クロック信号内で誤ったクロックパルス消失と誤った余剰ク

10

20

ロックパルスとを検出する、状態機械論理回路を実装する、請求項1に記載のデバイス。

【請求項3】

前記複数のビットが20ビットシーケンスである、請求項1に記載のデバイス。

【請求項4】

エラー検出のための前記3桁の最下位ビットの使用が、前記20ビットシーケンス全体におけるエラーの検出を保証する、請求項3に記載のデバイス。

【請求項5】

第2の最下位ビットおよび第3の最下位ビットの使用が、データ送信またはエラー検出のいずれかに柔軟に割り振られる、請求項1に記載のデバイス。

【請求項6】

エラー検出のための前記最下位ビットの使用が、時間の約50%のみで、20ビットシーケンス全体における単一シンボルエラーの検出を保証する、請求項5に記載のデバイス。

【請求項7】

前記最下位ビットのみがエラー検出のために使用されるとき、前記最下位ビットの値が「1」または「0」のいずれかの一定の2進値に設定される、請求項1に記載のデバイス。

【請求項8】

前記3桁の最下位ビットがエラー検出のために使用されるとき、前記3桁の最下位ビットの値が「000」または「111」のいずれかの一定の2進値に設定される、請求項4に記載のデバイス。

【請求項9】

送信機デバイス上で動作可能な方法であって、  
2ラインバスである共有バスを介して送信されるべき複数のビットを取得するステップであって、前記複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかが、エラー検出のために使用されるステップと、  
前記複数のビットを3進数に変換するステップと、  
前記3進数の桁を順次シンボルに変換するステップであって、  
前記3進数の桁を順次シンボルに変換するステップが、送信のための現在の順次シンボルが前の順次シンボルと現在の遷移番号の関数である一時遷移番号とに基づいて取得される変換関数を使用するステップをさらに含み、  
前記遷移番号が前記3進数の個々の桁であり、  
前記現在の順次シンボルがトランスコーディングされることで、シンボル状態がシンボルサイクルごとに変化して、クロック情報をシンボル間遷移内に埋め込むとともに前記2ラインバスの両方のラインを使用してシンボルを転送する、ステップと、  
複数のシンボルを送信のためにワードに結合するステップとを含む方法。

【請求項10】

前記共有バスを介して受信デバイスへ前記ワードを送信するステップをさらに含む、請求項9に記載の方法。

【請求項11】

前記送信されるビットのためのチェックサムを計算するステップと、  
前記送信されるビットのための前記チェックサムを受信デバイスへ送信するステップとをさらに含む、請求項9に記載の方法。

【請求項12】

チェックサムが、受信デバイスへの送信のために、いずれかの任意のワード内に挿入される、請求項9に記載の方法。

【請求項13】

送信機デバイスであって、  
2ラインバスである共有バスに結合するためのバスインターフェースと、  
前記バスインターフェースに結合される処理回路であって、  
前記共有バスを介して送信されるべき複数のビットを取得することであって、前記複

10

20

30

40

50

数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかが、エラー検出のために使用されること、

前記複数のビットを3進数に変換すること、

前記3進数の桁を順次シンボルに変換することであって、

前記3進数の桁を順次シンボルに変換することが、送信のための現在の順次シンボルが前の順次シンボルと現在の遷移番号の関数である一時遷移番号とに基づいて取得される変換関数を使用することをさらに含み、

前記遷移番号が前記3進数の個々の桁であり、

前記現在の順次シンボルがトランスコーディングされることで、シンボル状態がシンボルサイクルごとに変化して、クロック情報をシンボル間遷移内に埋め込むとともに前記2ラインバスの両方のラインを使用してシンボルを転送する、こと、および

10

複数のシンボルを送信のためにワードに結合すること  
を行うように構成された処理回路と  
を備える送信機デバイス。

【請求項14】

前記処理回路が、

前記共有バスを介して受信デバイスへ前記ワードを送信すること  
を行うようにさらに構成される、請求項13に記載のデバイス。

【請求項15】

受信機デバイスであって、

20

2ラインバスである共有バスに結合するためのバスインターフェースと、

前記バスインターフェースに結合される処理回路であって、

前記共有バスを介して複数のシンボルを受信すること、

前記複数のシンボルを桁に変換すること、

前記桁を3進数に結合すること、および

前記3進数をビットに変換することであって、

前記複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかが、エラー検出のために使用され、前記3進数をビットに変換することが、一時遷移番号が現在の順次シンボルと前の順次シンボルとから取得される変換関数を使用することをさらに含み、

30

前記一時遷移番号が現在の遷移番号の関数であり、

前記遷移番号が前記3進数の個々の桁であり、

前記現在の順次シンボルがトランスコーディングされることで、シンボル状態がシンボルサイクルごとに変化して、クロック情報をシンボル間遷移内に埋め込むとともに前記2ラインバスの両方のラインを使用してシンボルを転送する、こと

を行うように構成された処理回路と

を備える受信機デバイス。

【発明の詳細な説明】

【技術分野】

【0001】

40

関連出願の相互参照

本特許出願は、両方が本出願の譲受人に譲渡され、参照により明示的に本明細書に組み込まれる、2013年10月9日に出願した「Error Correction Capability Over CCle Bus」という名称の米国仮出願第61/889,030号、および、2014年2月28日に出願した「Bit Allocation Over A Camera Control Interface Extended Bus」という名称の米国仮出願第61/946,647号の優先権を主張するものである。

【0002】

本開示は、共有バスを介して効率的な動作を可能にすることに関し、より詳細には、共有カメラ制御インターフェース拡張(CCle)バスを介してエラー検出を容易にすることに関する。

50

## 【背景技術】

## 【0003】

I2C(I<sup>2</sup>Cとも呼ばれる)は、マザーボード、埋込みシステム、携帯電話または他の電子デバイスに低速周辺装置を取り付けるために使用されるマルチマスタシリアルシングルエンドバスである。I2Cバスは、クロック(SCL)ラインと、7ビットアドレス指定を用いるデータ(SDA)ラインとを含む。バスは、デバイスに関する2つの役割、すなわち、マスタおよびスレーブを有する。マスタデバイスは、クロックを生成し、スレーブデバイスとの通信を開始するデバイスである。スレーブデバイスは、クロックを受信し、マスタによってアドレス指定されたときに応答するデバイスである。I2Cバスは、任意の数のマスタデバイスが存在できることを意味するマルチマスタバスである。加えて、マスタおよびスレーブの役割は、メッセージの間で(STOPが送られた後に)変更される場合がある。I2Cは、メッセージの基本タイプを定義し、それらの各々はSTARTで開始し、STOPで終了する。

10

## 【0004】

カメラの実装形態との関連では、一方向送信を用いて、センサから画像を取り込み、そのような画像データをベースバンドプロセッサ内のメモリに送信することができ、一方、制御データは、ベースバンドプロセッサとセンサならびに他の周辺デバイスとの間で交換することができる。一例では、ベースバンドプロセッサとイメージセンサ(および/または1つもしくは複数のスレーブデバイス)との間のそのような制御データのために、カメラ制御インターフェース(CCI)プロトコルが使用される場合がある。一例では、CCIプロトコルは、イメージセンサとベースバンドプロセッサとの間のI2Cシリアルバスを介して実装される場合がある。

20

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0005】

バスを共有するデバイス間の送信におけるエラー検出を提供することが望ましいであろう。

## 【課題を解決するための手段】

## 【0006】

共有バスと、スレーブデバイスと、マスタデバイスとを備えるデバイスを提供する。スレーブデバイスは、共有バスに結合され得る。マスタデバイスは、共有バスに結合され、共有バス上の通信を管理するように適応され得る。共有バスを介した送信は、送信のためにシンボルに次にトランスコーディングされる3進数に符号化される複数のビットであり、複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかが、送信のエラー検出のために使用される。

30

## 【0007】

共有バスは2ラインバスであってよく、2ラインバスの両方のラインが、シンボルを転送するために使用される。クロック信号は、シンボル間遷移内に埋め込まれ得る。スレーブデバイスは、クロック信号内で誤ったクロックパルス消失と誤った余剰クロックパルスとを検出する、状態機械論理回路を実装することができる。

## 【0008】

40

一例では、複数のビットは20ビットシーケンスであり得る。エラー検出のための3桁の最下位ビットの使用は、20ビットシーケンス全体におけるエラーの検出を保証する。加えて、第2の最下位ビットおよび第3の最下位ビットの使用が、データ送信またはエラー検出のいずれかに柔軟に割り振られ得る。3桁の最下位ビットがエラー検出のために使用されるとき、3桁の最下位ビットの値が「000」または「111」のいずれかの一定の2進値に設定され得る。

## 【0009】

エラー検出のための最下位ビットの使用は、時間の約50%のみで、20ビットシーケンス全体における単一シンボルエラーの検出を保証する。一例では、最下位ビットのみがエラー検出のために使用されるとき、最下位ビットの値が「1」または「0」のいずれかの一定

50



の2進値に設定される。

【0010】

一態様によれば、バスインターフェースと処理回路とを備える送信機デバイスが提供される。バスインターフェースは、送信機デバイスを、(受信またはスレーブデバイスが結合される)共有バスに結合するように働き得る。送信機デバイスは、共有バスを介した通信を管理または制御することができる。処理回路は、(a)共有バスを介して送信されるべき複数のビットを取得することであって、複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかが、エラー検出のために使用されること、(b)複数のビットを3進数に変換すること、(c)3進数の桁を順次シンボルに変換すること、(d)複数のシンボルを送信のためにワードに結合すること、および/あるいは(e)共有バスを介して受信デバイスへワードを送信することを行うように構成され得る。クロックは、シンボル間遷移内に埋め込まれ得る。共有バスは2ラインバスであってよく、2ラインバスの両方のラインが、シンボルを転送するために使用される。処理回路は、(a)送信されるビットのためのチェックサムを計算すること、および/または(b)送信されるビットのためのチェックサムを受信デバイスへ送信することを行うようにさらに構成され得る。チェックサムは、受信デバイスへの送信のために、いずれかの任意のワード内に挿入され得る。

10

【0011】

別の態様によれば、バスインターフェースと処理回路とを備える受信機デバイスが提供される。バスインターフェースは、その上で通信がマスタデバイスによって管理される共有バスに結合するように働き得る。処理回路は、(a)共有バスを介して複数のシンボルを受信すること、(b)複数のシンボルを桁に変換すること、(c)桁を3進数に結合すること、(d)3進数をビットに変換することであって、複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかが、エラー検出のために使用されること、および/あるいは(e)複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかに基づいて、受信された複数のビット内のエラーを確かめることを行うように構成され得る。クロックは、複数の受信されたシンボルのシンボル間遷移内に埋め込まれ得る。共有バスは2ラインバスであってよく、2ラインバスの両方のラインが、シンボルを転送するために使用される。受信機デバイスは、クロック信号内で誤ったクロックパルス消失と誤った余剰クロックパルスとを検出する、状態機械論理回路を実装することができる。

20

【0012】

様々な特徴、性質、および利点は、同様の参照文字が全体を通じて対応して識別する図面と併せて読まれたとき、以下に記載の詳細な説明から明らかになる場合がある。

30

【図面の簡単な説明】

【0013】

【図1】ベースバンドプロセッサおよびイメージセンサを有し、画像データバスおよびマルチモード制御データバスを実装するデバイスを示すブロック図である。

【図2】どのようにクロックがCCLeモードでシンボル間遷移内に埋め込まれ、それによって、データ送信のためにI2Cバスにおける2本のライン(すなわち、SDAラインおよびSCLライン)の使用を可能にし得るかを示す図である。

【図3】トランスコーディングされたシンボル内にクロック信号を埋め込むために、送信機においてデータビットをトランスコーディングし、トランスコーディングされたシンボルにするための例示的な方法を示すブロック図である。

40

【図4】遷移番号と順次シンボルとの間の例示的な変換を示す図である。

【図5】遷移番号と順次シンボルとの間の変換を示す図である。

【図6】最上位ビットから最下位ビットまで、2進ビットを3進数に変換するための方法を示す図である。

【図7】最上位ビットから最下位ビットまで、2進ビットを3進数に変換するための送信機側論理回路を示す図である。

【図8】最上位ビットから最下位ビットまで、3進数を2進ビットに変換するための方法を示す図である。

50

【図 9】12桁の3進数を20ビットに変換するための受信機側論理回路を示す図である。

【図 10】ビット19(すなわち、ビットカウントが、ビット0である第1のビットにおいて開始するとき、第20のビット)が、CCleプロトコルでは大抵使用されず、共有バス上のデバイス間のコマンドのために使用され得ることを、概念的に示す図である。

【図 11】共有バスがI2CモードからCCleモードへ動作するように切り替え中であることをスレーブデバイスに示すために、共有バスを介してマスタデバイスによって送られ得る、CCleモードエントリインジケータのための例示的な一般呼出し(general call)を示す図である。

【図 12】CCleモードからI2Cモードへの遷移をすべてのCCle可能デバイスに示すために、CCleマスタデバイス(たとえば、I2Cモードである間に、図1におけるマスタデバイス)によって発行され得る、例示的なCCle呼出しを示す図である。

10

【図 13】例示的なCCleスレーブ識別子(SID)ワードフォーマットを示す図である。

【図 14】例示的なCCleアドレスワードフォーマットを示す図である。

【図 15】例示的な書込みデータワードフォーマットを示す図である。

【図 16】例示的な読取り指定ワードフォーマットを示す図である。

【図 17】例示的な読取りデータワードフォーマットを示す図である。

【図 18】I2Cの1バイト書込みデータ動作の例示的なタイミング図である。

【図 19】SDAラインおよびSCLラインを介した送信のために、データビットが12シンボルにトランスコーディングされた、例示的なCCle送信を示す図である。

【図 20】図2～図10に示した符号化方式から得られた第20のビット(ビット19)の例示的なマッピングを示す図である。

20

【図 21】図20の第20のビット(ビット19)領域の例示的なマッピング内の下位領域の詳細を示す図である。

【図 22】発生し得る様々なシンボルエラー状態を示す図である。

【図 23】送信シンボルシーケンス0321\_0321\_0321(2進シーケンス0000\_0000\_0000\_0000\_0000および3進数0000\_0000\_0000<sub>3</sub>に変換する)内の可能性があるエラーを示すテーブル、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す図である。

【図 24】送信シンボルシーケンス2301\_2301\_2301(2進シーケンス0100\_0000\_1101\_1111\_1000および3進数1111\_1111\_1111<sub>3</sub>に変換する)内の可能性があるエラーを示すテーブル、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す図である。

30

【図 25】送信シンボルシーケンス3131\_3131\_3131(2進シーケンス1000\_0001\_1011\_1111\_0000および3進数2222\_2222\_2222<sub>3</sub>に変換する)内の可能性があるエラーを示すテーブル、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す図である。

【図 26】送信シンボルシーケンス0132\_3101\_3231内の可能性があるエラーを示すテーブル、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す図である。

【図 27】送信シンボルシーケンス2030\_2120\_3021内の可能性があるエラーを示すテーブル、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す図である。

40

【図 28】送信シンボルシーケンス3231\_0132\_3101内の可能性があるエラーを示すテーブル、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す図である。

【図 29】CCleワードが、エラー検出のために3桁の最下位ビットまたは最下位ビットをどのように使用することができるかを示す図である。

【図 30】8ビットのチェックサムがCCleワード内でどのように実装され得るかを示す図である。

【図 31】8ビットチェックサム発生器の例示的な実装形態を示す図である。

50

【図32】エラー検出を容易にする例示的なCCleワードフォーマットを示す図である。

【図33】本開示によるマスタ/スレーブデバイスの例示的な構成要素を示すブロック図である。

【図34】プロトコル(たとえば、CCleプロトコル)の1つまたは複数のエラー検出機能に従って通信を容易にする例示的な方法を示す図である。

【図35】クロックエラーなしの例示的なCCleワード送信を示す図である。

【図36】単一クロック消失エラーありの例示的なCCleワード送信を示す図である。

【図37】ダブルクロック消失エラーありの例示的なCCleワード送信を示す図である。

【図38】余剰クロックエラーありの例示的なCCleワード送信を示す図である。

【発明を実施するための形態】

10

【0014】

以下の説明では、実施形態の完全な理解を提供するために、具体的な詳細が与えられる。ただし、実施形態はこれらの具体的な詳細なしに実施され得ることを当業者は理解されよう。たとえば、回路は、不必要な詳細で実施形態を不明瞭にしないために、ブロック図で示される場合がある。他の例では、よく知られている回路、構造、および技法は、実施形態を不明瞭にしないために、詳細に示されない場合がある。

【0015】

《概要》

2進数を3進数に変換し、次に3進数が、以前よりも高い速度を可能にするために、2ラインI2Cバスを介した送信のためにクロックを埋め込まれたシンボルにトランスコーディングされる、CCle(カメラ制御インターフェース拡張)と呼ばれるCCIの拡張が開発されている。2進数は、最初に3進数(すなわち、3進法の数)に変換される。次に、3進数の各桁がシンボルに変換され、そこで、いかなる2つの順次シンボルも繰り返さず、限られた数のシンボル(たとえば、4つのシンボル)のみが使用される。次に、それらのシンボルがバスを介して送信される。

20

【0016】

3進数空間およびシンボルへの変換の使用は、余剰ビットが利用可能になる結果となる。一例では、この余剰ビットは最上位であり、3進数の領域が、さもなければ利用可能ではない他の機能をサポートするために利用可能になる場合がある。たとえば、エラー検出およびチェックサムが、この余剰ビット内に含まれ得る余剰情報のために、容易になり得る。

30

【0017】

一態様によれば、各送信ワード(たとえば、20ビットワード)内の最下位ビットが、送信中に発生し得るエラーのほぼ半数を検出するために使用され得る。

【0018】

第2の態様によれば、送信におけるエラーを検出するために、柔軟なチェックサムワードが通信プロトコルに追加される。チェックサムワードは、送信におけるいずれかの任意の点において挿入され得る。

【0019】

第3の態様によれば、ワード内のビットは、3桁の最下位ビットがワード全体にわたるエラー検出のために利用可能であるように配置される。

40

【0020】

第4の態様によれば、受信機デバイスは、エラーを検出するために、CCleワードのシンボル間遷移内に埋め込まれたクロック信号内の消失したクロックパルスと余剰クロックパルスとを検出する助けとなる、状態機械論理回路を維持することができる。

【0021】

《例示的な動作環境》

図1は、ベースバンドプロセッサ104およびイメージセンサ106を有し、画像データバス116およびマルチモード制御データバス108を実装するデバイス102を示すブロック図である。図1はカメラデバイス内のマルチモード制御データバス108を示すが、この制御データバ

50

ス108が様々な異なるデバイスおよび/またはシステム内に実装され得ることは明らかであろう。画像データは、画像データバス116(たとえば、高速差動DPHYリンク)を介して、イメージセンサ106からベースバンドプロセッサ104に送られる場合がある。

