

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2017-511044
(P2017-511044A)

(43) 公表日 平成29年4月13日(2017.4.13)

(51) Int. Cl.	F I	テーマコード (参考)
HO4L 1/00 (2006.01)	HO4L 1/00 A	5J065
HO3M 13/09 (2006.01)	HO3M 13/09	5K014

審査請求 未請求 予備審査請求 有 (全 46 頁)

(21) 出願番号 特願2016-554356 (P2016-554356)
 (86) (22) 出願日 平成27年2月28日 (2015. 2. 28)
 (85) 翻訳文提出日 平成28年8月26日 (2016. 8. 26)
 (86) 国際出願番号 PCT/US2015/018202
 (87) 国際公開番号 W02015/131164
 (87) 国際公開日 平成27年9月3日 (2015. 9. 3)
 (31) 優先権主張番号 61/946, 647
 (32) 優先日 平成26年2月28日 (2014. 2. 28)
 (33) 優先権主張国 米国 (US)
 (31) 優先権主張番号 14/634, 106
 (32) 優先日 平成27年2月27日 (2015. 2. 27)
 (33) 優先権主張国 米国 (US)

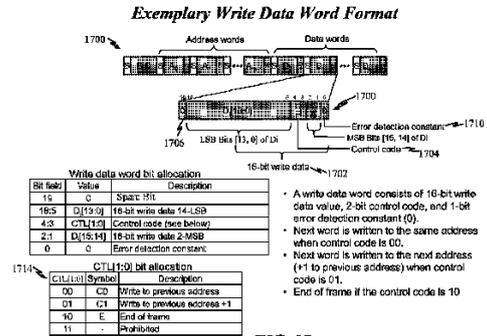
(71) 出願人 507364838
 クアルコム, インコーポレイテッド
 アメリカ合衆国 カリフォルニア 921
 21 サン ディエゴ モアハウス ドラ
 イブ 5775
 (74) 代理人 100108453
 弁理士 村山 靖彦
 (74) 代理人 100163522
 弁理士 黒田 晋平
 (72) 発明者 仙石 祥一郎
 アメリカ合衆国・カリフォルニア・921
 21-1714・サン・ディエゴ・モアハ
 ウス・ドライブ・5775
 Fターム(参考) 5J065 AE01 AH02 AH06 AH07
 5K014 AA01 BA01

最終頁に続く

(54) 【発明の名称】 エラー検出最適化を容易にするための共有バスを介したビット割振り

(57) 【要約】

共有バスを介したエラー検出最適化を容易にすることに向けられた様々な態様を開示する。マスタデバイスがスレーブデバイスに結合され、ワードの符号化通信が、制御データバスを介してマスタデバイスとスレーブデバイスとの間で容易にされる。符号化通信は、エラー検出定数を最大化することを容易にするために、符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化される。プロトコルは、ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、複数の最下位ビットを割り振る。



【特許請求の範囲】**【請求項1】**

制御データバスに結合されたプロセッサを備え、

前記プロセッサが、前記制御データバスを介したマスタデバイスとスレーブデバイスとの間のワードの符号化通信を容易にするように構成され、

前記符号化通信が、エラー検出定数を最大化することを容易にするために、前記符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化され、前記プロトコルが、前記ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、前記複数の最下位ビットを割り振る、デバイス。

10

【請求項2】

前記制御データバスが2ラインバスである、請求項1に記載のデバイス。

【請求項3】

前記プロトコルがカメラ制御インターフェース拡張(CCIe)プロトコルである、請求項1に記載のデバイス。

【請求項4】

前記デバイスが、

ビット割り振り方式に従ってビットを割り振るよう構成されたビット割り振り回路であって、前記ビット割り振り方式が、前記符号化通信の前記複数の最下位ビットを割り振る、ビット割り振り回路と、

20

ワードの符号化を容易にするよう構成されたエンコーダ回路であって、

前記プロトコルに関連付けられた前記ワードのワードフォーマットを決定するよう構成されたプロトコルサブ回路と、

前記符号化通信を生成するために、前記ワードフォーマットと前記ビット割り振り方式とに従って、ワードを符号化するよう構成された符号化サブ回路と

を備えるエンコーダ回路と、

前記制御データバスを介して前記符号化通信を送信するよう構成された通信回路とをさらに備える、請求項1に記載のデバイス。

【請求項5】

前記符号化サブ回路が、シンボルにトランスコーディングされた、符号化された3進数として、ワードを符号化するよう構成される、請求項4に記載のデバイス。

30

【請求項6】

前記エンコーダ回路が、前記ワードフォーマットと前記ビット割り振り方式とを介して実施するための最適化を確認するよう構成された最適化サブ回路をさらに備える、請求項4に記載のデバイス。

【請求項7】

前記最適化サブ回路が、第1のビット割り振り方式を有するエラー検出最適化に従うワードの符号化と、第2のビット割り振り方式を有するデータ最適化に従うワードの符号化との間で切り替えることを容易にするよう構成される、請求項6に記載のデバイス。

【請求項8】

前記符号化サブ回路が、前記複数の最下位ビットが固定数の3ビットを備えるデータ最適化に従って、ワードを符号化するよう構成され、前記ビット割り振り回路が、最下位ビットをエラー検出のために、第2の最下位ビットを前記ワードの前記データ部分の前記第1の最上位ビットのために、および第3の最下位ビットを前記ワードの前記データ部分の第2の最上位ビットのために割り振ることによって、前記データ最適化を容易にするよう構成される、請求項6に記載のデバイス。

40

【請求項9】

前記符号化サブ回路が、前記複数の最下位ビットが固定数の3ビットを備えるエラー検出最適化に従って、ワードを符号化するよう構成され、前記ビット割り振り回路が、最下位ビット、第2の最下位ビット、および第3の最下位ビットの各々をエラー検出のために割

50

り振ることによって、前記エラー検出最適化を容易にするように構成される、請求項6に記載のデバイス。

【請求項10】

前記制御データバスを介して前記符号化通信を受信するように構成された通信回路と、前記符号化通信の復号を容易にするように構成されたデコーダ回路とをさらに備える、請求項1に記載のデバイス。

【請求項11】

前記デコーダ回路が、前記プロトコルに関連付けられた前記ワードのワードフォーマットを検出するように構成されたプロトコルサブ回路と、前記符号化通信の最適化と、前記最適化に対応するビット割振り方式とを確認するように構成された最適化サブ回路と、前記ワードフォーマットと前記ビット割振り方式とに従って、前記符号化通信を復号するように構成された復号サブ回路とを備える、請求項10に記載のデバイス。

10

【請求項12】

マスタデバイスをスレーブデバイスに結合するステップと、制御データバスを介した前記マスタデバイスと前記スレーブデバイスとの間のワードの符号化通信を容易にするステップであって、前記符号化通信が、エラー検出定数を最大化することを容易にするために、前記符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化され、前記プロトコルが、前記ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、前記複数の最下位ビットを割り振るステップとを含む方法。

20

【請求項13】

前記制御データバスが2ラインバスである、請求項12に記載の方法。

【請求項14】

前記プロトコルがカメラ制御インターフェース拡張(CCIe)プロトコルである、請求項12に記載の方法。

【請求項15】

前記プロトコルに関連付けられた前記ワードのワードフォーマットを決定するステップと、

30

ビット割振り方式に従ってビットを割り振るステップであって、前記ビット割振り方式が、前記符号化通信の前記複数の最下位ビットを割り振るステップと、

前記符号化通信を生成するために、前記ワードフォーマットと前記ビット割振り方式とに従って、前記ワードを符号化するステップと、

前記制御データバスを介して前記符号化通信を送信するステップと

をさらに含む、請求項12に記載の方法。

【請求項16】

前記符号化するステップが、シンボルにトランスコーディングされた、符号化された3進数として、ワードを符号化するステップを含む、請求項15に記載の方法。

40

【請求項17】

前記ワードフォーマットと前記ビット割振り方式とを介して実施するための最適化を確認するステップをさらに備える、請求項15に記載の方法。

【請求項18】

第1のビット割振り方式を有するエラー検出最適化に従うワードの符号化と、第2のビット割振り方式を有するデータ最適化に従うワードの符号化との間で切り替えるステップをさらに含む、請求項17に記載の方法。

【請求項19】

前記符号化するステップが、前記複数の最下位ビットが固定数の3ビットを備えるデー

50

タ最適化に従って、ワードを符号化するステップを含み、前記割り振るステップが、最下位ビットをエラー検出のために、第2の最下位ビットを前記ワードの前記データ部分の前記第1の最上位ビットのために、および第3の最下位ビットを前記ワードの前記データ部分の第2の最上位ビットのために割り振ることによって、前記データ最適化を容易にするステップを含む、請求項17に記載の方法。