#### 【0022】

一例では、制御データバス108は、2本の線、クロックライン(SCL)およびシリアルデータライン(SDA)を備える、I2Cバスであり得る。クロックラインSCLは、I2Cバス(制御データバス108)を介してすべてのデータ転送を同期させるために使用されるクロックを送るために使用され得る。データラインSDAおよびクロックラインSCLは、I2Cバス(制御データバス108)上ですべてのデバイス112、114、および118に結合される。この例では、制御データが、制御データバス108を介して、ベースバンドプロセッサ104とイメージセンサ106ならびに他の周辺デバイス118、122、および/または124との間で交換され得る。I2Cのための標準クロック(SCL)速度は、最高100KHzである。I2C高速モードにおける標準クロックSCL速度は、最高400KHzであり、I2C高速モードプラス(Fm+)では、最高1MHzである。I2Cバスを介したこれらの動作モードは、カメラの適用例に使用されるとき、カメラ制御インターフェース(CCI)モードと呼ばれる場合がある。

#### 【0023】

一態様によれば、カメラ動作をサポートするために、改善された(すなわち、1MHzよりも大きい制御データバス送信周波数を用いる)動作モードを、マルチモード制御データバス108を介して実装することができる。I2Cバスを介してのこの改善された動作モードは、カメラの適用例のために使用されるとき、カメラ制御インターフェース拡張(CCIE)モードと呼ばれる場合がある。CCIEモードでは、SCLラインおよびSDAラインは両方とも、それらの2本のラインを介したシンボル間遷移内にクロックが埋め込まれる間に、データを送信するために使用され得る。この例では、ベースバンドプロセッサ104はマスタデバイス112を含み、イメージセンサ106はスレーブデバイス114を含み、マスタデバイス112とスレーブデバイス114の両方は、制御データバス108に結合された他のレガシーI2Cデバイスの適切な動作に影響を与えることなく、制御データバス108を介してカメラ制御インターフェース拡張(CCIE)モードに従って動作することができる。一態様によれば、制御データバス108を介したこの改善されたモードは、CCIEデバイスとレガシーI2Cスレーブデバイスとの間にいかなるブリッジデバイスを用いることもなく、実装され得る。

#### 【0024】

I2C互換デバイスとCCIE互換デバイスとが共有制御データバス108に並行して結合されることを可能にするプロトコルが提供される。制御データバス108は、別個の通信プロトコル(たとえば、I2CモードおよびCCIEモード)に従った動作間で動的に切り替えることができる。前記のように、共有制御データバス108への通信および/またはアクセスは、マルチモードマスタデバイス112によって管理される。マスタデバイスは、制御データバス108がその通信プロトコルを第1のプロトコルモード(たとえば、I2Cモード)から第2のプロトコルモード(たとえば、CCIEモード)へ切り替えるべきであることを示すために、エントリ呼出し(entry call)を送信する。同様に、マスタデバイスは、制御データバス108がその通信プロトコルを第2のプロトコルモード(たとえば、CCIEモード)から第1のプロトコルモード(たとえば、I2Cモード)へ切り替えるべきであることを示すために、終了呼出し(exit call)を送信する。共有バス108に結合されたスレーブデバイスは、これらのエントリ呼出しと終了呼出しとを監視して、いつ共有バス108上で動作することができるかを確認する。

#### 【0025】

##### 《例示的なCCIE符号化技法》

図2は、どのようにクロックがCCIEモードでシンボル間遷移内に埋め込まれ、それによって、データ送信のためにI2Cバスにおける2本のライン(すなわち、SDAラインおよびSCLライン)の使用を可能にし得るかを示す。一例では、このクロックの埋め込みは、遷移クロックトランスコーディングによって達成され得る。たとえば、物理リンク(線)を介して送信されるべきデータ204は、送信されるシンボルが、送信シンボル206のシンボルサイク

ルまたは遷移ごとに状態を変化させることを保証されるように、トランスコーディングされる。一例では、ビットのシーケンスが3進数に変換され、3進数の各桁が、送信のためにシンボルに変換される。順次シンボルは、3進数の2つの順次の桁が同じであるときでも、異なるように保証される。したがって、元のクロック202は、シンボルサイクルごとのシンボル状態の変化において埋め込まれ得る。受信機は、(送信シンボル206内の)各シンボルにおける状態遷移からクロック情報208を復元し、次に、送信シンボル206のトランスコーディングを逆転させて、元のデータ210を取得する。一例では、各シンボルが桁に変換され、複数の桁が3進数を構成し、次に、3進数が複数のビットに変換される。したがって、元のクロック202は、シンボルサイクルごとのシンボル状態の変化において埋め込まれ得る。これによって、I2Cバスの両方の線(図1における制御データバス108、SDAラインおよびSCLライン)がデータ情報を送るために使用されることが可能になる。加えて、クロック信号とデータ信号との間のセットアップ時間および保持時間を有する必要がもはやないので、シンボルレートが倍増され得る。

#### 【0026】

図3は、トランスコーディングされたシンボル内にクロック信号を埋め込むために、送信機においてデータビットをトランスコーディングし、トランスコーディングされたシンボルにするための例示的な方法を示すブロック図である。送信機302において、データビット304のシーケンスが3進(3進法の)数(すなわち、「遷移番号」)に変換され、次に、3進数が(順次)シンボルに変換され、それらのシンボルがクロックラインSCL312およびデータラインSDA314を介して送信される。

#### 【0027】

一例では、2進データの元の20ビットが、ビット-遷移番号変換器ブロック308に入力されて、12桁の3進数に変換される。12桁の3進数の各桁は、「遷移番号」を表す。2つの連続する遷移番号は、同じ数であってよい(すなわち、3進数の連続する桁が同じであってよい)。各遷移番号は、2つの連続する順次シンボルが同じ値を有することがないように、遷移-シンボルブロック310において順次シンボルに変換される。順次シンボルごとに遷移が保証されるので、そのような順次シンボル遷移は、クロック信号を埋め込むように働き得る。次に、各順次シンボル316が、2線式物理リンク(たとえば、SCLライン312とSDAライン314とを備えるI2Cバス)を介して送られる。

#### 【0028】

図4は、遷移番号402と順次シンボル404との間の例示的な変換を示す。遷移番号とも呼ばれる、3進数、3進法の数の個々の桁は、3つの可能な数字または状態、0、1または2のうちの1つを有することができる。3進数の2つの連続する桁において同じ数字が現れる場合があるが、2つの連続する順次シンボルは同じ値を有しない。遷移番号と順次シンボルとの間の変換は、連続する遷移番号が同じである場合でも、順次シンボルが(順次シンボルから順次シンボルへと)常に変化することを保証する。

#### 【0029】

この変換関数を図5に例示的に記載する。送信機側(TX:TからSへ)502では、遷移番号(T)が順次シンボル(S)に変換され得る。たとえば、現在の順次シンボル( $C_s$ )は、前の順次シンボル( $P_s$ )と、現在の遷移番号(T)の関数である一時遷移番号( $T_{tmp}$ )とに基づいて、取得され得る。一時遷移番号( $T_{tmp}$ )は、現在の遷移番号Tを0と比較することによって取得することができ、 $T=0$ のとき、一時遷移番号( $T_{tmp}$ )は3に等しくなり、そうでない場合( $T$ が0に等しくないとき)、 $T_{tmp}$ はTに等しくなる(すなわち、 $T_{tmp}=T \div 3$ )。現在の順次シンボルは、現在の順次シンボル( $C_s$ )+前の順次シンボル( $P_s$ )+一時遷移番号( $T_{tmp}$ )の和として取得され得る(すなわち、 $C_s=P_s+T_{tmp}$ )。

#### 【0030】

受信機側(RX:SからTへ)504では、変換演算が逆にされて、現在の順次シンボル( $C_s$ )および前の順次シンボル( $P_s$ )から遷移番号が取得される。一時遷移番号( $T_{tmp}$ )は、現在の順次シンボル( $C_s$ )+4の和-前の順次シンボル( $P_s$ )として取得され得る(すなわち、 $T_{tmp}=C_s+4-P_s$ )。現在の遷移番号(T)は一時遷移番号( $T_{tmp}$ )に等しいが、一時遷移番号( $T_{tmp}$ )が3と比較

10

20

30

40

50

され、 $T_{tmp}=3$ のとき、一時遷移番号( $T_{tmp}$ )は0に等しくなり、そうでない場合( $T_{tmp}$ が3に等しくないとき)、 $T$ は $T_{tmp}$ に等しくなる(すなわち、 $T=T_{tmp}=3 \neq 0:T$ )。

【0031】

テーブル506は、遷移番号と順次シンボルとの間の変換を示す。

【0032】

再び図4を参照すると、遷移番号と順次シンボルとの間の変換の一例がその中に示されている。たとえば、第1のサイクル406では、現在の遷移番号( $T_a$ )は2であり、そのため $T_{tmp}$ もまた2であり、前の順次シンボル $P_s$ は1であり、新しい現在の順次シンボル $C_s$ は現在3である。

【0033】

第2のサイクル408では、遷移番号( $T_b$ )は1である。遷移番号( $T_b$ )は0に等しくないので、一時遷移番号 $T_{tmp}$ は、1である遷移番号( $T_b$ )値に等しい。現在の順次シンボル( $C_s$ )は、3である前の順次シンボル( $P_s$ )値を、1である一時遷移番号 $T_{tmp}$ に加算することによって、取得される。加算演算の結果が、3よりも大きい4に等しいので、ロールオーバーされた数0が現在の順次シンボル( $C_s$ )になる。

【0034】

第3のサイクル410では、現在の遷移番号( $T$ )は1である。遷移番号 $T$ が1であるので、一時遷移番号 $T_{tmp}$ もまた1である。現在の順次シンボル( $C_s$ )は、0である前の順次シンボル( $P_s$ )値を、1である一時遷移番号 $T_{tmp}$ に加算することによって、取得される。加算演算の結果が、3よりも大きくない1に等しいので、現在の順次シンボル( $C_s$ )は1に等しい。

【0035】

第4のサイクル412では、現在の遷移番号( $T$ )は0である。遷移番号 $T$ が0であるので、一時遷移番号 $T_{tmp}$ は3である。

【0036】

現在の順次シンボル( $C_s$ )は、1である前の順次シンボル( $P_s$ )値を、3である一時遷移番号 $T_{tmp}$ に加算することによって、取得される。加算演算の結果が、3よりも大きい4であるので、ロールオーバーされた数0が現在の順次シンボル( $C_s$ )になる。

【0037】

2つの連続する3進数の桁 $T_b$ および $T_c$ が同じ数を有する場合でも、この変換は、2つの連続する順次シンボルが異なる状態値を有することを保証することに留意されたい。このため、順次シンボル404内の保証された遷移がクロック信号を埋め込むように働き、それによって、データ送信のためにI2Cバス内のクロックラインSCLを解放することができる。

【0038】

図3を再び参照すると、受信機320において、このプロセスが逆にされて、トランスコーディングされたシンボルがビットに戻るように変換され、このプロセスにおいて、クロック信号がシンボル遷移から抽出される。受信機320は、2線式物理リンク(たとえば、SCLライン324およびSDAライン326を備えるI2Cバス)を介して、順次シンボル322のシーケンスを受信する。受信された順次シンボル322は、クロック-データ復元(CDR)ブロック328に入力されて、クロックタイミングが復元され、トランスコーディングされたシンボル( $S$ )がサンプリングされる。次に、シンボル-遷移番号変換器ブロック330が、トランスコーディングされた(順次)シンボルを、遷移番号、すなわち、1つの3進数字に変換する。次に、遷移番号-ビット変換器332は、12個の遷移番号を変換して、12桁の3進数から20ビットの元のデータを回復させる。

【0039】

2線式バスおよび12個の遷移番号について図3および図4に示した例は、 $n$ 線式システムおよび $m$ 個の遷移番号に一般化され得る。 $T_0 \sim T_{m-1}$ の1つの $T$ あたりに $r$ 個の可能なシンボル遷移状態がある場合、 $m$ 個の遷移が $r^m$ 個の異なる状態を送ることができ、すなわち、 $r=2^n-1$ 個である。したがって、遷移 $T_0 \dots T_{m-1}$ は、 $(2^n-1)^m$ 個の異なる状態を有することができるデータを含む。

【0040】

10

20

30

40

50

本明細書で説明する本技法を使用して、I2C標準バスが提供し、本明細書でCCleモードと呼ばれるものを超えて、制御バス108(図1)のリンクレートを高めることができる。一例では、制御データバス108に結合されるマスタデバイスおよび/またはスレーブデバイスは、同じ制御データバスを介して、標準I2Cバスを使用して可能であるよりも高いビットレートを達成するために、(図2、図3、図4、および図5に示すような)シンボル送信内にクロック信号を埋め込む送信機および/または受信機を実装することができる。

【0041】

図6は、最上位ビットから最下位ビットまで、2進ビットを3進数に変換するための方法を示す。3進数の各桁は、受信デバイスに送信されるシンボルにトランスコーディング(変換)され得る。3進数を表す $T_0, T_1 \dots T_{11}$ を有する12桁の3進数602では、 $T_0$ は $3^0$ の桁を表し(かつ、最下位桁であり)、 $T_{11}$ は $3^{11}$ の桁を表す(かつ、最上位桁である)。受信ビット(たとえば、20ビットのシーケンス)で開始すると、3進数602の最上位桁 $T_{11}$ が最初取得される。次に、次の最上位桁 $T_{10}$ が次に取得される。このプロセスは、最下位桁 $T_0$ が取得されるまで続く。3進数602の桁の各々はまた、「遷移番号」と呼ばれることもある。

【0042】

図7は、最上位ビットから最下位ビットまで、2進ビットを3進数に変換するための送信機側論理回路を示す。図6および図7は、 $T_{11}, T_{10}, T_9, \dots, T_0$ の順序で送られる12桁の3進数602を示す。最上位ビットを最初取得し、送ることによって、関連する論理および回路の複雑さが簡略化される。図6および図7における手法では、最上位の順次シンボルが受信デバイスに最初に送信され、したがって、MSS最優先(MSS first:most significant symbol first)と呼ばれる。本明細書で使用する「最下位シンボル」は、3進数602の最下位桁に対応するトランスコーディングされたシンボルを指す。たとえば、ならびに図4および図5の説明を参照すると、 $T_0$ が順次シンボルにトランスコーディングされるとき、それは最下位の3進数の桁から発生したので、最下位シンボルである。同様に、本明細書で使用する「最上位シンボル」は、3進数602の最上位桁に対応するトランスコーディングされたシンボルを指す。たとえば、ならびに図4および図5の説明を参照すると、 $T_{11}$ が順次シンボルにトランスコーディングされるとき、それは最上位の3進数の桁から発生したので、最上位シンボルである。また、シンボル-遷移番号変換器ブロック330(図3)が、後で、トランスコーディングされた(順次)シンボルを受信し、遷移番号、すなわち、3進数の桁に変換するとき、それは、最上位桁 $T_{11}$ が最初に、最下位桁 $T_0$ が最後になる。

【0043】

図3に戻って参照すると、20ビットの元のデータが、逆順で3進数に変換され(すなわち、最上位ビットが変換器に最初に供給され)、次に、3進数の各桁(たとえば、各遷移番号)が、逆順で順次シンボルに変換(すなわち、トランスコーディング)され、これらのトランスコーディングされたシンボルが、バス上で逆順に(すなわち、最上位シンボルが最初に)送信される。

【0044】

図8は、最上位ビットから最下位ビットまで、3進数を2進ビットに変換するための方法を示す。すなわち、この受信機側変換は、図6および図7に示した送信機側変換において実施された動作を逆にする。受信デバイス(たとえば、スレーブデバイス)は、逆順送信を受信し、クロック復元およびシンボルサンプリングを実施して、トランスコーディングされたシンボルを3進数に戻すように変換し、次に、3進数は逆順で図9における論理回路に供給され、論理回路は3進数を20ビットの2進の元データに戻すように変換する。図7は、論理デバイスへの単出力に結合された12個の入力を有するマルチプレクサを示す。

【0045】

図9は、12桁の3進数を20ビットに変換するための受信機側論理回路を示す。

【0046】

図10は、ビット19(すなわち、ビットカウントが、ビット0である第1のビットにおいて開始するとき、第20のビット)が、CCleプロトコルでは大抵使用されず、共有バス上のデバイス間のコマンドのために使用され得ることを、概念的に示す。すなわち、図3~図9に

10

20

30

40

50

示した符号化方式の結果として、余分のビット(すなわち、ビット19)が現在、送信シンボル内で利用可能である。より具体的には、図10は、ビット19(すなわち、第20のビット)を示す。言い換えれば、コンピュータサイエンスにおいて典型的であるように、ビットワイズのカウンタは0で開始し、ビット19は第20のビットである。ここで、ビット0~18は、00\_0000\_0000<sub>3</sub>から2221\_2201\_2001<sub>3</sub>の3進数範囲内で表される。2221\_2201\_2002<sub>3</sub>から2222\_2222\_2222<sub>3</sub>の範囲内の3進数は使用されない。したがって、3進数範囲2221\_2201\_2002<sub>3</sub>から2222\_2222\_2222<sub>3</sub>を使用して、ビット19(すなわち、第20のビット)を表すことができる。言い換えれば、2221\_2201\_2002<sub>3</sub> 3進数は、10,000,000,000,000,000,000 2進数(0x8000 0 16進数)であり、2222\_2222\_2222<sub>3</sub> 3進数(0x81BF0)は、可能な最大の12桁の3進数である。

10

#### 【0047】

《CCleモードのための例示的なプロトコル》

図11は、共有バスがI2CモードからCCleモードへ動作するように切り替え中であることをスレーブデバイスに示すために、共有バスを介してマスタデバイスによって送られ得る、CCleモードエントリーインジケータのための例示的な一般呼出しを示す。一般呼出し1102は、I2CモードからCCleモードへの遷移をすべてのI2C互換デバイスに示すために、共有バスを介してI2Cマスタデバイス(たとえば、SDAラインおよびSCLラインを介して、I2Cモードである間、図1におけるマスタデバイス112)によって発行され得る。

#### 【0048】

I2Cモードでは、CCleマスタデバイスは、このI2C一般呼出し1102を、「CCleモード」バイトまたはインジケータ1104とともに発行する。CCle互換スレーブデバイスは、一般呼出し1102の受信を肯定応答する。CCle互換スレーブデバイスは、必要な場合、一般呼出し中に(制御データバス108の)SCLラインを低に保持することによって、待機サイクルを挿入することができる。

20

#### 【0049】

CCleモードになると、すべてのCCle互換デバイスは、CCleマスタデバイスからの要求に応答することができる。CCleモードをサポートしない共有制御データバス上のレガシーI2C互換スレーブデバイスの動作状態または任意の機能は、いかなるCCleトランザクションによっても影響を受けない。

#### 【0050】

図12は、CCleモードからI2Cモードへの遷移をすべてのCCle可能デバイスに示すために、CCleマスタデバイス(たとえば、I2Cモードである間に、図1におけるマスタ112)によって発行され得る、例示的なCCle呼出し1202を示す。CCleマスタデバイスは、CCle SIDの代わりにこの終了呼出し1202を発行することができる。

30

#### 【0051】

CCleモードで、CCleモードにおける最後のデータおよび後続するSの後、CCleマスタは、CCleモードの終了とI2Cモードへ戻る遷移とを(たとえば、CCle互換デバイスに)示すために、特殊なCCle SIDコードである「終了」コード/インジケータ1204を送る。加えて、「終了」コード/インジケータ1204の後、CCleマスタデバイスは、S(開始ビット)および後続する「一般呼出し」1206を、I2Cプロトコルに従って、I2Cプロトコル内の第2のバイトにおける「終了」コード1208とともに送る。すべてのCCle対応スレーブは、一般呼出し1204に対して肯定応答しなければならない。

40

#### 【0052】

図13は、例示的なCCleスレーブ識別子(SID)ワードフォーマットを示す。これは、CCle SIDワードフォーマット1302の一部としての16ビットのスレーブ識別子(SID)1304の使用を示す。そのようなSIDワードフォーマットは、そのワードが制御データバス上に配置されるとき、特定のスレーブデバイスを識別するために使用されることになる。

#### 【0053】

図14は、例示的なCCleアドレスワードフォーマット1402を示す。これは、各アドレスワード1406が16ビットのアドレス1404を含むことを示す。アドレスワード1406はまた、2ビ

50



ットの制御コード1408と、1ビットのエラー検出定数1410とを含む。テーブル1412は、制御コードのための様々な可能な値を示す。

【0054】

複数のアドレスワードが順次送られ得る。現在の制御ワードが「00」である場合、これは、アドレスワードが後続することになることを意味する。制御コードが「01」である場合、次のデータワードは、書込みデータワードである。制御コードが「10」である場合、次のデータワードは、1ワード読取りデータワードである。制御コード「11」は禁止される。

【0055】

図15は、例示的な書込みデータワードフォーマット1500を示す。これは、各データワード1500が16ビットの書込みデータ部分1502を含むことを示す。書込みデータワード1500はまた、2ビットの制御コード1504と、1ビットのエラー検出定数1510とを含む。テーブル1514は、制御コードのための様々な可能な値を示す。

10

【0056】

複数の書込みデータワードが順次送られ得る。現在の書込みワードの制御コードが「00」(シンボルC0)である場合、データは、前のアドレスに書き込まれるべきである。現在の書込みワードの制御コードが「01」(シンボルC1)である場合、データは、前のアドレス+1に書き込まれるべきである。制御コードが「10」(シンボルE)である場合、次のワードはSIDまたは終了コードになる。

【0057】

20

図16は、例示的な読取り指定ワードフォーマット1600を示す。読取り指定データワード1600は、16ビットの読取りデータ値部分1604と、2ビットの制御コード1608と、3ビットのエラー検出定数1610とを含み得る。

【0058】

最後のアドレスワード1607の後、「読取り指定」(RS)ワード1612が後続する。読取り指定(RS)ワード1612は、後続する読取りデータワードの数を指定する。テーブル1616に示すように、制御コード「00」は、同じアドレスからの読取りワードを示すために使用される。制御コード「01」は、増分アドレスからの読取りワードを示すために使用される。(そこからデータが読み取られている)スレーブデバイスは、「読取り指定」(RS)ワード1604によって指定されるよりも多くのデータワード(CHKワードを含まない)を送らないものとする。スレーブデバイスは、少なくとも1つの読取りワード(CHKワードを含まない)を送るものとする。スレーブデバイスは、「読取り指定」(RS)ワード1604によって指定された数のワードを送る前に、読取り転送を終了することができる。

30

【0059】

図17は、例示的な読取りデータワードフォーマット1702を示す。読取りデータワード1702は、16ビットの読取りデータ値部分1704と、2ビットの制御コード1706と、1ビットのエラー検出定数1708とを含み得る。SID1707によってアドレス指定されたスレーブデバイスは、要求するマスタデバイスに戻すためのワードの数を決定する。テーブル1716に示すように、読取りワードが同じアドレスから続く場合、制御コードは「00」(シンボルR0)である。読取りワードが増分アドレスから続く場合、制御コードは「01」(シンボルR1)である。ワードが最後の読取りワードであり、その後、CHKがない場合、制御コードは「10」(シンボルE)である。制御コード「00」は禁止される。

40

【0060】

《例示的な共有バスを介したI2C送信対CC1e送信》

図18は、I2Cの1バイト書込みデータ動作の例示的なタイミング図を示す。この例では、共有制御データバス108(図1)は、シリアルデータラインSDA1802とシリアルクロックラインSCL1804とを含む。図18に示す送信方式は、「I2Cモード」と呼ばれることがある。SCLライン1804は、マスタデバイスからすべてのスレーブデバイスへクロックを送るために使用されるが、SDAライン1802は、データビットを送信する。I2Cマスタデバイスは、I2Cバス上のどのスレーブデバイスにマスタデバイスがアクセスすることを望むかを示すために

50

、SDAライン1802において7ビットのスレーブID1808を送り、次に、書込み動作を示すために1ビットを送る。そのIDが7ビットのスレーブID1808にマッチするスレーブデバイスのみが、意図されたアクションを引き起こすことができる。I2Cスレーブデバイスがそれ自体のIDを検出するために、マスタデバイスは、SDAライン上で少なくとも8ビット(または、SCLライン1804上で8個のクロックパルス)を送らなければならない。

【0061】

I2C規格は、すべてのI2C互換スレーブデバイスが、(たとえば、SCLラインが高である間にSDAライン上の高から低への遷移によって示される)START条件1806を受信すると、それらのバス論理をリセットすることを必要とする。

【0062】

CCleプロトコルは、クロック信号をデータ送信内に埋め込みながら、SDAライン1802とSCLライン1804の両方をデータ送信のために使用する。たとえば、データビットは、次にラインを介して送信される複数のシンボルにトランスコーディングされ得る。クロック信号(図18におけるI2CバスではSCLライン)をシンボル送信内に埋め込むことによって、SDAライン1802とSCLライン1804の両方が、データ送信のために使用され得る。

【0063】

図19は、SDAライン1902およびSCLライン1904を介した送信のために、データビットが12シンボルにトランスコーディングされた、例示的なCCle送信を示す。図19に示す送信方式は、「CCleモード」と呼ばれることがある。CCleモードは、ソース同期であり、プッシュプルドライバによって駆動される。共有制御データバスを介してデータを送出するものはどれもまた、データ内(たとえば、シンボル間遷移内)に埋め込まれたクロック情報を送出する。したがって、制御データバス上のただ1つのデバイスが一度に共有制御データバスを駆動することが可能にされる。

【0064】

同じバスを介してレガシーI2CデバイスとCCleデバイスの両方をサポートするために、CCleモード動作は、同じSTART条件1906、1908、1910を使用し、これによって、レガシーI2CスレーブデバイスがいかなるCCle動作に反応することも防止される(たとえば、CCleモード中の開始条件は、レガシーI2Cスレーブデバイスをリセットさせる)。この例では、START条件1906、1908、1910(すなわち、SCLライン1904が高である間に、SDAライン1902上の高から低への遷移によって示される)は、完全なスレーブID(すなわち、完全な7ビット)が送信される前に検出され、したがって、これは、不完全なスレーブID(7ビット未満)である。マスタデバイスが6個のSCLパルスを送り、次に、START条件1906、1908、1910を発行する場合、すべてのレガシーI2Cスレーブデバイスは、データをI2CスレーブIDとして認識する前に、それらのバス論理をリセットする。6ビットシーケンス(たとえば、2つおきのシンボルに対応する)が2つのSTART条件1906、1908、1910の間で送られるので、これらの6ビットシーケンスは、いかなるI2Cスレーブデバイスによっても有効なスレーブIDとして復号されない。したがって、レガシーI2Cスレーブデバイスは、不完全なスレーブIDに作用しないようになる。

【0065】

このシステムでは、マスタデバイスは、バスへのアクセスを制御する。そのため、制御データバスを介して送信することを望むいかなるデバイスも、たとえば、割込み要求を発行することによって、マスタデバイスにそのようなアクセスを要求しなければならない。割込みを発行するための従来技術の機構は、専用の割込みラインまたは専用の割込みバスに依拠していた。しかしながら、そのような専用の割込みラインまたはバスは、そのような割込みラインまたはバスを収容するために、デバイスが少なくとも1つの追加のピンを含まなければならないことを意味する。そのような専用の割込みピンおよびライン/バスの必要性をなくすために、CCle内の帯域内割込みのための機構が必要とされる。

【0066】

帯域内割込みの使用はまた、バス競合または衝突を回避すべきである。たとえば、衝突を回避するために、スレーブデバイスは、マスタデバイスが制御データバスを駆動中で

10

20

30

40

50

ある間に、IRQをアサートするために制御データバス(たとえば、SDAライン1902またはSCLライン1904のいずれか)を駆動することが可能にされるべきではない。

【0067】

《例示的なビット19領域およびチェックサム》

図20は、図2～図10に示した符号化方式から得られた第20のビット(ビット19)の例示的なマッピングを示す。諒解され得るように、利用可能な3進数は、マスタデバイスとスレーブデバイスとの間の特徴および能力を拡張するように働き得る。たとえば、ビット19内で利用可能なこの3進数空間(すなわち、そのビット19が「1」であるデータ領域)は、(a)スレーブ間送信、(b)送信のためのチェックサム、(c)スレーブデバイスへのマスタ動作ハンドオーバー、(d)ハートビートクロックなどを容易にするか、または示すように働き得る。

10

【0068】

図21は、図20の第20のビット(ビット19)領域の例示的なマッピング内の下位領域の詳細を示す。

【0069】

図22は、発生し得る様々なシンボルエラー状態を示す。タイミング図2202は、制御データバス(SDAラインおよびSCLライン)を介した正しい送信と、受信機クロック(RXCLK)とを示す。

【0070】

受信機クロック(RXCLK)が2つのサイクル2212および2214を消失し、データビット2210が誤って検出されるようになる、クロック消失2204が示されている。同じ転送方向により多くの後続のワードがある場合、ワードデータエラーは、後続するワードにおいて検出される可能性が最も高い。同期(SYNC)消失もまた検出され得る。エラーが最後のワードにおいて発生する場合、マスタデバイスは、タイムアウト検出機能を必要とする。

20

【0071】

受信機クロック(RXCLK)が、余剰クロックサイクル2220において検出された余剰シンボル「01」2216および2218を有する、余剰クロック2206が示されている。このエラーは、ワードまたは後続するワードにおいて検出される可能性が最も高い。同期消失もまた検出され得る。

【0072】

受信機クロック(RXCLK)消失がないが、単一シンボルエラー2222が発生する、シンボルエラー2208が示されている。このエラーは、ワードまたは後続するワードにおいて検出される可能性が最も高い。チェックサムエラーが検出される可能性が最も高い。

30

【0073】

《例示的な送信シンボル内のエラー検出》

図23～図28は、様々なCCleワードについて発生し得る様々なシンボルエラー状態(すなわち、シンボルスリップなしの単一シンボルエラー)を示す。図示のように、これらのエラーは、以下でさらに説明するように、3ビット(ビット0、1、および2)を使用することによって検出され得る。これらの例は、エラー検出のために3桁の最下位ビット(ビット[2:0])を使用する。

40

【0074】

図23は、送信シンボルシーケンス0321\_0321\_0321内の可能性があるエラーを示すテーブル2300、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す。図3、図4、図5、図6、図7および図8に示す方法を使用することによって、(ビット[19:0])0000\_0000\_0000\_0000\_0000の20ビットシーケンス2302が、3進数(T11...T0)0000\_0000\_0000<sub>3</sub> 2304に変換され、次にこの3進数が、順次シンボル(S11...S0)0321\_0321\_0321 2306に変換される。この例では、3桁の最下位ビット2308がすべて0(000)である。元の順次シンボル0321\_0321\_0321 2306のシンボルのいずれかにおいて、送信中にエラーが導入される場合、これらは誤ったシンボル2310を生じる。たとえば、最後のシンボル「1」が「0」に変更される場合、これは、3桁の最下位ビットの「000」から「010」への変化

50

を生じる。最後のシンボル「1」が「3」に変更される場合、これは、3桁の最下位ビットの「000」から「001」への変化を生じる。「0」である最初のシンボルが「2」に変更される場合、これは、3桁の最下位ビットの「000」から「100」への変化を生じる。テーブル2300は、3桁の最下位ビットが既知の定数(たとえば、「000」という固定定数)である限り、任意の単一シンボルの変化が3桁の最下位ビットによってどのように検出可能であるかの様々な他の例を示す。

#### 【0075】

図24は、送信シンボルシーケンス2301\_2301\_2301内の可能性があるエラーを示すテーブル2400、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す。図3、図4、図5、図6、図7および図8に示す方法を使用することによって、(ビット[19:0])0100\_0000\_1101\_1111\_1000の20ビットシーケンス2402が、3進数(T11...T0)1111\_1111\_1111<sub>3</sub> 2404に変換され、次にこの3進数が、順次シンボル(S11...S0)2301\_2301\_2301 2406に変換される。この例では、3桁の最下位ビット2408がすべて0(000)である。元の順次シンボル2301\_2301\_2301 2406のシンボルのいずれかにおいて、送信中にエラーが導入される場合、これらは誤ったシンボル2410を生じる。たとえば、最後のシンボル「1」が「3」に変更される場合、これは、3桁の最下位ビットの「000」から「111」への変化を生じる。最後のシンボル「1」が「2」に変更される場合、これは、3桁の最下位ビットの「000」から「001」への変化を生じる。「2」である最初のシンボルが「0」に変更される場合、これは、3桁の最下位ビットの「000」から「100」への変化を生じる。テーブル2400は、3桁の最下位ビットが既知の定数(たとえば、「000」という固定定数)である限り、任意の単一シンボルの変化が3桁の最下位ビットによってどのように検出可能であるかの様々な他の例を示す。

#### 【0076】

図25は、送信シンボルシーケンス3131\_3131\_3131内の可能性があるエラーを示すテーブル2500、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す。図3、図4、図5、図6、図7および図8に示す方法を使用することによって、(ビット[19:0])1000\_0001\_1011\_1111\_0000の20ビットシーケンス2502が、3進数(T11...T0)2222\_2222\_2222<sub>3</sub> 2504に変換され、次にこの3進数が、順次シンボル(S11...S0)3131\_3131\_3131 2506に変換される。この例では、3桁の最下位ビット2508がすべて0(000)である。元の順次シンボル3131\_3131\_3131 2506のシンボルのいずれかにおいて、送信中にエラーが導入される場合、これらは誤ったシンボル2510を生じる。たとえば、最後のシンボル「1」が「0」に変更される場合、これは、3桁の最下位ビットの「000」から「111」への変化を生じる。最後のシンボル「1」が「2」に変更される場合、これは、3桁の最下位ビットの「000」から「110」への変化を生じる。「3」である最初のシンボルが「0」に変更される場合、これは、3桁の最下位ビットの「000」から「001」への変化を生じる。テーブル2500は、3桁の最下位ビットが既知の定数(たとえば、「000」という固定定数)である限り、任意の単一シンボルの変化が3桁の最下位ビットによってどのように検出可能であるかの様々な他の例を示す。

#### 【0077】

図26は、送信シンボルシーケンス0132\_3101\_3231内の可能性があるエラーを示すテーブル2600、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す。図3、図4、図5、図6、図7および図8に示す方法を使用することによって、(ビット[19:0])0001\_1000\_1111\_0011\_1000の20ビットシーケンス2602が、3進数(T11...T0)0120\_1201\_2012<sub>3</sub> 2604に変換され、次にこの3進数が、順次シンボル(S11...S0)0132\_3101\_3231 2606に変換される。この例では、3桁の最下位ビット2608がすべて0(000)である。元の順次シンボル0132\_3101\_3231 2606のシンボルのいずれかにおいて、送信中にエラーが導入される場合、これらは誤ったシンボル2610を生じる。たとえば、最後のシンボル「1」が「0」に変更される場合、これは、3桁の最下位ビットの「000」から「111」への変化を生じる。最後のシンボル「1」が「2」に変更される場合、これは、3桁の最下位ビットの「000」から「110」への変化を生じる。「0」である最初のシンボルが「3」に変更され

る場合、これは、3桁の最下位ビットの「000」から「111」への変化を生じる。テーブル2600は、3桁の最下位ビットが既知の定数(たとえば、「000」という固定定数)である限り、任意の単一シンボルの変化が3桁の最下位ビットによってどのように検出可能であるかの様々な他の例を示す。

#### 【0078】

図27は、送信シンボルシーケンス2030\_2120\_3021内の可能性があるエラーを示すテーブル2700、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す。図3、図4、図5、図6、図7および図8に示す方法を使用することによって、(ビット[19:0])0100\_1010\_1101\_1010\_1000の20ビットシーケンス2702が、3進数(T11...T0)1201\_2012\_0120<sub>3</sub> 2704に変換され、次にこの3進数が、順次シンボル(S11...S0)2030\_2120\_3021 2706に変換される。この例では、3桁の最下位ビット2708がすべて0(000)である。元の順次シンボル2030\_2120\_3021 2706のシンボルのいずれかにおいて、送信中にエラーが導入される場合、これらは誤ったシンボル2710を生じる。たとえば、最後のシンボル「1」が「0」に変更される場合、これは、3桁の最下位ビットの「000」から「010」への変化を生じる。「2」である最初のシンボルが「0」に変更される場合、これは、3桁の最下位ビットの「000」から「011」への変化を生じる。テーブル2700は、3桁の最下位ビットが既知の定数(たとえば、「000」という固定定数)である限り、任意の単一シンボルの変化が3桁の最下位ビットによってどのように検出可能であるかの様々な他の例を示す。

#### 【0079】

図28は、送信シンボルシーケンス3231\_0132\_3101内の可能性があるエラーを示すテーブル2800、および、そのようなエラーが3桁の最下位ビット内でどのように検出可能であることを示す。図3、図4、図5、図6、図7および図8に示す方法を使用することによって、(ビット[19:0])0101\_1110\_1101\_0000\_1000の20ビットシーケンス2802が、3進数(T11...T0)2012\_0120\_1201<sub>3</sub> 2804に変換され、次にこの3進数が、順次シンボル(S11...S0)3231\_0132\_3101 2806に変換される。この例では、3桁の最下位ビット2808がすべて0(000)である。元の順次シンボル3231\_0132\_3101 2806のシンボルのいずれかにおいて、送信中にエラーが導入される場合、これらは誤ったシンボル2810を生じる。たとえば、最後のシンボル「1」が「3」に変更される場合、これは、3桁の最下位ビットの「000」から「111」への変化を生じる。「3」である最初のシンボルが「0」に変更される場合、これは、3桁の最下位ビットの「000」から「100」への変化を生じる。テーブル2800は、3桁の最下位ビットが既知の定数(たとえば、「000」という固定定数)である限り、任意の単一シンボルの変化が3桁の最下位ビットによってどのように検出可能であるかの様々な他の例を示す。