【請求項20】

前記符号化するステップが、前記複数の最下位ビットが固定数の3ビットを備えるエラー検出最適化に従って、ワードを符号化するステップを含み、前記割り振るステップが、最下位ビット、第2の最下位ビット、および第3の最下位ビットの各々をエラー検出のために割り振ることによって、前記エラー検出最適化を容易にするステップを含む、請求項17に記載の方法。

10

【請求項21】

前記制御データベースを介して前記符号化通信を受信するステップと、
前記符号化通信を復号するステップと
をさらに含む、請求項12に記載の方法。

【請求項22】

前記プロトコルに関連付けられた前記ワードのワードフォーマットを検出するステップと、
前記符号化通信の最適化と、前記最適化に対応するビット割り振り方式とを確認するステップと、
前記ワードフォーマットと前記ビット割り振り方式とに従って、前記符号化通信を復号するステップと
をさらに含む、請求項21に記載の方法。

20

【請求項23】

マスタデバイスをスレーブデバイスに結合するための手段と、
制御データベースを介した前記マスタデバイスと前記スレーブデバイスとの間のワードの符号化通信を容易にするための手段であって、前記符号化通信が、エラー検出定数を最大化することを容易にするために、前記符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化され、前記プロトコルが、前記ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、前記複数の最下位ビットを割り振る手段と
を備えるデバイス。

30

【請求項24】

前記プロトコルに関連付けられた前記ワードのワードフォーマットを決定するための手段と、
ビット割り振り方式に従ってビットを割り振るための手段であって、前記ビット割り振り方式が、前記複数の最下位ビットを割り振る手段と、
前記符号化通信を生成するために、前記ワードフォーマットと前記ビット割り振り方式とに従って、前記ワードを符号化するための手段と、
前記制御データベースを介して前記符号化通信を送信するための手段と
をさらに備える、請求項23に記載のデバイス。

40

【請求項25】

前記ワードフォーマットと前記ビット割り振り方式とを介して実施するための最適化を確認するための手段をさらに備える、請求項24に記載のデバイス。

【請求項26】

第1のビット割り振り方式を有するエラー検出最適化に従うワードの符号化と、第2のビット割り振り方式を有するデータ最適化に従うワードの符号化との間で切り替えるための手段をさらに備える、請求項25に記載のデバイス。

【請求項27】

1つまたは複数の命令を記憶している非一時的機械可読記憶媒体であって、前記1つまた

50

は複数の命令が、少なくとも1つのプロセッサによって実行されたときに、前記少なくとも1つのプロセッサに、

マスタデバイスをスレーブデバイスに結合すること、および

制御データバスを介した前記マスタデバイスと前記スレーブデバイスとの間のワードの符号化通信を容易にすることであって、前記符号化通信が、エラー検出定数を最大化することを容易にするために、前記符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化され、前記プロトコルが、前記ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、前記複数の最下位ビットを割り振ること

を行わせる、非一時的機械可読記憶媒体。

10

【請求項 28】

前記1つまたは複数の命令が、前記少なくとも1つのプロセッサによって実行されたときに、前記少なくとも1つのプロセッサに、

前記プロトコルに関連付けられた前記ワードのワードフォーマットを決定すること、

ビット割り振り方式に従ってビットを割り振ることであって、前記ビット割り振り方式が、前記複数の最下位ビットを割り振ること、

前記符号化通信を生成するために、前記ワードフォーマットと前記ビット割り振り方式とに従って、前記ワードを符号化すること、および

前記制御データバスを介して前記符号化通信を送信すること

を行わせる命令をさらに備える、請求項27に記載の非一時的機械可読記憶媒体。

20

【請求項 29】

前記1つまたは複数の命令が、前記少なくとも1つのプロセッサによって実行されたときに、前記少なくとも1つのプロセッサに、

前記複数の最下位ビットが固定数の3ビットを備えるデータ最適化に従って、ワードを符号化すること、ならびに

最下位ビットをエラー検出のために、第2の最下位ビットを前記ワードの前記データ部分の前記第1の最上位ビットのために、および第3の最下位ビットを前記ワードの前記データ部分の第2の最上位ビットのために割り振ることによって、前記データ最適化を容易にすること

を行わせる命令をさらに備える、請求項28に記載の非一時的機械可読記憶媒体。

30

【請求項 30】

前記1つまたは複数の命令が、前記少なくとも1つのプロセッサによって実行されたときに、前記少なくとも1つのプロセッサに、

前記複数の最下位ビットが固定数の3ビットを備えるエラー検出最適化に従って、ワードを符号化すること、ならびに

最下位ビット、第2の最下位ビット、および第3の最下位ビットの各々をエラー検出のために割り振ることによって、前記エラー検出最適化を容易にすること

を行わせる命令をさらに備える、請求項28に記載の非一時的機械可読記憶媒体。

【発明の詳細な説明】

【技術分野】

40

【0001】

関連出願の相互参照

本出願は、その内容全体が参照により本明細書に組み込まれる、2014年2月28日に出願した米国仮特許出願第61/946,647号、および2015年2月27日に出願した米国非仮特許出願第14/634,106号の優先権および利益を主張する。

【0002】

本開示は、共有バスを介した効率的な動作を可能にすることに関し、より詳細には、共有バスを介したエラー検出最適化を容易にするために、所望のワードフォーマットに従ってビットを割り振ることに関する。

【背景技術】

50

【0003】

一般に、共有バスは、複数のデバイスを結合するときに使用され得る。たとえば、Inter-Integrated Circuit(I²C、I²Cとも呼ばれる)は、マザーボード、埋め込みシステム、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、シリアルクロックライン(SCL)と、7ビットアドレス指定を用いるシリアルデータライン(SDA)とを含む。I²Cバスは、ノードに関する2つの役割、すなわち、マスタおよびスレーブを有する。マスタノードは、クロックを生成し、スレーブノードとの通信を開始するノードである。スレーブノードは、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ノードである。I²Cバスは、任意の数のマスタノード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、イメージセンサから画像を取り込み、対応する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イメージ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ノード)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。別の例では、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】

エラー検出アルゴリズムは、共有バス通信の精度を改善するために実装されることが多い。しかしながら、そのようなエラーは、従来のエラー検出アルゴリズムによって検出されないことが多い。したがって、共有バス上で通信されたエラーがより正確に検出されるアルゴリズムを実装することが望ましい。

【課題を解決するための手段】

30

【0006】

以下では、本開示の1つまたは複数の態様の基本的な理解をもたらすために、そのような態様の簡略化された概要を提示する。本概要は、本開示のすべての企図される特徴の広い概説ではなく、本開示のすべての態様の鍵となる要素または不可欠な要素を特定することも、本開示の任意またはすべての態様の範囲を正確に説明することも意図していない。その唯一の目的は、後で提示されるより詳細な説明の前置きとして、簡略化された形態で本開示の1つまたは複数の態様のいくつかの概念を提示することである。

【0007】

本開示の態様は、共有バスを介したエラー検出最適化を容易にすることに向けられた方法、装置、コンピュータプログラム製品、および処理システムを提供する。一態様では、本開示は、マスタデバイスをスレーブデバイスに結合するステップと、制御データバスを介したマスタデバイスとスレーブデバイスとの間のワードの符号化通信を容易にするステップとを含む、方法を提供する。この特定の実装形態では、符号化通信は、エラー検出定数を最大化することを容易にするために、符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化される。ここで、そのような最大化は、ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、複数の最下位ビットを割り振るプロトコルを介して達成される。

40

【0008】

別の態様では、共有バスを介したエラー検出最適化を容易にするように構成されたデバイスを開示する。デバイスは、制御データバスに結合されたプロセッサを備える。ここで

50

、プロセッサは、制御データバスを介したマスタデバイスとスレーブデバイスとの間のワードの符号化通信を容易にするように構成される。この実装形態における符号化通信は、エラー検出定数を最大化することを容易にするために、符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化される。特に、プロトコルは、ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、複数の最下位ビットを割り振る。

【0009】

さらなる態様では、共有バスを介したエラー検出最適化を容易にするように構成された別のデバイスを開示する。この実装形態では、デバイスは、マスタデバイスをスレーブデバイスに結合するための手段と、制御データバスを介したマスタデバイスとスレーブデバイスとの間のワードの符号化通信を容易にするための手段とを備える。ここで、符号化通信は、エラー検出定数を最大化することを容易にするために、符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化される。すなわち、プロトコルは、ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、複数の最下位ビットを割り振る。