#### 【0080】

##### 《第1の例示的なエラー検出手法》

図23～図28において言及したように、20ビットのCCleワードの3桁の最下位ビットを使用して、クロック消失またはシンボルスリップなしのいずれかの単一シンボルエラーを検出することが可能である。

#### 【0081】

図29は、CCleワード2902が、エラー検出のために3桁の最下位ビット2903または最下位ビット2905をどのように使用することができるかを示す。第1のCCleワード2902(たとえば、20ビットシーケンス)では、3桁の最下位ビット2903は、固定定数(たとえば、「000」)に設定されてよく、受信された20ビットシーケンスが、固定定数(たとえば、「000」)以外の何かである3桁の最下位ビットを有する場合、エラーが受信デバイスにおいて検出される。CCleワード内(たとえば、ワード内の12シンボルのいずれか内、または20ビット(16進の0x00000から0x81BF0までのビット[19:0])のいずれか内)のシンボルスリップなしのどの単一シンボルエラーも、この手法を使用することによって逃されず、それによって、いかなるそのような単一シンボルエラーも100%の検出が提供される。しかしながら、エラー検出のために20ビットの中から3ビットを使用することは、多くの実装形態にとってコストがかかりすぎる場合がある。

#### 【0082】

10

20

30

40

50

代替手法は、エラー検出のために3桁の最下位ビットの代わりに単一ビット(たとえば、最下位ビット0)を使用することを提供する。固定ビット0(たとえば、「0」に固定される)が、ほぼ半数の場合において、(クロック消失またはシンボルスリップなしの)シンボルエラーの検出を助けるために使用され得る。固定ビット0を使用することに加えて、順次CCleワードのビット1および2のチェックサムが計算され(図31参照)、受信デバイスに送られて、送信されたCCleワード内でエラーが発生したか否かが確かめられ得る。

#### 【0083】

第2のCCleワード2904(たとえば、20ビットシーケンス)では、最下位ビット2905が固定定数(たとえば、「0」)に設定され得る。図23~図28から諒解され得るように、最下位ビット2905は、対応するシンボルのシーケンス内にシンボルエラーがあるとき、時間の約半分で変化する。したがって、最下位ビットにおいて反映されたエラーについて、受信デバイスは、シーケンス内の最下位ビットが固定定数(たとえば、「0」)以外の何かである場合、受信された20ビットシーケンス内に単一シンボルエラー(クロック消失またはシンボルスリップなし)があることを成功裏に確かめることができる。

#### 【0084】

エラー検出のために最下位ビット2905を使用することに加えて、またはそれとともに、チェックサムメッセージが、CCleワードのシーケンスを介した高度のエラー検出を提供するために使用され得る。そのようなチェックサムメッセージがエラーを検出しない唯一のシナリオは、チェックサムを計算するときに2つのシンボルエラーの影響が打ち消しになる場合である。

#### 【0085】

図30は、8ビットのチェックサムがCCleワード内でどのように実装され得るかを示す。この例では、CCleアドレスワード3004a、書込みワード3004b、および/または読取りワード3004cについて、CCleフレーム内にチェックサム値を配置しながら、ビット19を「1」に、およびビット12を「0」に設定することによって、チェックサムワード3005が(20ビットシーケンス内で)示され得る(たとえば、各ワードのビット4~11がチェックサムのために使用され得る)。8ビットのチェックサム値3003は、いくつかのCCleワード送信を介して計算され得る。たとえば、このチェックサムワード3005は、CCleワード送信のシーケンスの終了において送られ得る。

#### 【0086】

このチェックサムワードは、ビット19のために利用可能な追加の空間内で定義され得る(図20参照)。チェックサムワード3005は、書込みコマンド3006および/または読取りコマンド3010のアドレスまたはデータ部分内に挿入され得る。たとえば、書込みコマンド3006において、1つもしくは複数のアドレスワード3008a、3008b、ならびに/または、1つもしくは複数の書込みワード3008cおよび3008dが、チェックサム値を(たとえば、マスタデバイスからスレーブデバイスへ)搬送するために使用され得る。同様に、読取りコマンド3010において、1つもしくは複数のアドレスワード3012a、3012b、ならびに/または、1つもしくは複数の読取りワード3012cおよび3012dが、チェックサム値を搬送するために使用され得る。

#### 【0087】

他のチェックサム手法とは対照的に、本CCleチェックサムワード3005は、送信側によって送信内のどこでもいつでも挿入され得る。

#### 【0088】

図31は、8ビットチェックサム発生器の例示的な実装形態を示す。チェックサム発生器3102は、チェックサム3104の8ビットの各々について、1ビットレジスタを含み得る。第1のCCleワードN3110のビット2および1と、チェックサム3104内の最下位ビットとの排他的論理和(XOR)がとられる。次に、第2のCCleワードN+1 3112のビット2および1と、チェックサム3104内の次の最下位ビットとの排他的論理和(XOR)がとられる。次に、第3のCCleワードN+2 3114のビット2および1と、チェックサム3104内の次の最下位ビットとの排他的論理和(XOR)がとられる。次に、第4のCCleワードN+3 3116のビット2および1と、チェックサム31

10

20

30

40

50

04内の最上位ビットとの排他的論理和(XOR)がとられる。チェックサム値3104の最上位ビットに到達した後、次のCCleワードN+4において、チェックサム発生器は、チェックサム3104の2桁の最下位ビットにおいて再度開始することができる。

【0089】

このようにして、異なるCCleワードからの寄与が、8ビットのチェックサムにわたるこれらの使用において伝播する。1ビットレジスタが、転送の開始ごと(たとえば、SIDの前、および/または読取りワードの開始前)に初期化され得る。CCleチェックサムワードを実装するために、送信デバイスは、各CCleワードが送られるときにチェックサム値を計算することができる。受信機デバイスでは、チェックサム値がまた、受信されたCCleワードごとに同様に計算される。次に、送信機デバイスは、受信機デバイスがそれ自体のチェックサム値と比較して、いずれかのシンボルエラーが発生したか否かを確かめることができる、直近のチェックサム値とともに、CCleチェックサムワードを受信機デバイスに送ることができる。

10

【0090】

チェックサムワード3005は、(a)CHKワード(CHKワードを繰り返す)、および/または(b)ハートビート/SYNCワードを含む、任意のCCleワードに後続することができる。CCle送信機および受信機は、チェックサム機能をサポートすることができる。たとえば、CCle送信機は、CCle受信機が次に使用するチェックサムワード3005を送ることができる。

【0091】

受信機デバイスがチェックサム(CHK)エラーを検出する場合、(a)SYNCワードもしくはWAKEUPイベントまで、CCleアクション(たとえば、レジスタ書込み)をサスペンドすることができ、(b)IRQを介してCHKエラーをレポートすることができ、(c)CHKエラーイベントを状態レジスタに設定することができ、かつ/または(d)チェックサムエラーを無視することができる。

20

【0092】

チェックサム(CHK)ワードは、任意のワード境界に挿入され得る。これによって、チェックサムの使用が、異なるシステム/プロトコル内で容易にポータブルになり、また、スケラブルなデータ品質になる(たとえば、信頼性がより低いバスを介して、より多くのCHKを使用する)。

【0093】

いくつかの実装形態では、チェックサム(CHK)サポートは必須でなくてもよい。これによって、低層の製品にとっての負担がそれほど生じず、また、チェックサムの使用が必要とされない場合、なだらかな下落が可能になる。

30

【0094】

《第2の例示的なエラー検出手法》

図29～図31に示すフォーマットを有するワード内のシステムエラーを検出することは、必ずしも信頼性が高いとは限らず、いくつかのシステムにとっては問題のある場合がある。たとえば、医療、自動車、航空宇宙、または軍事的使用など、高いデータの完全性を必要とするバスシステムでは、すべてのシンボルエラーをすぐに検出することが望ましいことが多い。この問題に対処するために、100%のシンボルエラー検出が達成可能である、代替CCleワードフォーマットを開示する(すなわち、シンボルスリップなしのワードごとの単一シンボルエラー)。この目的で、まず、本明細書で開示する態様は、部分的に、図23～図28における観測に依拠しており、これらの観測では、符号化ワードが、既知の定数を用いてすべてが固定されたそれらの3桁の最下位ビット(LSB)とともに送信されるとき、100%のシンボルエラー検出が達成され得ることに留意されたい。しかしながら、3桁のLSBを利用して、エラービット検出定数を表すには、3桁のLSBが、それらのビット値が制御され得るように柔軟であることが必要である。

40

【0095】

図30におけるCCleチェックサムワード3005では、たとえば、ビット1および2がCCleワードのための制御コードに割り振られる。そのような制御コードは、ワードが読取り動作を

50

含むか、書込み動作を含むかを識別し、CCleワードの動作に影響を及ぼすことなく修正することはできない。

【 0 0 9 6 】

1つの解決策は、3桁のLSBが3ビットのエラー検出定数を表すことができるように、図30に示すCCleワードフォーマットを修正することである。そのような修正の一例は、図13～図15に提供されており、その例は、必要に応じて、データスループットまたはエラー検出を最適化することができる、柔軟なビット割振り方式を実装する。

【 0 0 9 7 】

図32は、エラー検出を容易にする例示的な柔軟なCCleワードフォーマット3202を示す。この20ビットワード3202では、ビット[0]がエラー検出定数3204のために割り振られ、ビット[2,1]がデータのため、またはエラー検出定数を拡張するためのいずれか3206に割り振られ、ビット[4,3]が制御ビット3208のために割り振られ、ビット[18,5]が14ビットのデータフィールド3210のために割り振られ、ビット[19]が前述の「余剰」ビット3212のために割り振られる。すなわち、ビット[2,1]が、エラー訂正のために使用することができない制御コード3228のために使用され得る、代替CCleワードフォーマット3220に対して、柔軟なCCleワード3202は、2つの制御ビットをビット[2,1]から離して移動させ、それによって、データスループットを最適化するために(すなわち、ビット[18,5]およびビット[2,1]を介して、16ビットをデータのために割り振ることによって)データのために、または、エラー検出を最適化するために(すなわち、前述の100%エラー検出を達成するために、3桁のLSBをエラー検出定数のために割り振ることによって)、エラー検出定数拡張のために、ビット[2,1]を利用する柔軟性を提供する。したがって、ユーザは、必要に応じて、非100%のシンボルエラー検出とともに16ビットのデータを有する符号化ワードを送ること、または、100%のシンボルエラー検出とともに14ビットのデータを有する符号化ワードを送ることの間で選ぶことができる。

【 0 0 9 8 】

様々なビット割振り方式のいずれもが、本明細書で開示する態様の範囲内に入るように企図されることに留意されたい。たとえば、図32は、この特定の例では、LSBにより近く割り振るための(すなわち、より高い値のMSBにより近く割り振るのではない)、制御ビットのためのビット[4,3]の割振りを示すが、他のビット割振り方式は、これらの制御ビットを、ビット[18,3]内のビットの任意の組合せにおいて配置することを含み得る。また、図32は20ビットのCCleワードフォーマットを示すが、ビット割振り方式は、任意のビット長さのワードについて同様に企図され、データまたはエラー検出定数拡張のために割り振られた固定数の最下位ビットが、適宜に増加/減少され得ることに留意されたい。

【 0 0 9 9 】

さらに、本明細書で開示する態様は、第1のビット割振り方式と第2のビット割振り方式との間で動的に切り替えることを容易にすることができ、第1のビット割振り方式は、データ最適化(すなわち、データのための16ビットの割振り)に向けられ、第2のビット割振り方式は、エラー検出最適化(すなわち、エラー検出定数のための3桁のLSBの割振り)に向けられる。

【 0 1 0 0 】

《例示的なクロック/シンボルスリップエラー検出》

クロック消失または余剰クロックによって引き起こされるシンボルスリップエラーは、エラー検出定数および/またはチェックサムによって検出されない場合がある。しかしながら、これらのタイプのエラーの大多数は、受信機デバイスにおいて、次のワードにおいて、および/または状態機械を使用して検出され得る。

【 0 1 0 1 】

図35は、クロックエラーなしの例示的なCCleワード送信を示す。第1のライン(SDA)3502と第2のライン(SCL)3504とを備える共有バスを使用して、12個のシンボル3512(CCleワードを構成する)が開始条件3508と3510との間で送信される。そのような開始条件3508および3510は、第2のライン(SCL)3504が高である間に、第1のライン(SDA)3502上の高から低へ



の遷移において発生する。すべてのCCle準拠デバイスは、これが、受信機デバイスの状態機械によって追跡され得る期待CCleワード境界における、開始条件(すなわち、第2のライン(SCL)3504が高である間の第1のライン(SDA)3502上の高から低への遷移)であると理解する。加えて、シンボル間遷移内に埋め込まれたクロック3506も示されている。

#### 【0102】

図36は、単一クロック消失エラーありの例示的なCCleワード送信を示す。すなわち、この例では、クロックパルス3602が、受信機デバイスによって検出されておらず、それが、対応するシンボル「2」が無視されることを引き起こす。この状態3604は、シンボルスリップと呼ばれることがある。受信機デバイスがクロックを消失したので、受信機デバイスは、第1のライン3502および第2のライン3504からシンボルを読み取り続け、第2の開始条件3510をシンボルとして解釈することになり、次のシンボル3602が開始条件であると期待することになる。その次のシンボル3602は決して開始条件にならないので、クロック消失エラーが受信機デバイスによって検出されるのは、この点である。次に、受信機デバイスは、同期消失を仮定し、その受信機論理をリセットすることができる。

#### 【0103】

図37は、ダブル消失クロックエラーありの例示的なCCleワード送信を示す。すなわち、この例では、2つのクロックパルス3702が、受信機デバイスによって検出されず、それが、シンボル「2」および「3」3704が無視されることを引き起こす。この状態は、ダブルシンボルスリップと呼ばれることがある。受信機デバイスが2つのクロックを消失したので、受信機デバイスは、第1のライン3502および第2のライン3504からシンボルを読み取り続け、第2の開始条件3510をシンボルとして解釈することになる。9回中8回の場合について、受信機デバイス論理は、期待される開始条件が(次のワード3702の第1のシンボルと第2のシンボルとの間で)消失していることを認識し、したがって、ダブルクロック消失を検出することになる。受信機デバイスは、いつ開始条件が期待されるかを追跡するために、状態機械論理を含み得る。次のワード3702の第1のシンボルと第2のシンボルとの間の遷移が開始条件(すなわち、3から1へのシンボル遷移)として検出されることになる、9分の1の確率について、受信機デバイスは、次のワードにおいてこのダブルクロックエラーを検出することができる(すなわち、次のCCleワードにおけるある点において、第1のシンボルと第2のシンボルとの間の遷移が開始条件にならず、受信機デバイスによってエラーが検出される)。

#### 【0104】

図38は、余剰クロックエラーありの例示的なCCleワード送信を示す。すなわち、この例では、余剰クロックパルス3802が、受信機デバイスによって検出されており、それが、余剰シンボル「1」が読み取られることを引き起こす。この状態3804下で、受信機デバイスは、最後の有効なシンボル遷移3806中に次の開始条件を期待することになる。9回中8回の場合では、CCleワードの最後の2つのシンボルにおけるシンボルの組合せが開始条件を生じないことになるので、受信機デバイスは、同期エラーを認識することになる。9回中1回の場合では、CCleワードの最後の2つのシンボルが、受信機デバイスによって開始条件(すなわち、3から1へのシンボル遷移)として認識されることになる。その状況では、受信機デバイスは、このエラーを検出するために、現在および/または次のCCleワードのエラー検出定数に依拠することができる。

#### 【0105】

##### 《例示的な柔軟なエラー検出を用いるデバイス》

図33は、本開示によるマスタ/スレーブデバイスの例示的な構成要素を示すブロック図である。図示のように、マスタ/スレーブデバイス3314は、制御データバス3350を介してマスタ/スレーブデバイス3360に結合される。ここで、マスタ/スレーブデバイス3314またはマスタ/スレーブデバイス3360のいずれかが、本明細書で開示する前述の態様に従って、マスタまたはスレーブとして動作することができること、ならびに、マスタ/スレーブデバイス3314およびマスタ/スレーブデバイス3360が、実質的に同様の構成要素を有し得ることが企図される。

## 【0106】

この例では、マスタ/スレーブデバイス3314は、バス3302によって概略的に表される内部バスアーキテクチャを用いて実装され得る。バス3302は、マスタ/スレーブデバイス3314の具体的な適用例および全体的な設計制約に応じて、任意の数の相互接続するバスおよびブリッジを含む場合がある。バス3302は、(処理回路3304によって概略的に表される)1つまたは複数のプロセッサ、メモリ3305、および(コンピュータ可読媒体3306によって概略的に表される)コンピュータ可読媒体を含む様々な回路を互いにリンクさせる。バス3302は、タイミングソース、周辺機器、電圧調整器、および電力管理回路などの様々な他の回路をリンクさせることもできるが、これらの回路は当技術分野でよく知られており、したがって、これ以上は説明しない。バスインターフェース/回路3308は、制御データバス3350とマスタ/スレーブデバイス3314との間のインターフェースを提供する。

10

## 【0107】

本開示の一態様では、コンピュータ可読媒体3306は、本明細書で開示するようなCCleプロトコル通信を容易にするために、様々な命令3306a、3306b、および/または3306cを含むように構成される。同様の態様では、そのような通信は、代わりに、処理回路3304を図示のように回路3320、3330、および/または3340のうちのいずれかに結合することによって、ハードウェアを介して実装され得る。代替的に、処理回路3304は、回路3320、3330、および/または3340のうちのいずれかを含み、かつ/または実装することができる。その上、CCle通信は、命令3306a、3306b、および/または3306cの任意の組合せ、ならびに、回路3320、3330、および/または3340の任意の組合せによって容易にされ得ることが企図される。

20

## 【0108】

たとえば、エンコーダ/デコーダ命令3306aおよびエンコーダ/デコーダ回路3320は、CCleプロトコルに従ってワードを符号化/復号することに向けられる。前述のように(図2~図21参照)、そのような符号化/復号は、3進数を桁ごとに複数のシンボルに変換して(たとえば、12桁の3進数は12シンボルになる)、前述の「余剰ビット」(ビット19)を生じることを含み得る。

## 【0109】

本開示の別の態様では、エラー検出命令3306bおよびエラー検出回路3330は、図23~図32において説明し、示したように、かつ/または、所望のCCleワードフォーマットに従って、エラー検出を実施することに向けられる。この目的で、図31~図32のように、本明細書で開示する、企図されたCCleワードフォーマットは、符号化された20ビットワードを備え、3桁の最下位ビットがデータまたはエラー検出のために割り振られる。その上、エラー検出命令3306bおよび/またはエラー検出回路3330のいずれかは、柔軟なビット割振りを容易にするように構成されてよく、最下位ビットがエラー検出のために割り振られ、第2の最下位ビットおよび第3の最下位ビットの各々がデータまたはエラー検出のいずれかのために割り振られることが企図される。さらに、図23~図30および図35~図38のように、エラー検出命令3306bおよび/またはエラー検出回路3330のいずれかは、固定最下位ビットを使用して、および/もしくはプロトコルに追加されるチェックサムワードとともに、ならびに/または、消失した開始条件を検出する受信機デバイス論理を使用して、エラー検出を容易にするように構成され得ることが企図される。

30

40

## 【0110】

本開示の別の態様では、チェックサム命令3306cおよび/またはチェックサム回路3340が、バス3350から受信および/または送信されるワードのためのチェックサムを生成するように構成され得る。特に、チェックサム命令3306cおよび/またはチェックサム回路3340のいずれかは、CCleチェックサムワードをサポートするために、チェックサム計算を容易にするように構成され得る。

## 【0111】

処理回路3304は、バス3302を管理すること(すなわち、デバイス3314がマスタデバイスであるとき)、および、コンピュータ可読媒体3306に記憶されたソフトウェアの実行を含

50

む全体的な処理を担当することを諒解されたい。ソフトウェアは、処理回路3304によって実行されるとき、マスタ/スレーブデバイス3314に、任意の特定の装置のための下記で説明する様々な機能を実施させる。コンピュータ可読媒体3306は、ソフトウェアを実行するときに処理回路3304によって操作されるデータを記憶するために使用することもできる。

#### 【0112】

処理回路3304内の1つまたは複数のプロセッサは、ソフトウェアを実行することができる。ソフトウェアは、ソフトウェア、ファームウェア、ミドルウェア、マイクロコード、ハードウェア記述言語と呼ばれるか、または他の名称で呼ばれるかどうかにかかわらず、命令、命令セット、コード、コードセグメント、プログラムコード、プログラム、サブプログラム、ソフトウェアモジュール、アプリケーション、ソフトウェアアプリケーション、ソフトウェアパッケージ、ルーチン、サブルーチン、オブジェクト、実行可能ファイル、実行スレッド、プロシージャ、機能などを意味するように広く解釈されるべきである。ソフトウェアは、コンピュータ可読媒体3306上に存在することができる。コンピュータ可読媒体3306は、非一時的コンピュータ可読媒体である場合がある。非一時的コンピュータ可読媒体には、例として、磁気ストレージデバイス(たとえば、ハードディスク、フロッピー(登録商標)ディスク、磁気ストリップ)、光ディスク(たとえば、コンパクトディスク(CD)またはデジタル多用途ディスク(DVD))、スマートカード、フラッシュメモリデバイス(たとえば、カード、スティック、またはキードライブ)、ランダムアクセスメモリ(RAM)、読取り専用メモリ(ROM)、プログラマブルROM(PROM)、消去可能PROM(EPROM)、電気的消去可能PROM(EEPROM)、レジスタ、リムーバブルディスク、ならびに、コンピュータがアクセスし読み取ることができるソフトウェアおよび/または命令を記憶するための任意の他の適切な媒体が含まれる。コンピュータ可読媒体には、例として、搬送波、伝送路、ならびに、コンピュータがアクセスし読み取ることができるソフトウェアおよび/または命令を送信するための任意の他の適切な媒体も含まれ得る。コンピュータ可読媒体3306は、マスタ/スレーブデバイス3314の中に、マスタ/スレーブデバイス3314の外部に存在するか、またはマスタ/スレーブデバイス3314を含む複数のエンティティにわたって分散される場合がある。コンピュータ可読媒体3306は、コンピュータプログラム製品内で具現化される場合がある。例として、コンピュータプログラム製品には、パッケージング材料内のコンピュータ可読媒体が含まれ得る。当業者は、特定の適用例および全体的なシステムに課された設計制約全体に応じて、本開示全体にわたって提示された記載の機能を最もよく実装する方法を認識されよう。

#### 【0113】

図34は、プロトコル(たとえば、CCleプロトコル)の1つまたは複数のエラー検出機能に従って通信を容易にする例示的な方法を示す。3402で、共有バスを介して送信されるべき複数のビットが取得され、複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかが、エラー検出のために使用され得る。3404で、複数のビットが3進数に変換される。3406で、3進数の桁が順次シンボルに変換され、クロックがシンボル間遷移内に埋め込まれる。3408で、複数のシンボルが送信のためにワードに結合される。次に、3410で、ワードが共有バスを介して受信デバイスへ送信され得る。加えて、3412で、チェックサムが送信されるビットのために計算され得る。3414で、送信されるビットのためのチェックサムが受信デバイスへ送られ得る。

#### 【0114】

一態様によれば、バスインターフェースと処理回路とを備える送信機デバイスが提供される。バスインターフェースは、送信機デバイスを、(受信またはスレーブデバイスが結合される)共有バスに結合するように働き得る。送信機デバイスは、共有バスを介した通信を管理または制御することができる。処理回路は、(a)共有バスを介して送信されるべき複数のビットを取得することであって、複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかが、エラー検出のために使用されること、(b)複数のビットを3進数に変換すること、(c)3進数の桁を順次シンボルに変換すること、(d)複数のシンボルを送信のためにワードに結合すること、および/あるいは(e)共有バスを介して受信デバイ

スワードを送信することを行うように構成され得る。クロックは、シンボル間遷移内に埋め込まれ得る。共有バスは2ラインバスであってよく、2ラインバスの両方のラインが、シンボルを転送するために使用される。処理回路は、(a)送信されるビットのためのチェックサムを計算すること、および/または(b)送信されるビットのためのチェックサムを受信デバイスへ送信することを行うようにさらに構成され得る。チェックサムは、受信デバイスへの送信のために、いずれかの任意のワード内に挿入され得る。

#### 【0115】

別の態様によれば、バスインターフェースと処理回路とを備える受信機デバイスが提供される。バスインターフェースは、その上で通信がマスタデバイスによって管理される共有バスに結合するように働き得る。処理回路は、(a)共有バスを介して複数のシンボルを受信すること、(b)複数のシンボルを桁に変換すること、(c)桁を3進数に結合すること、(d)3進数をビットに変換することであって、複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかが、エラー検出のために使用されること、および/あるいは(e)複数のビットにおける3桁の最下位ビットまたは最下位のもののいずれかに基づいて、受信された複数のビット内のエラーを確かめることを行うように構成され得る。クロックは、複数の受信されたシンボルのシンボル間遷移内に埋め込まれ得る。共有バスは2ラインバスであってよく、2ラインバスの両方のラインが、シンボルを転送するために使用される。受信機デバイスは、クロック信号内で誤ったクロックパルス消失と誤った余剰クロックパルスとを検出する、状態機械論理回路を実装することができる。

#### 【0116】

図に示す構成要素、ステップ、特徴、および/または機能のうちの1つまたは複数は、並べ替えられてもよく、ならびに/あるいは、単一の構成要素、ステップ、特徴、または機能に組み合わされてもよく、またはいくつかの構成要素、ステップ、または機能で具現化されてもよい。また、本明細書で開示する新規の特徴から逸脱することなく追加の要素、構成要素、ステップ、および/または機能が、同様に追加され得る。図に示す装置、デバイス、および/または構成要素は、図に記載した方法、特徴、またはステップのうちの1つまたは複数を実施するように構成され得る。本明細書で説明する新規のアルゴリズムはまた、効率的にソフトウェアに実装され、および/またはハードウェアに埋め込まれ得る。

#### 【0117】

さらに、実施形態は、フローチャート、流れ図、構造図、またはブロック図として示されるプロセスとして説明される場合があることに留意されたい。フローチャートは動作を順次プロセスとして説明する場合があるが、動作の多くは並行してまたは同時に実施され得る。さらに、動作の順序は並べ替えることができる。プロセスは、その動作が完了したとき、終了する。プロセスは、方法、関数、手順、サブルーチン、サブプログラムなどに対応し得る。プロセスが関数に対応するときには、その終了は、その関数が、呼出し関数またはメイン関数に戻ることに対応する。

#### 【0118】

その上、記憶媒体は、読取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、磁気ディスク記憶媒体、光記憶媒体、フラッシュメモリデバイス、および/または情報を記憶するための他の機械可読媒体を含む、データを記憶するための1つまたは複数のデバイスを表す場合がある。「機械可読媒体」という用語は、限定はしないが、ポータブルまたは固定ストレージデバイス、光ストレージデバイス、ワイヤレスチャネル、ならびに命令および/またはデータを記憶するか、包含するか、または搬送することができる様々な他の媒体を含む。

#### 【0119】

さらに、実施形態は、ハードウェア、ソフトウェア、ファームウェア、ミドルウェア、マイクロコード、またはそれらの任意の組合せによって実装され得る。ソフトウェア、ファームウェア、ミドルウェア、またはマイクロコードで実装されるとき、必要なタスクを実施するためのプログラムコードまたはコードセグメントは、記憶媒体または他のストレージなどの機械可読媒体に記憶され得る。プロセッサは必要なタスクを実施することがで

きる。コードセグメントは、手順、関数、サブプログラム、プログラム、ルーチン、サブルーチン、モジュール、ソフトウェアパッケージ、クラス、または命令、データ構造もしくはプログラムステートメントの任意の組合せを表すことができる。コードセグメントは、情報、データ、引数、パラメータ、またはメモリ内容を渡し、かつ/または受け取ることによって、別のコードセグメントまたはハードウェア回路に結合することができる。情報、引数、パラメータ、データなどは、メモリ共有、メッセージパッシング、トークンパッシング、ネットワーク送信などを含む任意の適切な手段を介して渡すことができるか、転送することができるか、または送信することができる。

#### 【0120】

本明細書で開示する例に関して説明する様々な例示的な論理ブロック、モジュール、回路、要素、および/または構成要素は、汎用プロセッサ、デジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)もしくは他のプログラマブル論理構成要素、個別ゲートもしくはトランジスタ論理、個別ハードウェア構成要素、または本明細書で説明する機能を実施するように設計されたそれらの任意の組合せで実装または実施され得る。汎用プロセッサはマイクロプロセッサであり得るが、代替としてプロセッサは、任意の従来のプロセッサ、コントローラ、マイクロコントローラ、または状態機械であり得る。プロセッサはまた、コンピューティング構成要素の組合せ、たとえば、DSPとマイクロプロセッサとの組合せ、いくつかのマイクロプロセッサ、DSPコアと連係した1つまたは複数のマイクロプロセッサ、あるいは他の任意のそのような構成として実装され得る。

#### 【0121】

本明細書で開示する例に関して説明する方法またはアルゴリズムは、直接ハードウェアにおいて、プロセッサによって実行可能なソフトウェアモジュールにおいて、または両方の組合せにおいて、処理ユニット、プログラミング命令、または他の指示の形態で具現化されてよく、かつ、単一のデバイスに含まれてよく、または複数のデバイスにわたって分散されてよい。ソフトウェアモジュールは、RAMメモリ、フラッシュメモリ、ROMメモリ、EPROMメモリ、EEPROMメモリ、レジスタ、ハードディスク、リムーバブルディスク、CD-ROM、または当技術分野で知られている任意の他の形態の記憶媒体に存在することができる。記憶媒体は、プロセッサがその記憶媒体から情報を読み取り、かつその記憶媒体に情報を書き込むことができるように、プロセッサに結合され得る。代替として、記憶媒体は、プロセッサと一体化され得る。

#### 【0122】

本明細書で開示する実施形態に関して説明する、様々な例示的な論理ブロック、モジュール、回路、およびアルゴリズムステップは、電子ハードウェア、コンピュータソフトウェア、または両方の組合せとして実装され得ることを当業者はさらに諒解されよう。ハードウェアとソフトウェアのこの互換性を明確に示すために、様々な例示的な構成要素、ブロック、モジュール、回路、およびステップを、上記では概してそれらの機能の観点から説明した。そのような機能がハードウェアとして実装されるか、またはソフトウェアとして実装されるかは、特定の適用例および全体的なシステムに課された設計制約によって決まる。

#### 【0123】

本明細書で説明する本発明の様々な特徴は、本発明から逸脱することなく、異なるシステムにおいて実施され得る。上記の実施形態は例にすぎず、本発明を限定するものと解釈すべきではないことに留意されたい。実施形態の説明は、例示的なものであり、特許請求の範囲を限定するものではない。したがって、本教示は、他のタイプの装置に容易に適用することができ、多くの代替形態、変更形態、および変形形態が当業者には明らかであろう。

#### 【符号の説明】

#### 【0124】

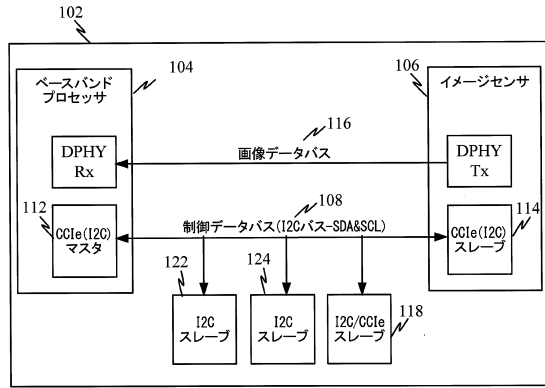
104	ベースバンドプロセッサ	
106	イメージセンサ	
108	マルチモード制御データバス、制御データバス、共有制御データバス、共有バス、制御バス	
112	デバイス、マスタデバイス、マルチモードマスタデバイス、マスタ	
114	デバイス、スレーブデバイス	
116	画像データバス	
118	デバイス、周辺デバイス	
122、124	周辺デバイス	
202	元のクロック	10
204	データ	
206	送信シンボル	
208	クロック情報	
210	元のデータ	
302	送信機	
304	データビット	
308	ビット-遷移番号変換器ブロック	
310	遷移-シンボルブロック	
316、322、404	順次シンボル	
320	受信機	20
324	クロックラインSCL、SCLライン	
326	データラインSDA、SDAライン	
328	クロック-データ復元(CDR)ブロック	
330	シンボル-遷移番号変換器ブロック	
332	遷移番号-ビット変換器	
402	遷移番号	
406	第1のサイクル	
408	第2のサイクル	
410	第3のサイクル	
412	第4のサイクル	30
502	送信機側(TX:TからSへ)	
504	受信機側(RX:SからTへ)	
506、1412、1514、1616、1716、2300、2400、2500、2600、2700、2800	テーブル	
602	3進数	
1102	一般呼出し、I2C一般呼出し	
1104	「CCleモード」バイトまたはインジケータ	
1202	CCle呼出し、終了呼出し	
1204	「終了」コード/インジケータ、一般呼出し	
1206	「一般呼出し」	
1208	「終了」コード	40
1302	CCle SIDワードフォーマット	
1304	16ビットのスレーブ識別子(SID)	
1402	CCleアドレスワードフォーマット	
1404	16ビットのアドレス	
1406	アドレスワード	
1408、1504、1608、1706	制御コード	
1410、1510、1610、1708	エラー検出定数	
1500	書込みデータワードフォーマット、データワード、書込みデータワード	
1502	16ビットの書込みデータ部分	
1600	読取り指定ワードフォーマット、読取り指定データワード	50

1604	16ビットの読取りデータ値部分、「読取り指定」(RS)ワード	
1607	最後のアドレスワード	
1612	「読取り指定」(RS)ワード	
1702	読取りデータワードフォーマット、読取りデータワード	
1704	16ビットの読取りデータ値部分	
1707	SID	
1802	シリアルデータラインSDA、SDAライン	
1804	シリアルクロックラインSCL、SCLライン	
1806、1906、1908、1910	START条件	
1808	スレーブID	10
1902	SDAライン	
1904	SCLライン	
2202	タイミング図	
2204	クロック消失	
2206	余剰クロック	
2208	シンボルエラー	
2210	データビット	
2212、2214	サイクル	
2216、2218	余剰シンボル「01」	
2220	余剰クロックサイクル	20
2222	単一シンボルエラー	
2302	(ビット[19:0])0000_0000_0000_0000_0000の20ビットシーケンス	
2304	3進数(T11...T0)0000_0000_0000 <sub>3</sub>	
2306	順次シンボル(S11...S0)0321_0321_0321、順次シンボル0321_0321_0321	
2308、2408、2508、2608、2708、2808	3桁の最下位ビット	
2310、2410、2510、2610、2710、2810	誤ったシンボル	
2402	(ビット[19:0])0100_0000_1101_1111_1000の20ビットシーケンス	
2404	3進数(T11...T0)1111_1111_1111 <sub>3</sub>	
2406	順次シンボル(S11...S0)2301_2301_2301、順次シンボル2301_2301_2301	
2502	(ビット[19:0])1000_0001_1011_1111_0000の20ビットシーケンス	30
2504	3進数(T11...T0)2222_2222_2222 <sub>3</sub>	
2506	順次シンボル(S11...S0)3131_3131_3131、順次シンボル3131_3131_3131	
2602	(ビット[19:0])0001_1000_1111_0011_1000の20ビットシーケンス	
2604	3進数(T11...T0)0120_1201_2012 <sub>3</sub>	
2606	順次シンボル(S11...S0)0132_3101_3231、順次シンボル0132_3101_3231	
2702	(ビット[19:0])0100_1010_1101_1010_1000の20ビットシーケンス	
2704	3進数(T11...T0)1201_2012_0120 <sub>3</sub>	
2706	順次シンボル(S11...S0)2030_2120_3021、順次シンボル2030_2120_3021	
2802	(ビット[19:0])0101_1110_1101_0000_1000の20ビットシーケンス	
2804	3進数(T11...T0)2012_0120_1201 <sub>3</sub>	40
2806	順次シンボル(S11...S0)3231_0132_3101、順次シンボル3231_0132_3101	
2902	CCleワード、第1のCCleワード	
2903	3桁の最下位ビット	
2904	第2のCCleワード	
2905	最下位ビット	
3003	チェックサム値	
3004a	CCleアドレスワード	
3004b、3008c、3008d	書込みワード	
3004c、3012c、3012d	読取りワード	
3005	チェックサムワード	50