10

【0010】

また別の態様では、記憶された1つまたは複数の命令を介して、共有バスを介したエラー検出最適化を容易にするように構成された非一時的機械可読記憶媒体を開示する。ここで、少なくとも1つのプロセッサによって実行されたときに、1つまたは複数の命令は、少なくとも1つのプロセッサに、マスタデバイスをスレーブデバイスに結合すること、および、制御データバスを介したマスタデバイスとスレーブデバイスとの間のワードの符号化通信を容易にするを行わせる。この実装形態では、符号化通信は、エラー検出定数を最大化することを容易にするために、符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化される。特に、プロトコルは、ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、複数の最下位ビットを割り振る。

20

【0011】

これらの開示する態様および他の開示する態様は、以下の詳細な説明を検討すれば、より十分に理解されるであろう。添付図面とともに本発明の特定の例示的な態様の以下の説明を検討すれば、本発明の他の態様、特徴、および実施形態が当業者に明らかになる。本発明の特徴は、以下のいくつかの態様および図面に関連して説明され得るが、本発明のすべての態様は、本明細書で説明される有利な特徴のうちの1つまたは複数を含み得る。言い換えれば、1つまたは複数の態様は、いくつかの有利な特徴を有するものとして説明され得るが、そのような特徴のうちの1つまたは複数またはまた、本明細書で説明される本発明の様々な態様に従って使用され得る。同様に、例示的な態様は、デバイスの態様、システムの態様、または方法の態様として以下で説明され得るが、そのような例示的な態様は様々なデバイス、システム、および方法において実施され得ることを理解されたい。

30

【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。

40

【図面の簡単な説明】

【0013】

【図1】本開示の一態様による例示的なマルチマスタバスを示す図である。

【図2】本開示の一態様による例示的なマスタ/スレーブデバイスのブロック図である。

【図3】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4】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5】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

50

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。

【図6】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7】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8】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9】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10】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。

【図11】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12】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13】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(general call)を示す図である。

【図14】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。

【図15】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16】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17】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18】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19】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20】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21】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22】本明細書で開示する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。

【図23】図22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24】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25】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_0000₃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かを示す図である。

【図26】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_1111₃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かを示す図である。

【図27】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_2222₃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かを示す図である。

【図28】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かを示す図である。

【図29】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

10

20

30

40

50

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30】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31】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32】本開示の一態様による例示的な符号化/復号方法を示すフローチャートである。

【図33】本開示の一態様による例示的なエンコーダ構成要素を示すブロック図である。

【図34】本開示の一態様による例示的な符号化方法を示すフローチャートである。

【図35】本開示の一態様による例示的なデコーダ構成要素を示すブロック図である。

【図36】本開示の一態様による例示的な復号方法を示すフローチャートである。

【発明を実施するための形態】

【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》

背景技術において説明したように、従来のエラー検出アルゴリズムの制限のために、共有バス上で通信されるエラーが逃されることが多い。本明細書で開示する態様は、エラー検出最適化を容易にするために、所望のワードフォーマットに従ってビットを割り振ることによって、そのような制限を克服することに向けられる。すなわち、エラー検出最適化を容易にするために、追加のエラー検出ビットが戦略的に割り振られ得る、共有バス通信のための柔軟なワードフォーマットを利用することに向けられた態様を開示する。

【0016】

次に図1を参照すると、本明細書で開示するエラー検出最適化態様を容易にする、例示的なマルチマスタバスアーキテクチャが提供されている。図示のように、複数のマスタ/スレーブデバイス110、120、130、および140は、共有バス100を介して互いに結合される。ここで、共有バス100はマルチマスタバスであり、マスタ/スレーブデバイス110、120、130、および140のいずれも、マスタデバイスまたはスレーブデバイスとして動作することができることが企図される。この特定の例では、マスタ/スレーブデバイス120は、共有バス100を介して他のマスタ/スレーブデバイス110、130、および140へワード122を送信し、ワード122は、マスタ/スレーブデバイス120によって、エラー検出を最適化するビット割り振り方式に従って符号化される。たとえば、そのような方式は、非最適化方式で割り振られるエラー検出ビットの数と比較して、より多数のエラー検出ビットを割り振ることによって、エラー検出を最適化することができる。したがって、ワード122を適切に復号するために、マスタ/スレーブデバイス110、130、および140は、ワード122がエラー検出最適化を用いて符号化されたか否かを検出し、次いで、ワード122が対応するビット割り振り方式に基づいて復号されることが企図される。

【0017】

図2では、本開示の一態様による例示的なマスタ/スレーブデバイスのブロック図が提供されている。図示のように、マスタ/スレーブデバイス200は、エンコーダ構成要素210と、デコーダ構成要素220と、通信構成要素230とを含む、本明細書で開示するエラー検出最適化を実施することを容易にするための様々な構成要素を備える。マスタ/スレーブデバイス200は、たとえば、図1に示すマスタ/スレーブデバイス110、120、130、および140のいずれかを含む、本明細書で説明する任意のマスタ/スレーブデバイスとして構成され得

10

20

30

40

50

ることが企図される。たとえば、通信構成要素230は、共有バスを介して通信されるワードを送信および受信するように構成されてよく、エンコーダ構成要素210は、追加のエラー検出ビットを含めるためにワードを符号化することによって、エラー検出を最適化するように構成され、デコーダ構成要素220は、これらの追加のエラー検出ビットを含むワードを復号するように構成されることが企図される。

【0018】

本開示の特定の態様では、マスタ/スレーブデバイス200は、CCIプロトコルに従ってワードを符号化/復号するように構成される。このために、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、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ことに留意されたい。例示的な実装形態では、20ビットの2進数が3進数変換器(すなわち、ビット-12xT変換器)に並行して入力される。すべての2進ビットを受信した後、3進数変換器は、対応する3進数を出力する。次いで、出力された数が同様の方法でトランスコーダへ送られる。本明細書で開示するCCleの一態様では、3進遷移番号-順次シンボル変換がシンボルごとに実施され、この変換には、複数のシンボルを同時に処理するよりも少ないハードウェアリソースが必要となることが望ましい。次に、それらのシンボルがバスを介して送信される。

【0019】

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、ホットプラグ機能、および/またはSIDスキャンがすべて、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。

【0020】

《例示的な動作環境》

図3は、ベースバンドプロセッサ304およびイメージセンサ306を有し、画像データバス316およびマルチモード制御データバス308を実装するデバイス302を示すブロック図である。図3はカメラデバイス内のマルチモード制御データバス308を示すが、この制御データバス308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316(たとえば、高速差動DPHYリンク)を介して、イメージセンサ306からベースバンドプロセッサ304に送られる場合がある。

【0021】

一例では、制御データバス308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308)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308)上ですべてのデバイス312、314、および318に結合される。この例では、制御データが、制御データバス308を介して、ベースバンドプロセッサ304とイメージセンサ306ならびに他の周辺デバイス318、322、および/または324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。

【0022】

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308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(CCle)モードと呼ばれる場合がある。CCle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304はマスタノード312を含み、イメージセンサ306はスレーブノード314を含み、マスタノード312とスレーブノード314の両方は、制御データバス308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308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308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。

【0023】

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308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308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308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312によって管理される。マスタデバイスは、制御データバス308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308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308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308上で動作することができるかを確認する。

【0024】

《例示的なCCIe符号化技法》

図4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404は、送信されるシンボルが、送信シンボル406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402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406内の)各シンボルにおける状態遷移からクロック情報408を復元し、次に、送信シンボル406のトランスコーディングを逆転させて、元のデータ410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402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(たとえば、図3における制御データバス308、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。

【0025】

図5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502において、データビット504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SCL512およびデータラインSDA514を介して送信される。

【0026】

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508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510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516が、2線式物理リンク(たとえば、SCLライン512とSDAライン514とを備えるI2Cバス)を介して送られる。

【0027】

図6は、遷移番号602と順次シンボル604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のうちの一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。

10

【0028】

この変換関数を図7に例示的に記載する。送信機側(TX:TからSへ)702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され得る。一時遷移番号(T_{tmp})は、現在の遷移番号Tが0と等しくなり、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、 $T_{tmp}=T=0?3:T$)。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、 $C_s=P_s+T_{tmp}$)。

20

【0029】

受信機側(RX:SからTへ)704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、 $T=T_{tmp}=3?0:T$)。

30

【0030】

テーブル706は、遷移番号と順次シンボルとの間の変換を示す。

【0031】

再び図6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606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。

【0032】

第2のサイクル608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。

40

【0033】

第3のサイクル610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。

【0034】

第4のサイクル612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

50

遷移番号 T_{tmp} は3である。

【0035】

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。

【0036】

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604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。

10

【0037】

図5を再び参照すると、受信機520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520は、2線式物理リンク(たとえば、SCLライン524およびSDAライン526を備えるI2Cバス)を介して、順次シンボル522のシーケンスを受信する。受信された順次シンボル522は、クロック-データ復元(CDR)ブロック528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530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532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。