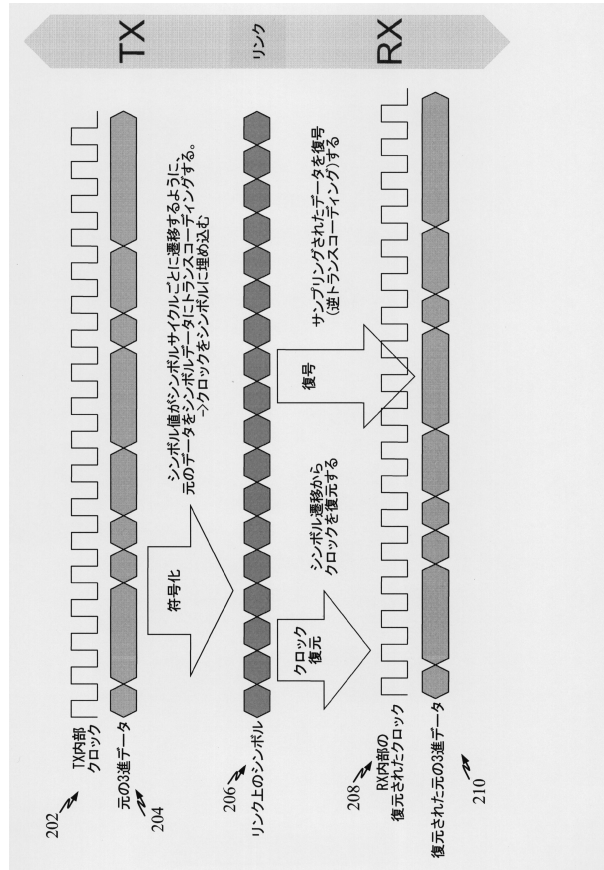
3006	書込みコマンド	
3008a、3008b、3012a、3012b	アドレスワード	
3010	読取りコマンド	
3102	チェックサム発生器	
3104	チェックサム、チェックサム値	
3110	第1のCCleワードN	
3112	第2のCCleワードN+1	
3114	第3のCCleワードN+2	
3116	第4のCCleワードN+3	
3202	例示的な柔軟なCCleワードフォーマット、20ビットワード、柔軟なCCleワード	10
3204	エラー検出定数	
3206	データのため、またはエラー検出定数を拡張するためのいずれか	
3208	制御ビット	
3210	14ビットのデータフィールド	
3212	「余剰」ビット	
3220	代替CCleワードフォーマット	
3228	制御コード	
3302	バス	
3304	処理回路	
3305	メモリ	20
3306	コンピュータ可読媒体	
3306a	命令、エンコーダ/デコーダ命令	
3306b	命令、エラー検出命令	
3306c	命令、チェックサム命令	
3308	バスインターフェース/回路	
3314	マスタ/スレーブデバイス、デバイス	
3314	マスタ/スレーブデバイス	
3320	回路、エンコーダ/デコーダ回路	
3330	回路、エラー検出回路	
3340	回路、チェックサム回路	30
3350	制御データバス、バス	
3360	マスタ/スレーブデバイス	
3502	第1のライン(SDA)、第1のライン	
3504	第2のライン(SCL)、第2のライン	
3506	クロック	
3508	開始条件	
3510	開始条件、第2の開始条件	
3512	12個のシンボル	
3602	クロックパルス、次のシンボル	
3604、3804	状態	40
3702	2つのクロックパルス、次のワード	
3704	シンボル「2」および「3」	
3802	余剰クロックパルス	
3806	シンボル遷移	



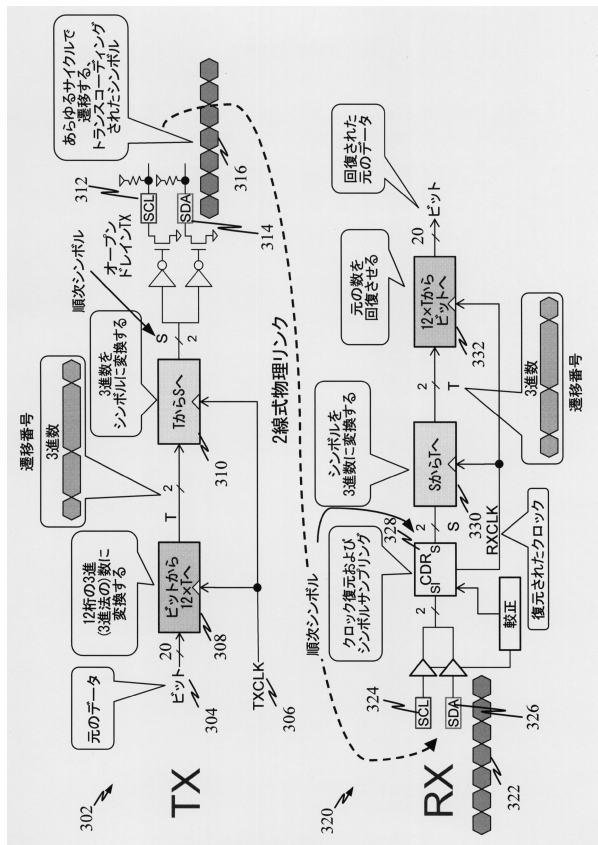
【 図 1 】



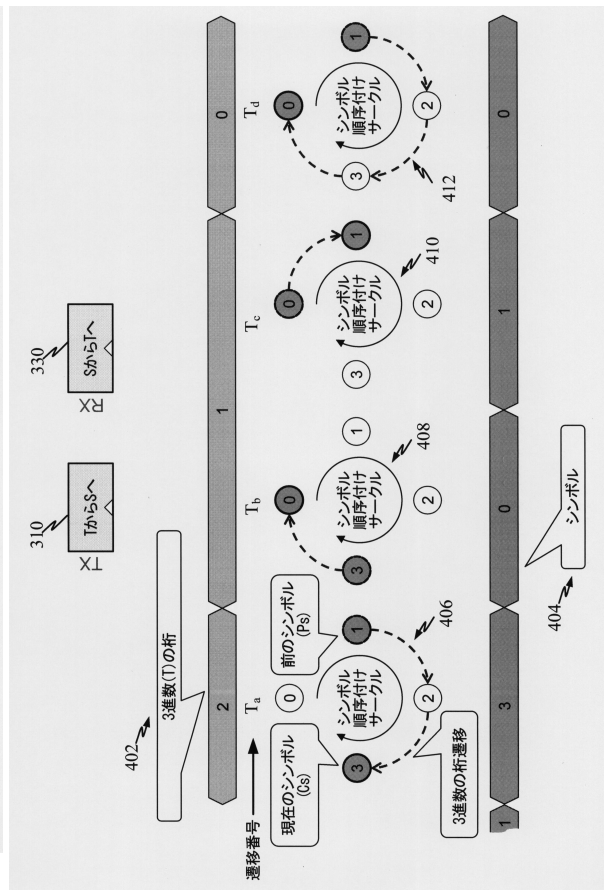
【 図 2 】



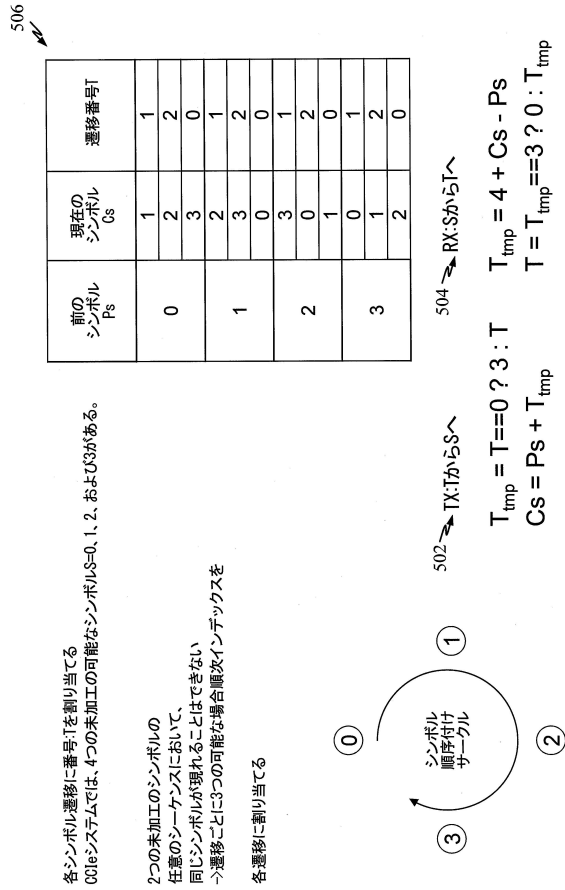
【 図 3 】



【 図 4 】

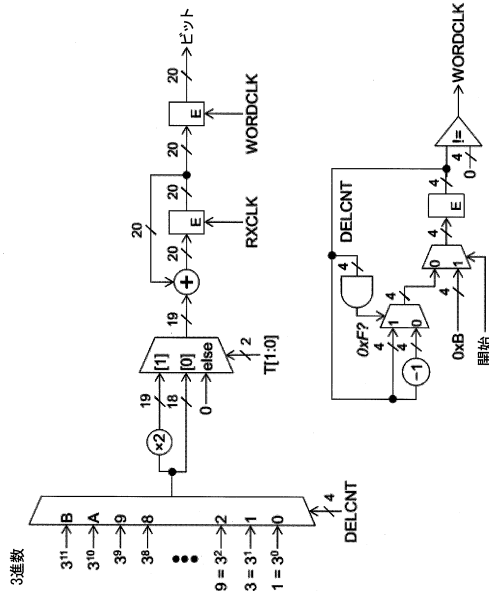


【図 5】



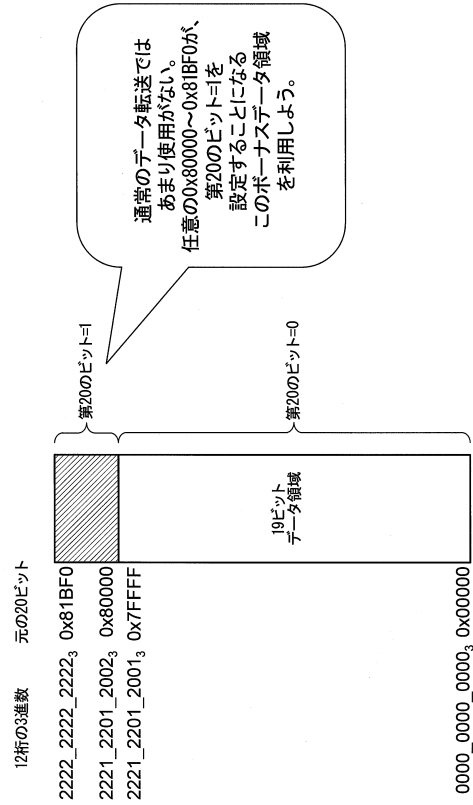
【図 9】

MSS最優先Ccie:12×T-ビット回路



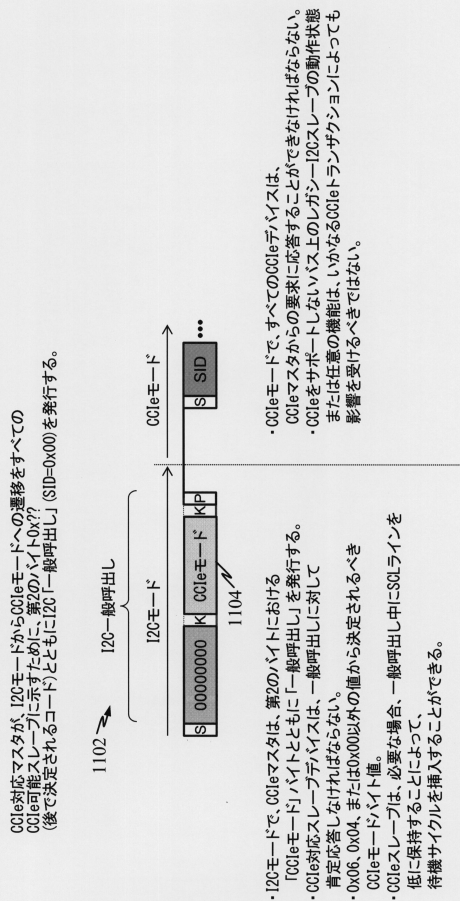
【図 10】

ビット19(第20のビット)



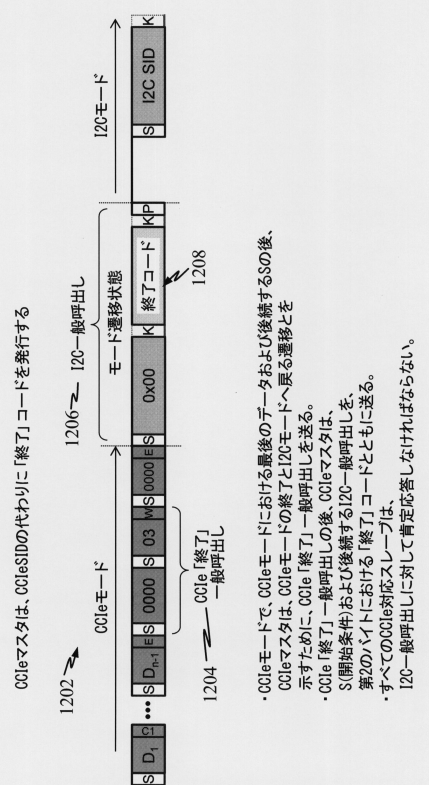
【図 11】

Ccieモードからの例示的なエントリインジケータ

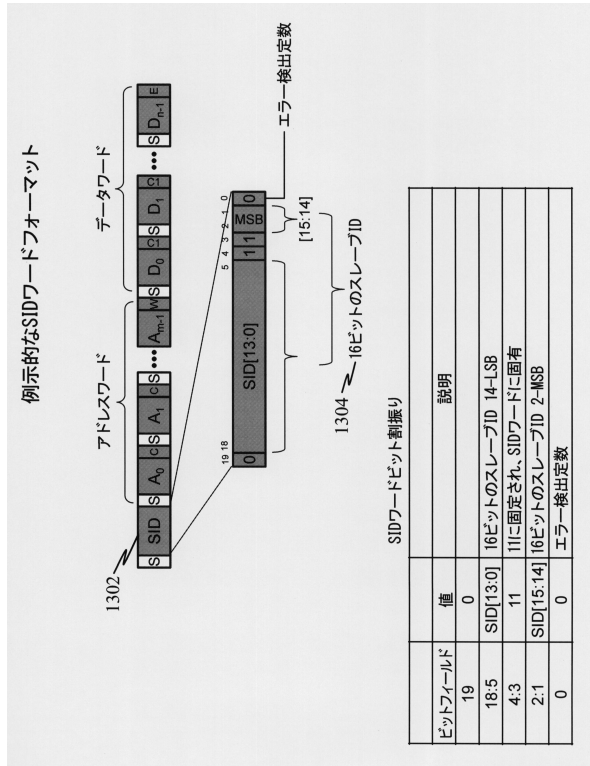


【図 12】

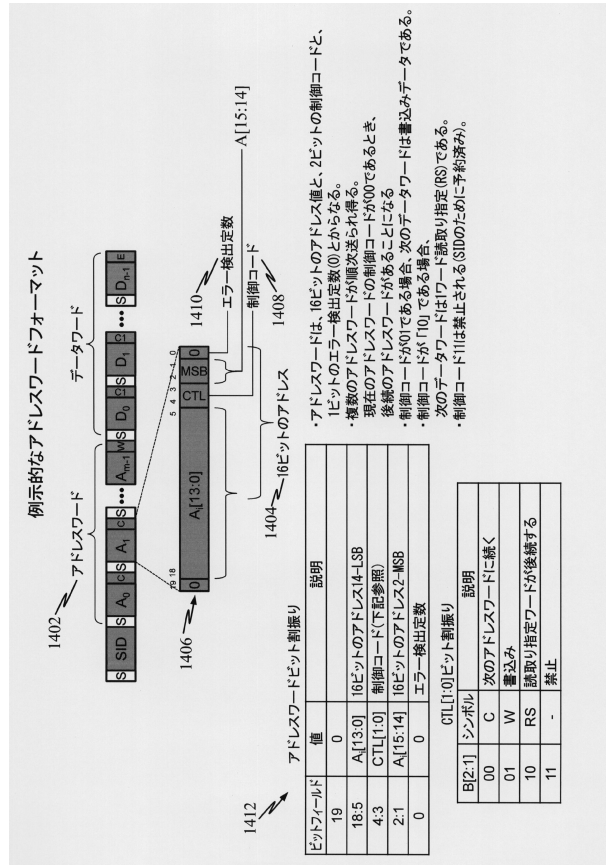
例示的なCcieモードからの終了インジケータ



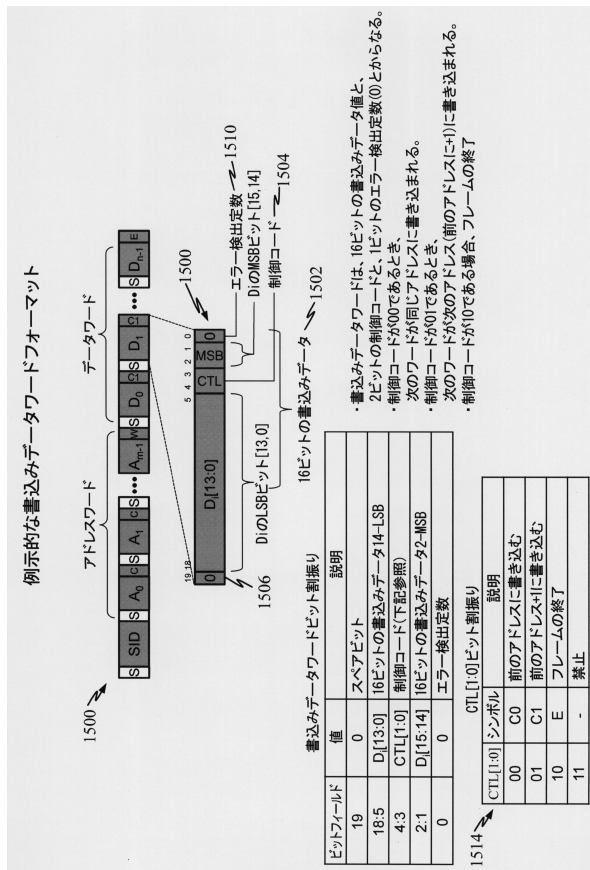
【図 13】



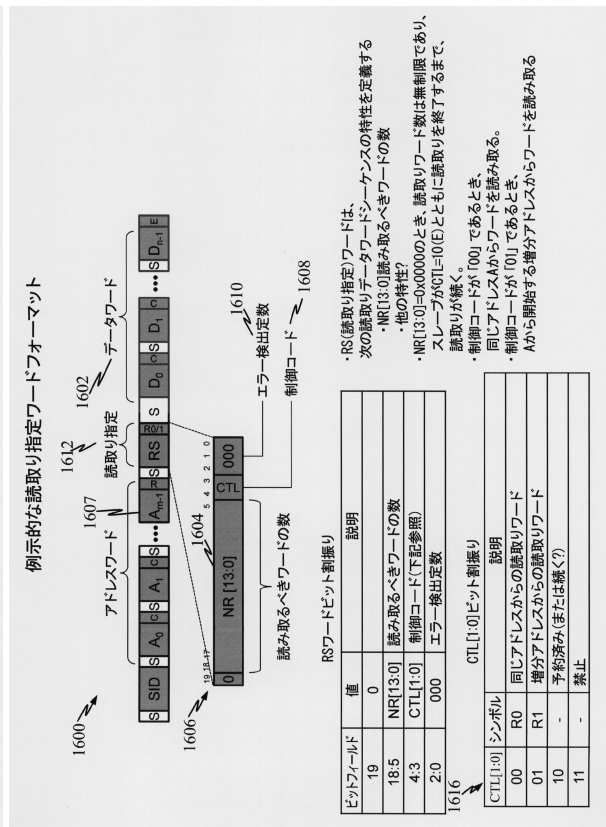
【図 14】



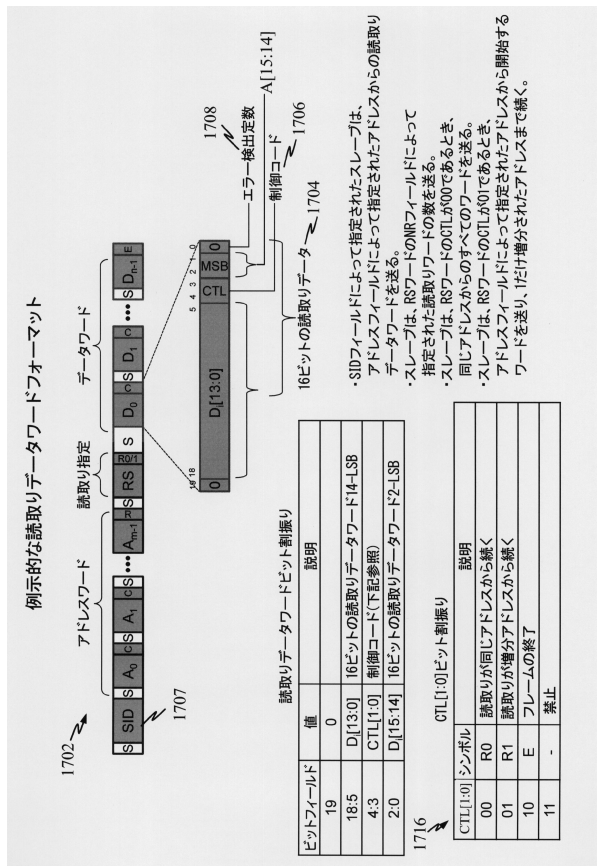
【図 15】



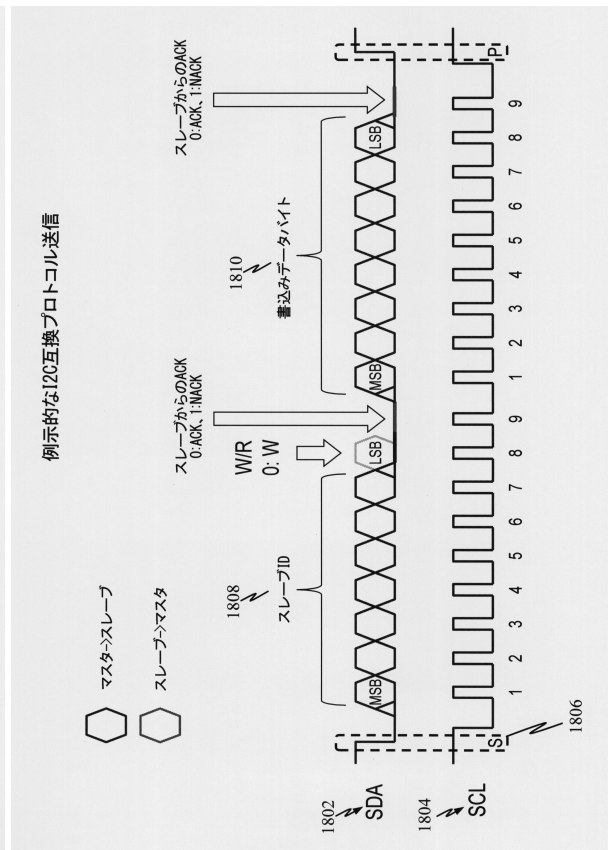
【図 16】



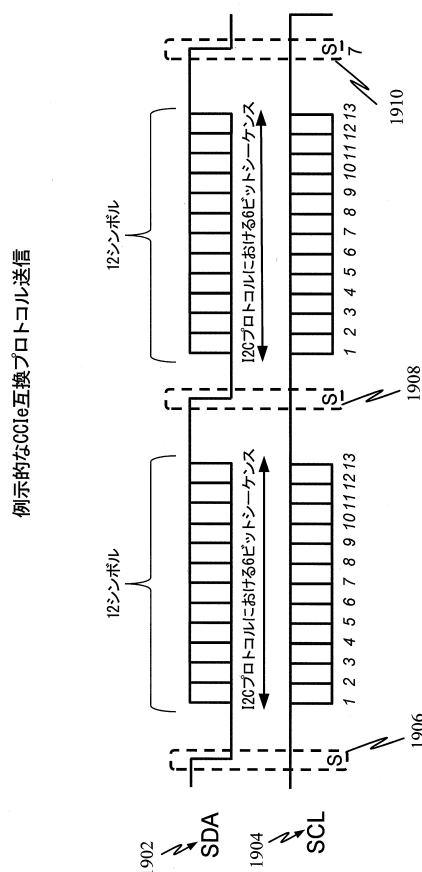
【 図 1 7 】



【 図 1 8 】



【 圖 1 9 】



【 図 2 0 】

[illegible]

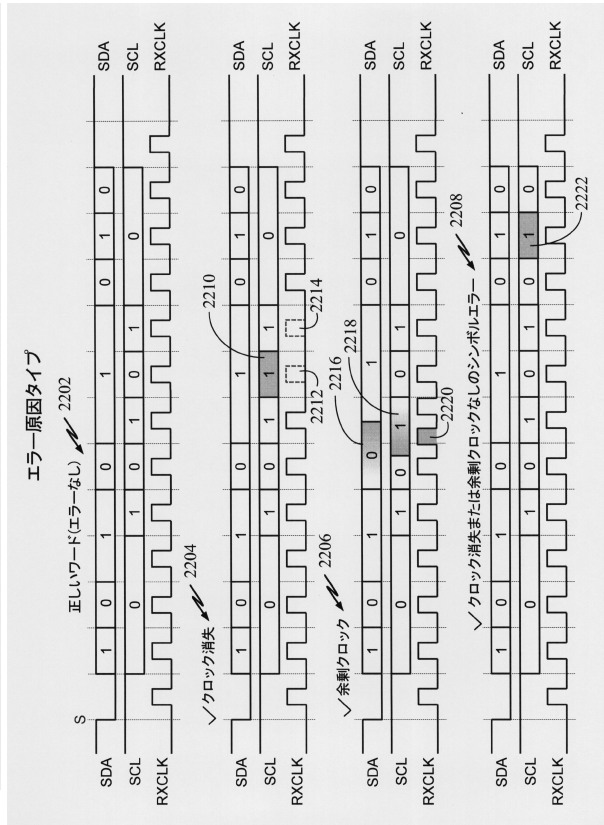


【 図 2 1 】

OC1eビット[9マッピング続き

3連ビット[9:0]				アドレス	書き込み	読取り	ビット[0]																			
				アドレス	書き込み	読取り	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2222_2222_2222 <sub>3</sub>	0x81BF0					禁止	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
2222_2222_2221 <sub>3</sub>	0x81BEF																									
2222_2222_2220 <sub>3</sub>	0x2A (42) ↓																									
2222_2222_1102 <sub>3</sub>	0x81B08																									
2222_2222_1101 <sub>3</sub>	0x81B05						1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1
2222_2222_1100 <sub>3</sub>	0x81BC4																									
2222_2222_1000 <sub>3</sub>	0x2A (42) ↓																									
2222_2222_0222 <sub>3</sub>	0x81BBA				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1
2222_2222_0221 <sub>3</sub>	0x81B89				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1
2222_2222_0220 <sub>3</sub>	0x81B88				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
2222_2222_0212 <sub>3</sub>	0x81B87				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
2222_2222_0211 <sub>3</sub>	0x81B86				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
2222_2222_0210 <sub>3</sub>	0x81B85				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
2222_2222_0200 <sub>3</sub>	0x81B80				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
2222_2222_0100 <sub>3</sub>	0x81BAF				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
2222_2222_0000 <sub>3</sub>	0x81BA0				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
2222_2221_2222 <sub>2</sub>	0x81B9F					禁止																				
2222_2221_2102 <sub>2</sub>	0x81B90																									
2222_2221_2101 <sub>2</sub>	0x81B8F	禁止	SIDスキャン応答	禁止			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0
2222_2222_2100 <sub>2</sub>	0x81B8E																									
2222_2222_2000 <sub>2</sub>	0x4 (10) ↓																									
2222_2221_2200 <sub>1</sub>	0x81B85				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0
2222_2221_1222 <sub>1</sub>	0x81B84				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0
2222_2221_1221 <sub>1</sub>	0x81B83				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0
2222_2221_1211 <sub>1</sub>	0x81B80				予約済み	予約済み	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0

【 図 2 2 】



【 図 2 3 】

シンボルエラー調査

CCleワード=0000\_0000\_0000\_00000000上で、  
単一シンボルエラーが発生するとき

エラー箇所	S11_S0	シンボル	T11_10 (遷移)	Hex	ビット[19:0]	ビット[2:0]
Good	0321_0321	0321	0000_0000	00000	0000_0000_0000_0000	000
T00*1	0321_0321	0320	0000_0000	00002	0000_0000_0000_0010	010
T00*2	0321_0321	0323	0000_0000	00001	0000_0000_0000_0001	001
T01*2	0321_0321	0301	0000_0000	00111	0000_0000_0000_0100	100
T02*2	0321_0321	0121	0000_0000	01110	0000_0000_0000_1100	100
T03*2	0321_0321	2321	0000_0000	1100	0000_0000_0010_0100	100
T04*2	0321_0323	0321	0000_0001	1000	0000_0000_0000_0110	100
T05*2	0321_0301	0321	0000_0110	0000	0000_0000_1_0100_0100	100
T06*2	0321_0121	0321	0000_0110	0000	0000_0000_0011_1100_0100	100
T07*2	0321_2321	0321	0000_1100	0000	0000_0000_1011_0110_0100	100
T08*2	0323_0321	0321	0001_1000	0000	0000_0010_0010_0010_1100	100
T09*2	0301_0321	0321	0011_0000	0000	0000_0110_0110_1000_0100	100
T10*2	0121_0321	0321	0110_0000	0000	0001_0011_0011_1000_1100	100
T11*2	2321_0321	0321	1100_0000	0000	0011_1001_1010_1010_0100	100

2300

2306

2304

2302

2308

3桁のLSBがすべての  
単一シンボルエラーを  
検出

シンボルエラーが高次であるほど  
パースエラーが多く発生

クロック消失が発生させない  
単一線エラー

【 図 2 4 】

例示的なクロック消失なしのシンボルエラー（シンボルスリップなし）

$\text{001eワード} = \text{1111}_1\text{111}_2\text{111}_3(\text{0x40DF8})$ 上で、  
単一シンボルエラーが発生するととき

エリア・箇所	S11_160 (シンボル)	T11...T0 (遷移)	Hex	ビット[19:0]	ビット[2:0]
<b>Good</b>	2301_2301_2301	1111_1111_1111	40df8	0100_0000_1101_1111_1000	000
T00*2	2301_2301_2303	1111_1111_1110	40df7	0100_0000_1101_1111_0111	111
T00*3	2301_2301_2302	1111_1111_1112	40df9	0100_0000_1101_1111_1001	001
T01*2	2301_2301_2321	1111_1111_1100	40df4	0100_0000_1101_1111_0100	100
T02*2	2301_2301_2101	1111_1111_1001	40dec	0100_0000_1101_1110_1100	100
T03*2	2301_2301_0301	1111_1111_0011	40dd4	0100_0000_1101_1101_0100	100
T04*2	2301_2301_2301	1111_1110_0111	40dbc	0100_0000_1101_1000_1100	100
T05*2	2301_2321_2301	1111_1100_1111	40cb4	0100_0000_1100_1011_0100	100
T06*2	2301_2101_2301	1111_1100_1111	40azc	0100_0000_1010_0010_1100	100
T07*2	2301_0301_2301	1111_0011_1111	40294	0100_0000_0010_1001_0100	100
T08*2	2303_2301_2301	1100_0111_1111	3ebcc	0011_1110_1011_1100_1000	100
T09*2	2371_2301_2301	1100_0111_1111	3a774	0011_1110_0111_0111_0100	100
T10*2	2101_2301_2301	1001_1111_1111	2da6c	0010_1101_1010_0110_1100	100
T11*2	0301_2301_2301	0011_1111_1111	07354	0000_0111_0011_0101_0100	100

【図 25】

例示的なクロック消失なしのシンボルエラー（シンボルスリップなし）  
C0レワード=2222\_2222\_2222<sub>3</sub> (0x8BF0) 上で、  
クロック消失なしの単一シンボルエラーが発生するとき

エラー箇所	S11_S0 (シンボル)	T11_T0 (遷移)	Hex	ビット[19:0]	ビット[2:0]
Good	3131_3131_3131	2222_2222_2222	81b0	1000_0001_1011_1110_0000	000
T00*1	3131_3131_3130	2222_2222_2221	81bf	1000_0001_1011_1110_1111	111
T00*3	3131_3131_3132	2222_2222_2220	81be	1000_0001_1011_1110_1110	110
T01*1	3131_3131_3131	2222_2222_2210	81bb	1000_0001_1011_1110_1011	011
T01*3	3131_3131_3101	2222_2222_2102	81be9	1000_0001_1011_1110_1001	001
T02*1	3131_3131_3031	2222_2222_2102	81be1	1000_0001_1011_1110_0001	001
T02*3	3131_3131_3231	2222_2222_2012	81bdb	1000_0001_1011_1101_1011	011
T03*1	3131_3131_2131	2222_2222_1022	81bc3	1000_0001_1011_1100_0011	011
T03*3	3131_3131_3131	2222_2222_0122	81bb1	1000_0001_1011_1101_0001	001
T04*1	3131_3130_3131	2222_2221_0222	81b69	1000_0001_1011_0110_1001	001
T04*3	3131_3132_3131	2222_2220_1222	81b33	1000_0001_1011_0011_0011	011
T05*1	3131_3131_3131	2222_2210_2222	81a5b	1000_0001_1010_0101_1011	011
T05*3	3131_3101_3131	2222_2201_2222	81969	1000_0001_1001_1011_1001	001
T06*1	3131_3031_3131	2222_2102_2222	81731	1000_0001_0111_0011_0001	001
T06*3	3131_3231_3131	2222_2022_2222	8154b	1000_0001_0101_0100_1011	011
T07*1	3131_2131_3131	2222_1022_2222	80db3	1000_0000_1101_1011_0011	011
T07*3	3131_0131_3131	2222_0122_2222	80801	1000_0000_1000_0000_0001	001
T08*1	3130_3131_3131	2221_0222_2222	7f139	0111_1111_0001_0011_1001	001
T08*3	3132_3131_3131	2220_1222_2222	7e023	0111_1110_0000_0010_0011	011
T09*1	3121_3131_3131	2210_2222_2222	796cb	0111_1001_1011_1100_1011	011
T09*3	3101_3131_3131	2201_2222_2222	76889	0111_0110_1000_1000_1001	001
T10*1	3031_3131_3131	2102_2222_2222	69681	0110_1001_1011_1000_0001	001
T10*3	3231_3131_3131	2012_2222_2222	601bb	0110_0000_0001_1011_1011	011
T11*1	2131_3131_3131	1022_2222_2222	39aa3	0011_1001_1010_1010_0011	011
T11*3	0131_3131_3131	0122_2222_2222	1cd51	0001_1100_1101_0101_0001	001

【図 26】

例示的なクロック消失なしのシンボルエラー（シンボルスリップなし）  
C0レワード=0120\_1201\_2012<sub>3</sub> (0x18F38) 上で、  
単一シンボルエラーが発生するとき

エラー箇所	S11_S0 (シンボル)	T11_T0 (遷移)	Hex	ビット[19:0]	ビット[2:0]
Good	0132_3101_3231	0120_1201_2012	18f38	0001_1000_1111_0011_1000	000
T00*1	0132_3101_3230	0120_1201_2011	18f37	0001_1000_1111_0011_0111	111
T00*3	0132_3101_3232	0120_1201_2010	18f36	0001_1000_1111_0011_0110	110
T01*1	0132_3101_3201	0120_1201_2021	18f3a	0001_1000_1111_0011_1010	010
T02*1	0132_3101_3031	0120_1201_2102	18f3e	0001_1000_1111_0011_1110	110
T02*3	0132_3101_3131	0120_1201_2222	18f4d	0001_1000_1111_0100_1101	101
T03*1	0132_3101_0231	0120_1201_0212	18f14	0001_1000_1111_0001_0100	100
T04*1	0132_3102_3231	0120_1202_1012	18f6e	0001_1000_1111_0110_1110	110
T05*2	0132_3121_3231	0120_1210_2012	18fda	0001_1000_1111_1101_1010	010
T05*3	0132_3131_3231	0120_1222_2012	1916f	0001_1001_0001_0110_1111	111
T06*1	0132_3201_3231	0120_1021_2012	18b6c	0001_1000_1011_0110_1100	100
T07*3	0132_0101_3231	0120_2101_2012	194ea	0001_1001_0100_1110_1010	010
T08*2	0130_3101_3231	0121_0201_2012	1a04e	0001_1010_0000_0100_1110	110
T08*3	0131_3101_3231	0122_2201_2012	1cd05	0001_1100_1011_0000_0101	101
T09*3	0102_3101_3231	0102_1201_2012	128b4	0001_0010_1000_1011_0100	100
T10*3	0232_3101_3231	0210_1201_2012	228fe	0010_0010_1000_1111_1110	110
T11*2	2132_3101_3231	0201_1201_2012	35c8a	0011_0101_1100_1000_1010	010
T11*3	3132_3101_3231	2220_1201_2012	7add7	0111_1101_1101_1101_0111	111

【図 27】

例示的なクロック消失なしのシンボルエラー（シンボルスリップなし）  
C0レワード=1201\_2012\_0120<sub>3</sub> (0x4DA8) 上で、  
単一シンボルエラーが発生するとき