20

【0038】

2線式バスおよび12個の遷移番号について図5および図6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。

【0039】

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、制御データバス(たとえば、図3における制御データバス308)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4、図5、図6、および図7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。

30

【0040】

図8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802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802の最上位桁 T_{11} が最初に取得される。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802の桁の各々はまた、「遷移番号」と呼ばれることもある。

40

【0041】

図9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8および図9は、 $T_{11}, T_{10}, T_9, \dots, T_0$ の順序で送られる12桁の3進数802を示す。最上位ビットを最初に取得し、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8および図9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802の最下位

50

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6および図7の説明を参照すると、T0が順次シンボルにトランスコーディングされる時、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802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6および図7の説明を参照すると、T11が順次シンボルにトランスコーディングされる時、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530(図5)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T11が最初に、最下位桁T0が最後になる。

【0042】

図5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。

【0043】

図10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8および図9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進数は逆順で受信機側論理回路に供給され、受信機側論理回路は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。

【0044】

図11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。言い換えれば、図5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)、次いで、この遷移番号が、逆順で再び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、バス上で逆順に送信される。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、次いで、3進数は逆順で図11における回路に供給され、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。

【0045】

図12は、どのようにビット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かを、概念的に示す。すなわち、本明細書で説明する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12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トは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、 $0000_0000_0000_3$ から $2221_2201_2001_3$ の3進数範囲内で表される。 $2221_2201_2002_3$ から $2222_2222_2222_3$ 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 $2221_2201_2002_3$ から $2222_2222_2222_3$ 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、 $2221, 2201, 2002_3$ 3進数は、10,000,000,000,000,000,000 2進数(0x80000 16進数)であり、 $2222_2222_2222_3$ 3進数(0x81BF0)は、可能な最大の12桁の3進数である。

【0046】

《CCleモードのための例示的なプロトコル》

図13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

10

20

30

40

50

、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を示す。一般呼出し1302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3におけるマスタデバイス312)によって発行され得る。

【0047】

I2Cモードでは、CCleマスタデバイスは、このI2C一般呼出し1302を、「CCleモード」バイトまたはインジケータ1304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1302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(たとえば、図3における制御データバス308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。

10

【0048】

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。

【0049】

図14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3におけるマスタデバイス312)によって発行され得る、例示的なCCle呼出し1402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1402を発行することができる。

20

【0050】

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1404を送る。加えて、「終了」コード/インジケータ1404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」1406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1408とともに送る。すべてのCCle対応スレーブは、一般呼出し1404に対して肯定応答しなければならない。

【0051】

図15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1502の一部としての16ビットのスレーブ識別子(SID)1504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。

30

【0052】

図16は、例示的なCCleアドレスワードフォーマット1602を示す。これは、各アドレスワード1606が16ビットのアドレス1604を含むことを示す。アドレスワード1606はまた、2ビットの制御コード1608と、1ビットのエラー検出定数1610とを含む。テーブル1612は、制御コードのための様々な可能な値を示す。

【0053】

複数のアドレスワードが順次送られ得る。現在の制御コードが「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、読取り指定データワードである。制御コード「11」は禁止される。

40

【0054】

図17は、例示的な書込みデータワードフォーマット1700を示す。これは、各書込みデータワード1700が16ビットの書込みデータ部分1702を含むことを示す。書込みデータワード1700はまた、2ビットの制御コード1704と、1ビットのエラー検出定数1710とを含む。テーブル1714は、制御コードのための様々な可能な値を示す。

【0055】

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

50

」(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。

【0056】

図18は、例示的な読取り指定ワードフォーマット1800を示す。読取り指定データワード1800は、16ビットの読取りデータ値部分1804と、2ビットの制御コード1808と、3ビットのエラー検出定数1810とを含み得る。

【0057】

最後のアドレスワード1807の後、「読取り指定」(RS)ワード1812が後続する。読取り指定(RS)ワード1812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1816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1804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1804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。

10

【0058】

図19は、例示的な読取りデータワードフォーマット1902を示す。読取りデータワード1902は、16ビットの読取りデータ値部分1904と、2ビットの制御コード1906と、1ビットのエラー検出定数1908とを含み得る。SID1907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1916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。

20

【0059】

《例示的な共有バスを介したI2C送信対CC1e送信》

図20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(たとえば、図3における制御データバス308)は、シリアルデータラインSDA2002とシリアルクロックラインSCL2004とを含む。図20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2004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2002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、SDAライン2002において7ビットのスレーブID2008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ID2008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ライン上で8個のクロックパルス)を送らなければならない。

30

40

【0060】

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2006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。

【0061】

CC1e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2002とSCLライン2004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20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

50

イン2002とSCLライン2004の両方が、データ送信のために使用され得る。

【0062】

図21は、SDAライン2102およびSCLライン2104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21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化する。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。

【0063】

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2106、2108、2110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2106、2108、2110(すなわち、SCLライン2104が高である間に、SDAライン2102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2106、2108、2110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2106、2108、2110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。

【0064】

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、割り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。割り込みを発行するための従来技術の機構は、専用の割り込みラインまたは専用の割り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り込みラインまたはバスは、そのような割り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り込みピンおよびライン/バスの必要性をなくすために、CCle内の帯域内割り込みのための機構が必要とされる。

【0065】

帯域内割り込みの使用はまた、バス競合または衝突を回避するべきである。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2102またはSCLライン2104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。

【0066】

《例示的なビット19領域およびチェックサム》

図22は、本明細書で開示する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。

【0067】

図23は、図22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。

10

20

30

40

50

【 0 0 6 8 】

図24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2402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。

【 0 0 6 9 】

受信機クロック(RXCLK)が2つのサイクル2412および2414を消失し、データビット2410が誤って検出されるようになる、クロック消失2404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。

10

【 0 0 7 0 】

受信機クロック(RXCLK)が、余剰クロックサイクル2420において検出された余剰シンボル「01」2416および2418を有する、余剰クロック2406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。

【 0 0 7 1 】

受信機クロック(RXCLK)消失がないが、単一シンボルエラー2422が発生する、シンボルエラー2408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。

20

【 0 0 7 2 】

《 例示的な送信シンボル内のエラー検出 》

図25～図30は、様々なCC1e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。

【 0 0 7 3 】

図25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5、図6、図7、図8、図9および図10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2502が、3進数(T11...T0)0000_0000_0000₃ 2504に変換され、次にこの3進数が、順次シンボル(S11...S0)0321_0321_0321 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0321_0321_0321 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「010」への変化を生じる。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。

30

40

【 0 0 7 4 】

図26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5、図6、図7、図8、図9および図10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2602が、3進数(T11...T0)1111_1111_1111₃ 2604に変換され、次にこの3進数が、順次シンボル(S11...S0)2301_2301_2301 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2301_2301_2301 2606のシンボルのいずれかにおいて、送信中にエラーが

50

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「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。

【 0 0 7 5 】

図27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かを示す。図5、図6、図7、図8、図9および図10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27502が、3進数(T11...T0)222_2222_2222₃ 2704に変換され、次にこの3進数が、順次シンボル(S11...S0)3131_3131_3131 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3131_3131_3131 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「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。

【 0 0 7 6 】

図28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かを示す。図5、図6、図7、図8、図9および図10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2802が、3進数(T11...T0)0120_1201_2012₃ 2804に変換され、次にこの3進数が、順次シンボル(S11...S0)0132_3101_3231 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0132_3101_3231 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「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。

【 0 0 7 7 】

図29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2900、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かを示す。図5、図6、図7、図8、図9および図10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2902が、3進数(T11...T0)1201_2012_0120₃ 2904に変換され、次にこの3進数が、順次シンボル(S11...S0)2030_2120_3021 2906に変換される。この例では、3桁の最下位ビット2908がすべて0(000)である。元の順次シンボル2030_2120_3021 2906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2910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

10

20

30

40

50

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2900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。

【 0 0 7 8 】

図30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3000、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かを示す。図5、図6、図7、図8、図9および図10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3002が、3進数(T11...T0)2012_0120_1201、3004に変換され、次にこの3進数が、順次シンボル(S11...S0)3231_0132_3101 3006に変換される。この例では、3桁の最下位ビット3008がすべて0(000)である。元の順次シンボル3231_0132_3101 3006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3010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3000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。

【 0 0 7 9 】

《 例示的なマスタ/スレーブデバイス実装形態 》

次に図31を参照すると、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が示されている。図示のように、マスタ/スレーブデバイス3114は、制御データバス3150を介して別のマスタ/スレーブデバイス3160に結合される。ここで、マスタ/スレーブデバイス3114または3160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3114および3160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。

【 0 0 8 0 】

この例では、マスタ/スレーブデバイス3114は、バス3102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3102は、マスタ/スレーブデバイス3114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3102は、(プロセッサ3104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3105、および(コンピュータ可読媒体3106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3102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。

【 0 0 8 1 】

特定の実装形態では、制御データバスインターフェース3108は、制御データバス3150とマスタ/スレーブデバイス3114との間のインターフェースを提供し、プロセッサ3104は、制御データバス3150を介した、マスタ/スレーブデバイス3114とマスタ/スレーブデバイス3160との間のワードの符号化通信を容易にするように構成される。ここで、制御データバス3150は2ラインバスであり得ること、および、符号化通信は、エラー検出定数を最大化することを容易にするために、符号化通信の複数の最下位ビットを割り振るプロトコル(たとえば、CCleプロトコル)に従って符号化され得ることが企図される。前述のように、そのような最大化は、ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、複数の最下位ビットを割り振るプロトコルを介して達成され得る。

【 0 0 8 2 】

本開示のさらなる態様では、コンピュータ可読媒体3106は、本明細書で開示するように、制御データバス3150を介したエラー検出最適化を容易にするために、様々な命令3106a

10

20

30

40

50

、3106b、および/または3106cを含むように構成される。同様に、そのような態様は、代わりに、図示のようにプロセッサ3104を図示の回路3120、3130、および/または3140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。その上、制御データバス3150を介したエラー検出最適化は、命令3106a、3106b、および/または3106cの任意の組合せ、ならびに、回路3120、3130、および/または3140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。

【0083】

たとえば、エンコーダ/デコーダ命令3106aおよびエンコーダ/デコーダ回路3120は、選択された/検出されたプロトコル(たとえば、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、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」を生じることを含み得る。

10

【0084】

本開示の別の態様では、ビット割振り命令3106bおよびビット割振り回路3130は、所望のワードフォーマット(たとえば、SIDワードフォーマット、アドレスワードフォーマット、書込みデータワードフォーマット、読取り指定ワードフォーマット、または読取りデータワードフォーマット)に従ってビットを割り振ることに向けられる。この目的で、本明細書で開示する様々な企図されたワードフォーマットは、20ビットのワードフォーマットを備え、エラー検出定数を最大化することを容易にするために、3桁の最下位ビットが割り振られる。その上、ビット割振り命令3106bおよび/またはビット割振り回路3130のいずれかは、エラー検出最適化が所望されるか、データ最適化が所望されるかによって、そのような最大化を容易にするために、柔軟なビット割振り方式を容易にするように構成され得ることが企図される。たとえば、特定の実装形態では、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、ワードのデータ部分の追加のエラー検出ビットまたは2桁の最上位ビットのいずれかのために割り振られる。

20

【0085】

本開示の別の態様では、通信命令3106cおよび/または通信回路3140は、マスタ/スレーブデバイス3114を制御データバス3150とインターフェースするように構成され得る。特に、通信命令3106cおよび/または通信回路3140のいずれかは、本明細書で開示するエラー検出最適化を容易にするプロトコル(たとえば、CCleプロトコル)に従って、マスタ/スレーブデバイス3114とマスタ/スレーブデバイス3160との間のワードの符号化通信を容易にするように構成され得る。

30

【0086】

図31の残りの要素に戻って参照すると、プロセッサ3104は、バス3102を管理すること、および、コンピュータ可読媒体3106上に記憶されたソフトウェアの実行を含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、プロセッサ3104によって実行される時、マスタ/スレーブデバイス3114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3106は、ソフトウェアを実行するときにプロセッサ3104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。

40

【0087】

処理システム内の1つまたは複数のプロセッサ3104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3106上に存在することができる。コンピュータ可読媒体3106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

50

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3106は、マスタ/スレーブデバイス3114の中に、マスタ/スレーブデバイス3114の外部に存在するか、またはマスタ/スレーブデバイス3114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3106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。

10

【0088】

次に図32を参照すると、本明細書で開示する態様による、共有バスを介したエラー検出最適化を容易にする例示的な方法を示すフローチャートが提供されている。図示のように、プロセス3200は、本明細書の一態様による、コンピューティングデバイス(たとえば、マスタ/スレーブデバイス3114)内で実施され得る一連の動作を含む。たとえば、プロセス3200は、プロセッサを採用して、コンピュータ可読記憶媒体上に記憶されたコンピュータ実行可能命令を実行して、一連の動作を実施することによって実施され得る。別の実施形態では、少なくとも1つのコンピュータにプロセス3200の動作を実施させるためのコードを備えるコンピュータ可読記憶媒体が企図される。

20

【0089】

図示のように、プロセス3200は、動作3210で、マスタデバイスをスレーブデバイスに結合することから開始する。ここで、そのような結合は、制御データバスを介してマスタデバイスとスレーブデバイスとを接続することを含み得ることを諒解されたい。次いで、プロセス3200は動作3220へ進み、そこで、制御データバスを介したマスタデバイスとスレーブデバイスとの間のワードの符号化通信が、(たとえば、所望のプロトコル、所望のワードフォーマットなどを選択して)容易にされる。ここで、符号化通信は、ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、符号化通信の複数の最下位ビットを割り振ることによって、エラー検出定数を最大化することを容易にするために、複数の最下位ビットを割り振るプロトコルに従って符号化されることが企図される。符号化態様と復号態様の両方が企図されるので、プロセス3200は、動作3230で、エンコード動作を進めるか、デコード動作を進めるかを決定することをさらに含み得る。たとえば、エンコードとして動作するとき、プロセス3200は動作3240へ進んでよく、そこで、本明細書で開示する態様に従って、エラー検出定数を最大化することを容易にするために、符号化通信の複数の最下位ビットを割り振るプロトコル(たとえば、CCleプロトコル)に従って、ワードが符号化され、その後、符号化通信が制御データバスを介して送信される動作3242で終了し得る。さもなければ、デコードとして動作する場合、プロセス3200は動作3250へ進んでよく、そこで、符号化通信が制御データバスを介して受信され、その後、本明細書で開示するエラー検出最適化を容易にするプロトコル(たとえば、CCleプロトコル)に従って、符号化通信が復号される動作3252で終了し得る。

30

40

【0090】**《例示的なエンコード実装形態》**

再び図31を参照すると、次に例示的な実装形態について、マスタ/スレーブデバイス311

50

4をエンコーダとして構成するコンテキスト内で説明する。そのような実装形態を容易にするために、エンコーダ/デコーダ回路3120がエンコーダ回路として構成され得ること、および、エンコーダ/デコーダ命令3106aがエンコーダ命令として構成され得ることが企図される。この目的で、図33に図示するように、エンコーダ回路3120およびエンコーダ命令3106aの各々が、複数のサブ構成要素のいずれかを介して、本明細書で開示する態様に従ってワードの符号化を容易にするように構成され得ることがさらに企図される。すなわち、図33に図示するように、エンコーダ回路3120は、プロトコルサブ回路3310と、最適化サブ回路3320と、符号化サブ回路3330とを備え得るのに対して、エンコーダ命令3106aは、プロトコル命令3312と、最適化命令3322と、符号化命令3332とを備え得る。この特定の実装形態では、ビット割振り回路3130およびビット割振り命令3106bの各々は、ビット割振り方式に従ってビットを割り振ることに向けられており、ビット割振り方式は、エラー検出定数を最大化することを容易にするために、符号化通信の複数の最下位ビットを割り振る。次いで、プロトコルサブ回路3310およびプロトコル命令3312の各々は、所望のプロトコル(たとえば、CCleプロトコル)に関連付けられたワードのワードフォーマットを決定することに向けられるのに対して、符号化サブ回路3330および符号化命令3332の各々は、(たとえば、シンボルにトランスコーディングされた、符号化された3進数として、ワードを符号化することによって)符号化通信を生成するために、前述のワードフォーマットとビット割振り方式とに従って、ワードを符号化することに向けられる。ワードが符号化されると、通信回路3140および/または通信命令3106cのいずれかが、制御データバスを介して符号化通信を送信するために使用され得る。