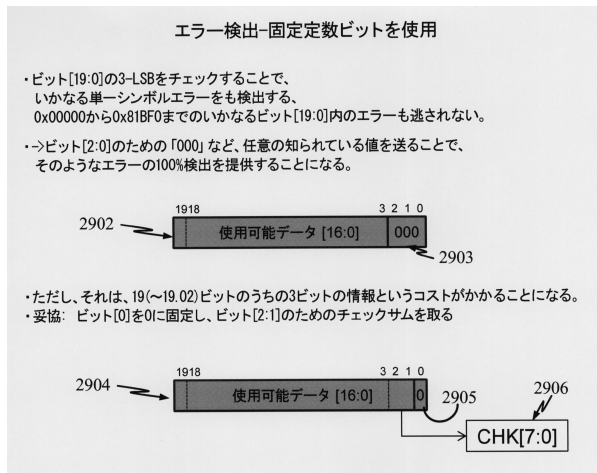
エラー箇所	S11_S0 (シンボル)	T11_T0 (遷移)	Hex	ビット[19:0]	ビット[2:0]
Good	2030_2120_3021	1201_2012_0120	4ada8	0100_1010_1101_1010_1000	000
T00*1	2030_2120_3020	1201_2012_0122	4adaa	0100_1010_1101_1010_1010	010
T00*2	2030_2120_3023	1201_2012_0121	4ada9	0100_1010_1101_1010_1001	001
T01*1	2030_2120_3031	1201_2012_0102	4ada4	0100_1010_1101_1010_0100	100
T02*1	2030_2120_3121	1201_2012_0210	4adae	0100_1010_1101_1010_1110	110
T03*1	2030_2120_2021	1201_2012_2220	4ade7	0100_1010_1101_1110_0111	111
T03*2	2030_2120_1021	1201_2012_1020	4adba	0100_1010_1101_1011_1010	010
T04*1	2030_2121_3021	1201_2010_2120	4ad3c	0100_1010_1101_0011_1100	100
T05*1	2030_2130_3021	1201_2021_0120	4ae4a	0100_1010_1110_0100_1010	010
T06*1	2030_2020_3021	1201_2222_0120	4b44d	0100_1011_0100_0100_1101	101
T06*2	2030_2320_3021	1201_2102_0120	4af8e	0100_1010_1111_1000_1110	110
T07*1	2030_3120_3021	1201_0212_0120	4a244	0100_1010_0010_0100_0100	100
T08*1	2031_2120_3021	1202_1012_0120	4bebe	0100_1011_1110_1011_1110	110
T08*2	2020_2120_3021	1222_2012_0120	5610f	0101_0110_0001_0000_1111	111
T09*2	2010_2120_3021	1210_2012_0120	4e0ea	0100_1110_0000_1110_1010	010
T10*1	2130_2120_3021	1021_2012_0120	37a1c	0011_0111_1010_0001_1100	100
T11*1	3030_2120_3021	2101_2012_0120	67afa	0110_0111_1010_1111_1010	010
T11*2	0030_2120_3021	0001_2012_0120	02c5b	0000_0010_1100_0101_1011	011

【図 28】

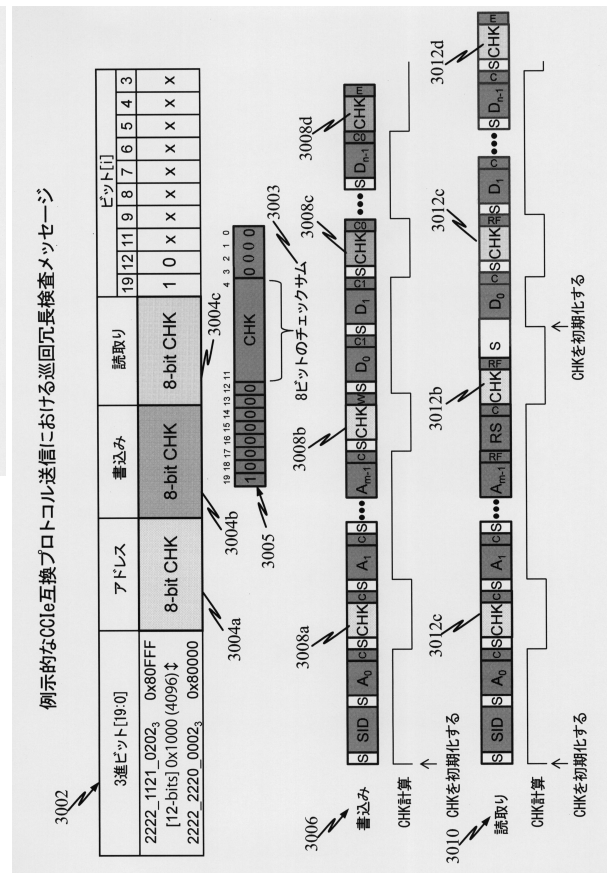
例示的なクロック消失なしのシンボルエラー（シンボルスリップなし）  
C0レワード=2012\_0120\_1201<sub>3</sub> (0x5ED08) 上で、  
単一シンボルエラーが発生するとき

エラー箇所	S11_S0 (シンボル)	T11_T0 (遷移)	Hex	ビット[19:0]	ビット[2:0]
Good	3231_0132_3101	2012_0120_1201	5ed08	0101_1110_1101_0000_1000	000
T00*2	3231_0132_3103	2012_0120_1200	5ed07	0101_1110_1101_0000_0111	111
T00*3	3231_0132_3102	2012_0120_1202	5ed09	0101_1110_1101_0000_1001	001
T01*2	3231_0132_3121	2012_0120_1210	5ed0a	0101_1110_1101_0000_1010	010
T01*3	3231_0132_3131	2012_0120_1222	5ed0f	0101_1110_1101_0000_1111	111
T02*3	3231_0132_3201	2012_0120_1021	5edcf	0101_1110_1100_1111_1100	100
T03*3	3231_0132_0101	2012_0120_2101	5ed1a	0101_1110_1101_0001_1010	010
T04*2	3231_0130_3101	2012_0121_0201	5ed3e	0101_1110_1101_0011_1110	110
T04*3	3231_0131_3101	2012_0122_2201	5edc5	0101_1110_1101_1100_0101	101
T05*3	3231_0102_3101	2012_0102_1201	5ebc4	0101_1110_1011_1100_0100	100
T06*3	3231_0232_3101	2012_0210_1201	5eeee	0101_1110_1110_1110_1110	110
T07*2	3231_2132_3101	2012_1020_1201	5f2ba	0101_1111_0010_1011_1010	010
T07*3	3231_3132_3101	2012_2220_1201	600f7	0110_0000_0000_1111_0111	111
T08*3	3232_0132_3101	2010_2120_1201	5cadc	0101_1100_1010_1101_1100	100
T09*3	3201_0132_3101	2012_0120_1201	6204a	0110_0010_0000_0100_1010	010
T10*2	3031_0132_3101	2102_0120_1201	686ce	0110_1000_0110_1100_1110	110
T10*3	3131_0132_3101	2222_0120_1201	8073d	1000_0000_0111_0011_1101	101
T11*3	0231_0132_3101	0212_0120_1201	25264	0010_0101_0010_0110_0100	100

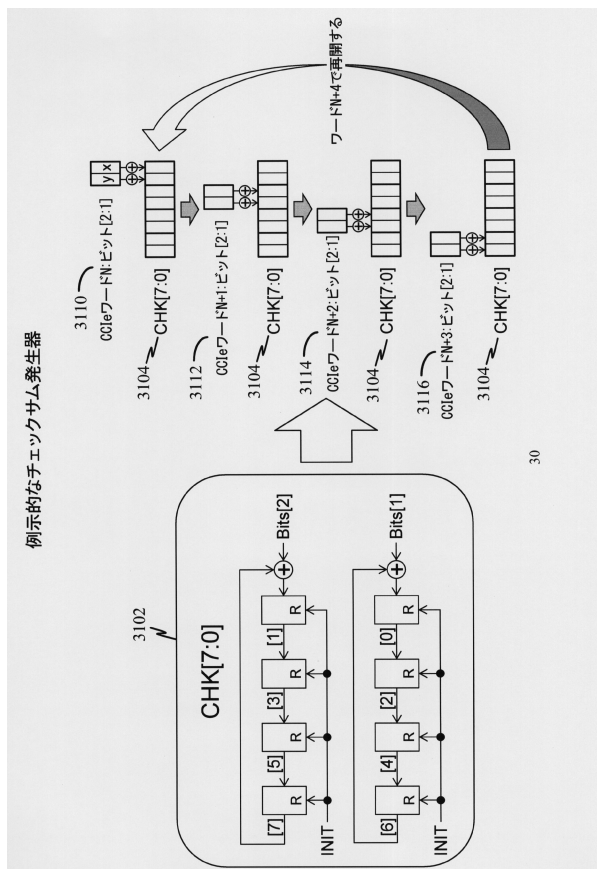
【図 29】



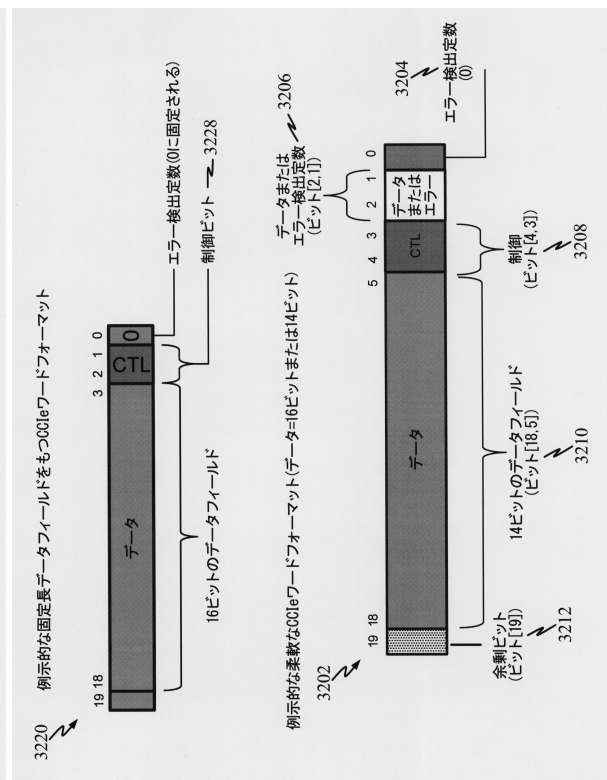
【図 30】



【図 31】

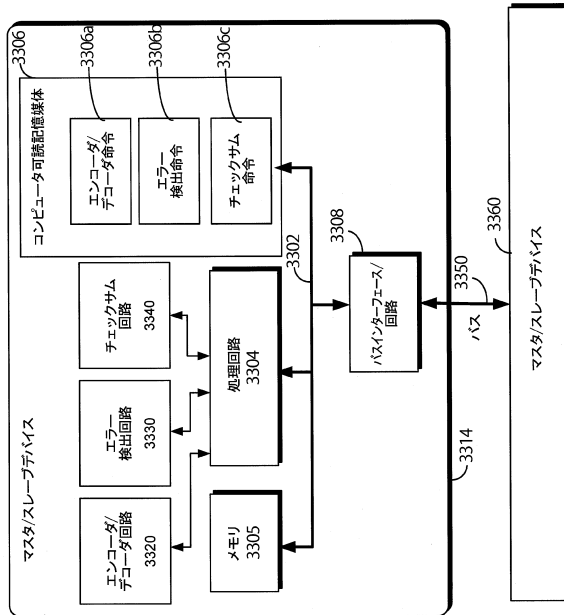


【図 32】

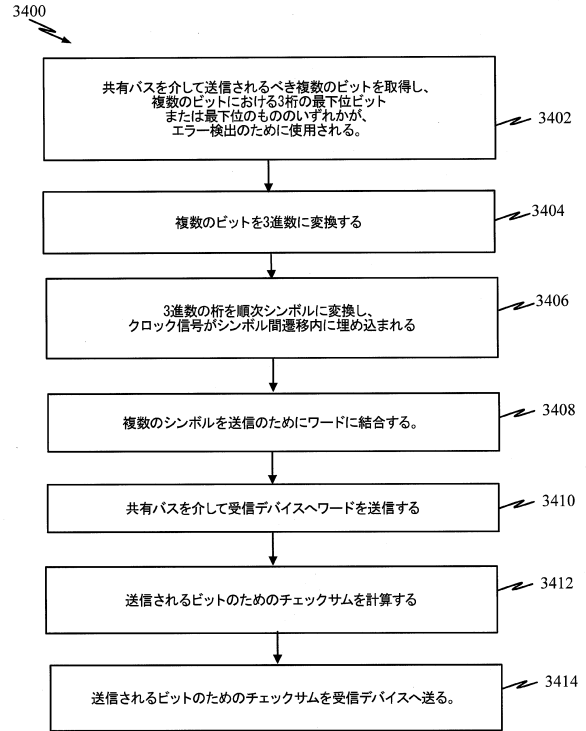




【 図 3 3 】

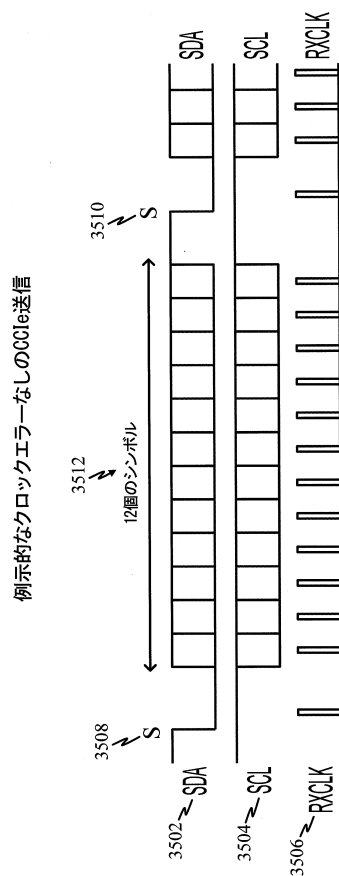


【 図 3 4 】

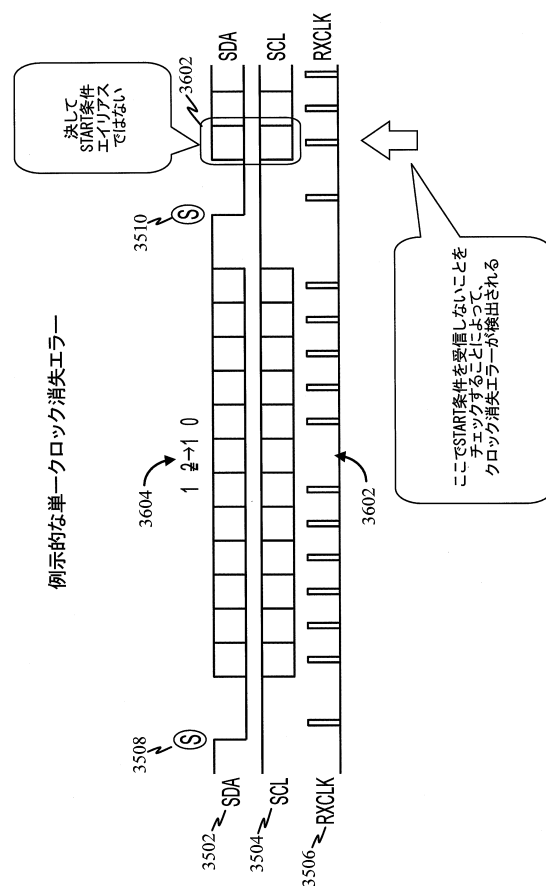


## エラー検出のための方法

【 図 3 5 】

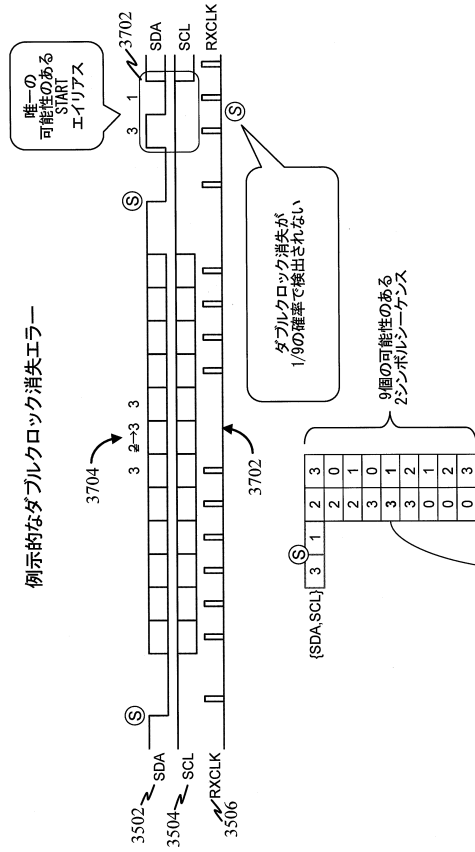


【 図 3 6 】



【図 37】

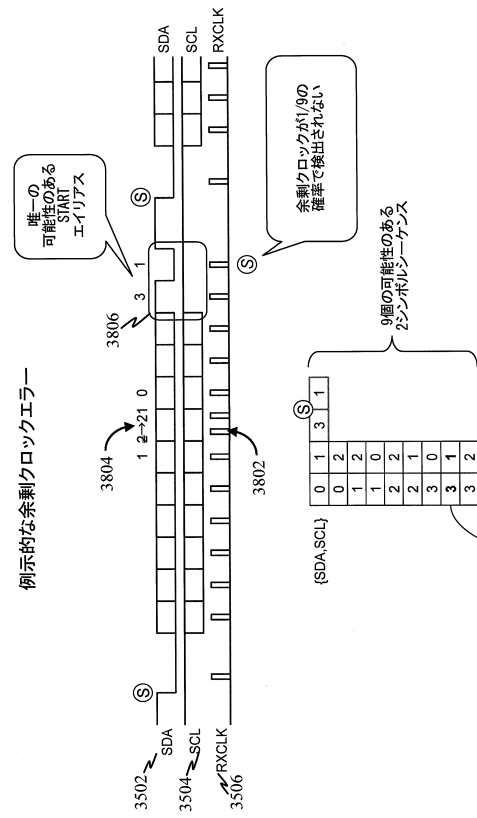
例示的なダブルクロック消失エラー



次のワードにおけるエラー検出に依拠する必要がある

【図 38】

例示的な余剰クロックエラー



現在または次のワードのエラー検出定数を用いたエラー検出に依拠する必要がある

---

フロントページの続き

(56)参考文献 特開 2 0 0 5 - 2 1 0 1 5 9 ( J P , A )  
特開昭 5 0 - 1 4 7 2 0 4 ( J P , A )  
特開平 1 0 - 0 3 2 5 6 5 ( J P , A )  
特開平 0 4 - 3 5 8 3 6 3 ( J P , A )  
特開 2 0 1 2 - 0 3 9 5 5 2 ( J P , A )  
米国特許第 0 4 3 9 8 2 6 5 ( U S , A )  
特開昭 5 9 - 1 7 8 8 4 2 ( J P , A )  
特開昭 5 0 - 0 3 7 3 0 5 ( J P , A )

(58)調査した分野(Int.Cl. , D B 名)  
H 0 4 L 1 / 0 0  
H 0 4 L 2 5 / 0 2