10

20

【0091】

本開示のさらなる態様では、最適化サブ回路3320および/または最適化命令3322のいずれかが、所望のワードフォーマットと対応するビット割振り方式とを介して実施するための最適化を確認するように構成され得ることが企図される。特定の実装形態では、最適化サブ回路3320および/または最適化命令3322は、第1のビット割振り方式を有するエラー検出最適化に従うワードの符号化と、第2のビット割振り方式を有するデータ最適化に従うワードの符号化との間で切り替えることを容易にするように構成され得る。たとえば、エラー検出最適化がデータ最適化よりも好適であるとき、符号化サブ回路3330および/または符号化命令3332は、複数の最下位ビットが固定数の3ビット(たとえば、3桁の最下位ビット)を備えるエラー検出最適化に従って、ワードを符号化するように構成されてよく、ビット割振り回路3130および/またはビット割振り命令3106bは、最下位ビット、第2の最下位ビット、および第3の最下位ビットの各々をエラー検出のために割り振ることによって、エラー検出最適化を容易にするように構成される。しかしながら、データ最適化がエラー検出最適化よりも好適であるとき、符号化サブ回路3330および/または符号化命令3332は、代わりに、複数の最下位ビットが固定数の3ビットを備えるデータ最適化に従って、ワードを符号化するように構成されてよく、ビット割振り回路3130および/またはビット割振り命令3106bは、最下位ビットをエラー検出のために、第2の最下位ビットをワードのデータ部分の最上位ビットのために、および第3の最下位ビットをワードのデータ部分の第2の最上位ビットのために割り振ることによって、データ最適化を容易にするように構成される。

30

40

【0092】

次に図34を参照すると、本明細書で開示する態様による、例示的な符号化方法を示すフローチャートが提供されている。図示のように、プロセス3400は、本明細書の一態様による、コンピューティングデバイス(たとえば、マスタ/スレーブデバイス3114)内で実施され得る一連の動作を含む。たとえば、プロセス3400は、プロセッサを採用して、コンピュータ可読記憶媒体上に記憶されたコンピュータ実行可能命令を実行して、一連の動作を実施することによって実施され得る。別の実施形態では、少なくとも1つのコンピュータにプロセス3400の動作を実施させるためのコードを備えるコンピュータ可読記憶媒体が企図される。

【0093】

50

図示のように、プロセス3400は、動作3410で、符号化プロトコル(たとえば、CCleプロトコル)の選択から開始する。次いで、プロセス3400は動作3420へ進み、そこで、マスタ/スレーブデバイスは、選択されたプロトコルを介して実施するための所望の最適化を確認し、次いで動作3430で、所望の最適化のための適切なワードフォーマットが決定され、その後、動作3440で、所望の最適化に従ってビットが割り振られる。たとえば、最大シンボルエラー検出が所望される場合、動作3430は、20ビットのCCleワードフォーマットを利用することを含んでよく、動作3440は、そのようなフォーマットの3桁の最下位ビットをエラー検出定数のために割り振ることを含み得る。さもなければ、データスループット最適化が所望される場合、動作3430は再び、20ビットのCCleワードフォーマットを利用することを含み得るが、動作3440は次に、最下位ビットのみをエラー検出定数のために割り振ることを含み得るのに対して、第2の最下位ビットがワードのデータ部分の最上位ビットのために割り振られ、第3の最下位ビットがワードのデータ部分の第2の最上位ビットのために割り振られる。

10

【0094】

動作3440で、適切なビット割振りが実施されると、プロセス3400は動作3450へ進み、そこで、所望の最適化のワードフォーマットとビット割振り方式とに従って、ワードが符号化される。ここで、前述のように、そのような符号化は、シンボルにトランスコーディングされた、符号化された3進数として、ワードを符号化することを含み得る。次いで、プロセス3400は動作3460で終了し、そこで、符号化通信が制御データバスを介して他のマスタ/スレーブデバイスへ送信される。

20

【0095】**《例示的なデコーダ実装形態》**

再び図31を参照すると、次に例示的な実装形態について、マスタ/スレーブデバイス3114をデコーダとして構成するコンテキスト内で説明する。そのような実装形態を容易にするために、エンコーダ/デコーダ回路3120がデコーダ回路として構成され得ること、および、エンコーダ/デコーダ命令3106aがデコーダ命令として構成され得ることが企図される。この目的で、図35に図示するように、デコーダ回路3120およびデコーダ命令3106aの各々が、複数のサブ構成要素のいずれかを介して、本明細書で開示する態様に従ってワードの復号を容易にするように構成され得ることがさらに企図される。すなわち、図35に図示するように、デコーダ回路3120は、プロトコルサブ回路3510と、最適化サブ回路3520と、復号サブ回路3530とを備え得るのに対して、デコーダ命令3106aは、プロトコル命令3512と、最適化命令3522と、復号命令3532とを備え得る。この特定の実装形態では、通信回路3140および/または通信命令3106cのいずれかが、制御データバスを介して符号化通信を受信するように構成され得るのに対して、デコーダ回路3120および/またはデコーダ命令3106aは、符号化通信の復号を容易にするように構成され得る。次いで、プロトコルサブ回路3510およびプロトコル命令3512の各々は、プロトコル(たとえば、CCleプロトコル)に関連付けられた符号化通信中に含まれたワードのワードフォーマットを検出することに向けられ、最適化サブ回路3520および最適化命令3522の各々は、符号化通信の最適化と、最適化に対応するビット割振り方式とを確認するように構成される。次いで、復号サブ回路3530および復号命令3532は、適切なワードフォーマットと対応するビット割振り方式とに従って(たとえば、ビットマップを利用することによって)、符号化通信を復号するように構成され得る。

30

40

【0096】

次に図36を参照すると、本明細書で開示する態様による、例示的な復号方法を示すフローチャートが提供されている。図示のように、プロセス3600は、本明細書の一態様による、コンピューティングデバイス(たとえば、マスタ/スレーブデバイス3114)内で実施され得る一連の動作を含む。たとえば、プロセス3600は、プロセッサを採用して、コンピュータ可読記憶媒体上に記憶されたコンピュータ実行可能命令を実行して、一連の動作を実施することによって実施され得る。別の実施形態では、少なくとも1つのコンピュータにプロセス3600の動作を実施させるためのコードを備えるコンピュータ可読記憶媒体が企図さ

50

れる。

【0097】

図示のように、プロセス3600は、動作3610で、符号化通信が別のマスタ/スレーブデバイスから共有バスを介して受信されることから開始する。次いで、プロセス3600は動作3620へ進み、そこで、マスタ/スレーブデバイスは、符号化通信に対応するワードフォーマットと関連付けられたプロトコルとを検出する。受信された通信が特定の最適化に従って符号化され得ることが企図されるので、次いで、プロセス3600は、動作3630でそのような最適化を確認し、その後、動作3640で、最適化に対応するビットマップを取り出すことができる。たとえば、最大シンボルエラー検出に対応するワードフォーマットが検出される場合、20ビットを備えるビットマップが使用されてよく、3桁の最下位ビットがエラー検出定数のために割り振られ得る。さもなければ、データスループット最適化が検出される場合、ビット割り振り方式は、最下位ビットのみをエラー検出定数のために割り振ること、ならびに、第2の最下位ビットおよび第3の最下位ビットを、それぞれワードのデータ部分の第1の最上位ビットおよび第2の最上位ビットのために割り振ることを含み得る。適切なビット割り振り方式が識別されると、次いで、プロセス3600は動作3650で終了し、そこで、符号化通信が、動作3640で取り出されたビットマップに従って復号される。

10

【0098】

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。

20

【0099】

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。

30

【0100】

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。

40

【0101】

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

50

、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。

【0102】

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。

10

【0103】

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。

20

【0104】

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。

30

【0105】

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る、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。

40

【符号の説明】

【0106】

- 100 共有バス
- 110 マスタ/スレーブデバイス
- 120 マスタ/スレーブデバイス
- 122 ワード

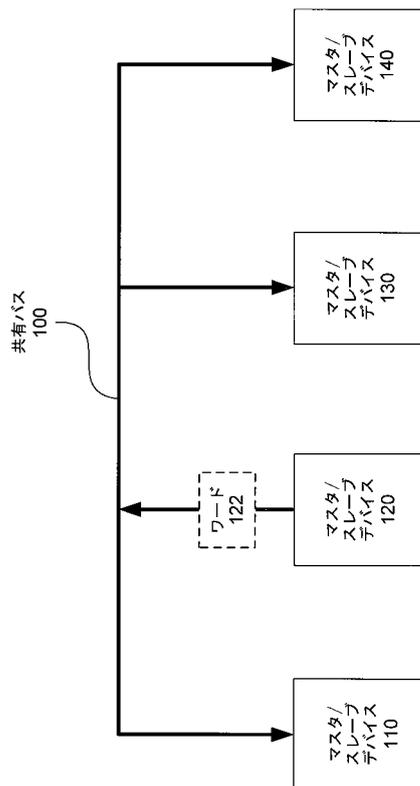
50

130	マスタ/スレーブデバイス	
140	マスタ/スレーブデバイス	
200	マスタ/スレーブデバイス	
210	エンコーダ構成要素	
220	デコーダ構成要素	
230	通信構成要素	
302	デバイス	
304	ベースバンドプロセッサ	
306	イメージセンサ	
308	マルチモード制御データベース、制御データベース、共有制御データベース、共有バ	10
ス		
312	デバイス、マスタデバイス、マルチモードマスタデバイス、マスタノード	
314	デバイス、スレーブノード	
316	画像データベース	
318	デバイス、周辺デバイス	
322、324	周辺デバイス	
402	元のクロック	
404	データ	
406	送信シンボル	
408	クロック情報	20
410	元のデータ	
502	送信機	
504	データビット	
508	ビット-遷移番号変換器ブロック	
510	遷移-シンボルブロック	
516、522、604	順次シンボル	
520	受信機	
524	SCLライン	
526	SDAライン	
528	クロック-データ復元(CDR)ブロック	30
530	シンボル-遷移番号変換器ブロック	
532	遷移番号-ビット変換器	
602	遷移番号	
606	第1のサイクル	
608	第2のサイクル	
610	第3のサイクル	
612	第4のサイクル	
702	送信機側(TX:TからSへ)	
704	受信機側(RX:SからTへ)	
706、1612、1714、1816、1916、2500、2600、2700、2800、2900、3000	テーブル	40
802	3進数	
1302	一般呼出し、I2C一般呼出し	
1304	「CCleモード」バイトまたはインジケータ	
1402	CCle呼出し、終了呼出し	
1404	「終了」コード/インジケータ、一般呼出し	
1406	「一般呼出し」	
1408	「終了」コード	
1502	CCle SIDワードフォーマット	
1504	16ビットのスレーブ識別子(SID)	
1602	CCleアドレスワードフォーマット	50

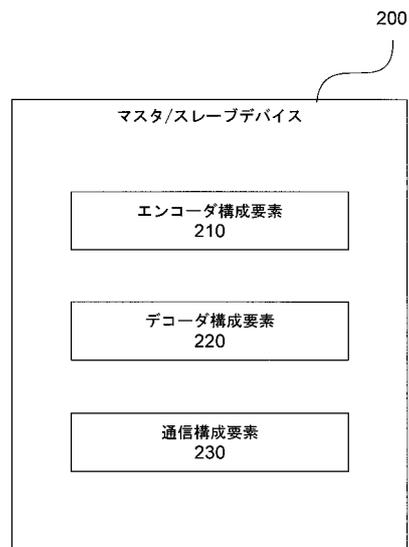
1604	16ビットのアドレス	
1606	アドレスワード	
1608、1704、1808、1906	制御コード	
1610、1710、1810、1908	エラー検出定数	
1700	書込みデータワードフォーマット、書込みデータワード	
1702	16ビットの書込みデータ部分	
1800	読取り指定ワードフォーマット、読取り指定データワード	
1804	16ビットの読取りデータ値部分、「読取り指定」(RS)ワード	
1807	最後のアドレスワード	
1812	「読取り指定」(RS)ワード	10
1902	読取りデータワードフォーマット、読取りデータワード	
1904	16ビットの読取りデータ値部分	
1907	SID	
2002	シリアルデータラインSDA、SDAライン	
2004	シリアルクロックラインSCL、SCLライン	
2006、2106、2108、2110	START条件	
2008	スレーブID	
2102	SDAライン	
2104	SCLライン	
2402	タイミング図	20
2404	クロック消失	
2406	余剰クロック	
2408	シンボルエラー	
2410	データビット	
2412、2414	サイクル	
2416、2418	余剰シンボル「01」	
240	余剰クロックサイクル	
2422	単一シンボルエラー	
2502	(ビット[19:0])0000_0000_0000_0000_0000の20ビットシーケンス	
2504	3進数(T11...T0)0000_0000_0000 ₃	30
2506	順次シンボル(S11...S0)0321_0321_0321、順次シンボル0321_0321_0321	
2508、2608、2708、2808、2908、3008	3桁の最下位ビット	
2510、2610、2710、2810、2910、3010	誤ったシンボル	
2602	(ビット[19:0])0100_0000_1101_1111_1000の20ビットシーケンス	
2604	3進数(T11...T0)1111_1111_1111 ₃	
2606	順次シンボル(S11...S0)2301_2301_2301、順次シンボル2301_2301_2301	
2702	(ビット[19:0])1000_0001_1011_1111_0000の20ビットシーケンス	
2704	3進数(T11...T0)2222_2222_2222 ₃	
2706	順次シンボル(S11...S0)3131_3131_3131、順次シンボル3131_3131_3131	
2802	(ビット[19:0])0001_1000_1111_0011_1000の20ビットシーケンス	40
2804	3進数(T11...T0)0120_1201_2012 ₃	
2806	順次シンボル(S11...S0)0132_3101_3231、順次シンボル0132_3101_3231	
2902	(ビット[19:0])0100_1010_1101_1010_1000の20ビットシーケンス	
2904	3進数(T11...T0)1201_2012_0120 ₃	
2906	順次シンボル(S11...S0)2030_2120_3021、順次シンボル2030_2120_3021	
3002	(ビット[19:0])0101_1110_1101_0000_1000の20ビットシーケンス	
3004	3進数(T11...T0)2012_0120_1201 ₃	
3006	順次シンボル(S11...S0)3231_0132_3101、順次シンボル3231_0132_3101	
3102	バス	
3104	プロセッサ	50

- 3105 メモリ
- 3106 コンピュータ可読媒体
- 3106a エンコーダ/デコーダ命令、命令、エンコーダ命令、デコーダ命令
- 3106b 命令、ビット割振り命令
- 3106c 命令、通信命令
- 3108 制御データバスインターフェース
- 3114 マスタ/スレーブデバイス
- 3120 エンコーダ/デコーダ回路、回路、エンコーダ回路、デコーダ回路
- 3130 回路、ビット割振り回路
- 3140 回路、通信回路 10
- 3150 制御データバス
- 3160 マスタ/スレーブデバイス
- 3200、3400、3600 プロセス
- 3310、3510 プロトコルサブ回路
- 3312、3512 プロトコル命令
- 3320、3520 最適化サブ回路
- 3322、3522 最適化命令
- 3330 符号化サブ回路
- 3332 符号化命令
- 3530 復号サブ回路 20
- 3532 復号命令

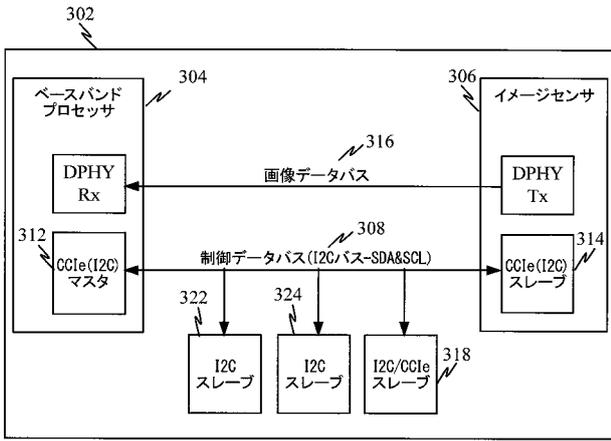
【 図 1 】



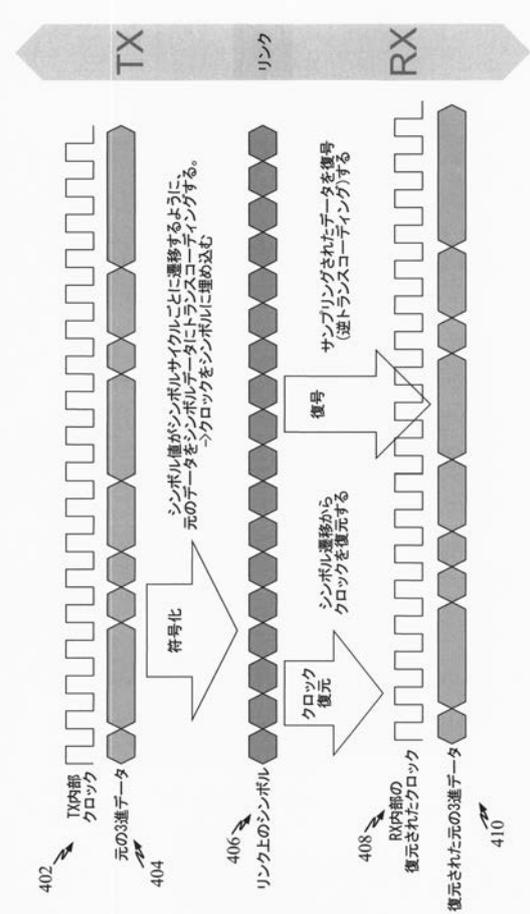
【 図 2 】



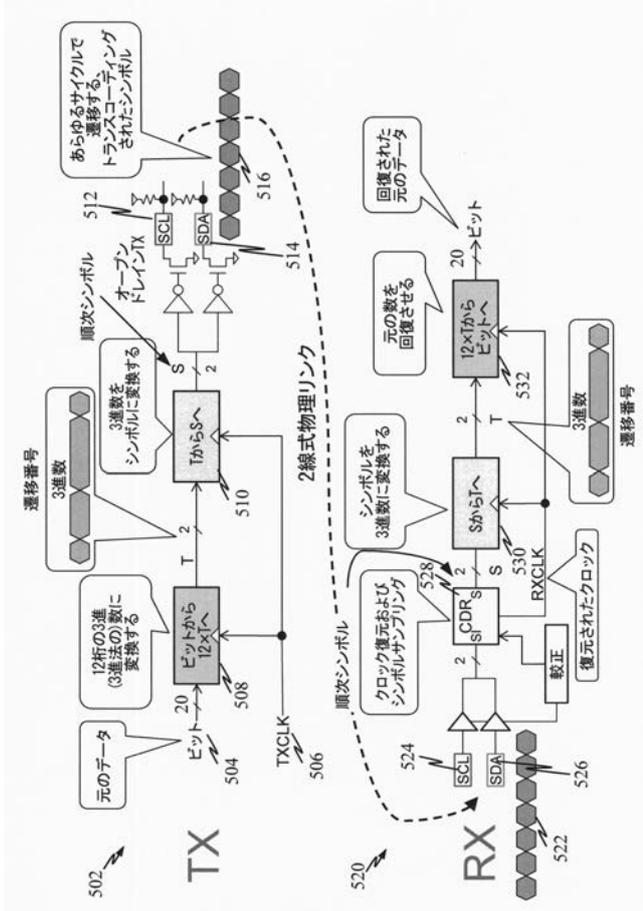
【図3】



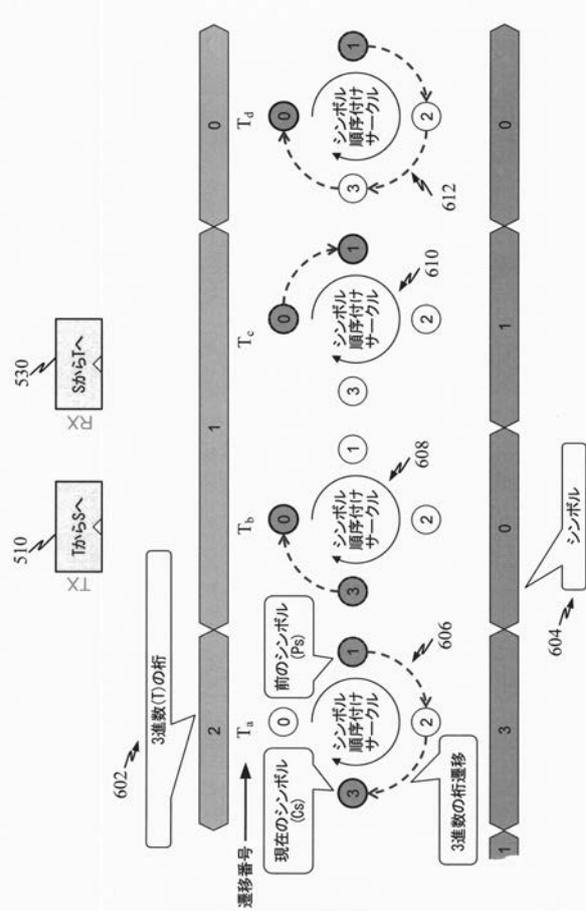
【図4】



【図5】

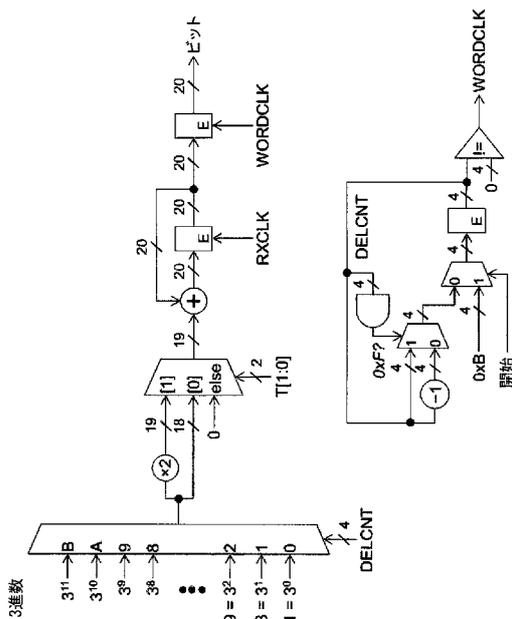


【図6】



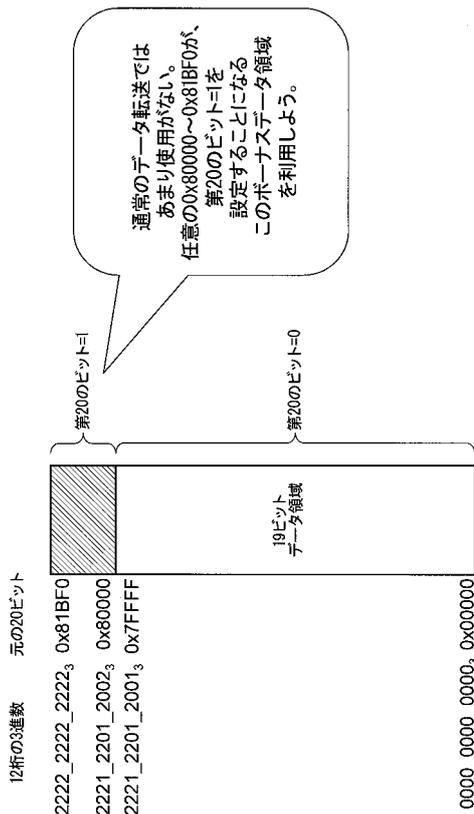
【 図 1 1 】

MSS最優先OCle:12×T-ビット回路



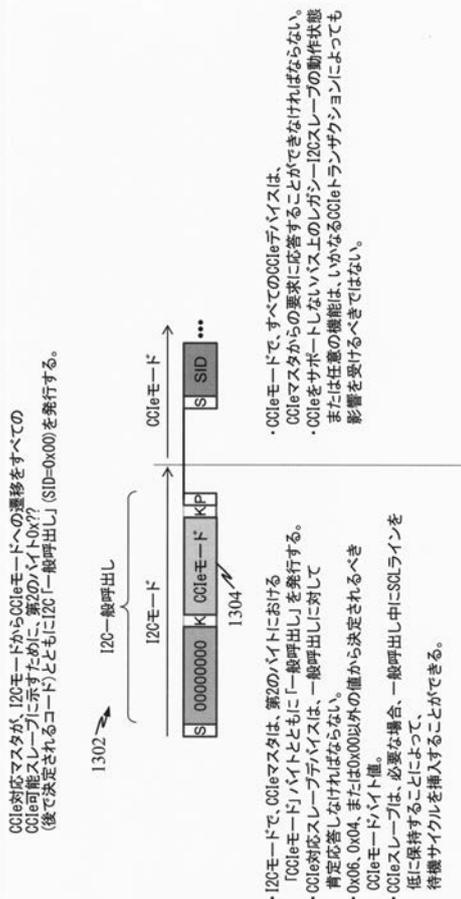
【 図 1 2 】

ビット19(第20のビット)



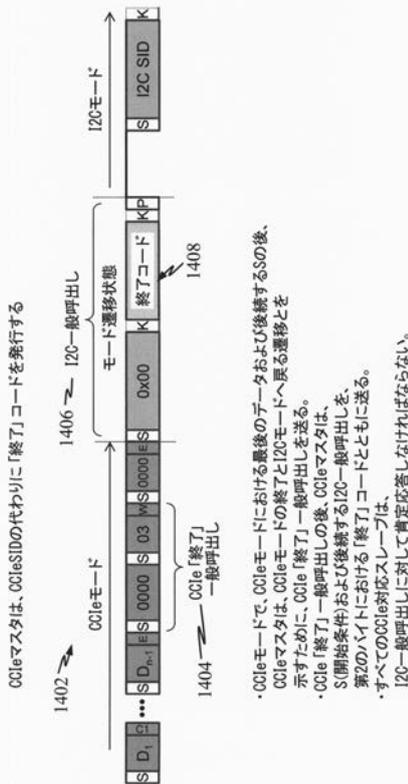
【 図 1 3 】

OCleモードからの例示的なエントリインジケータ



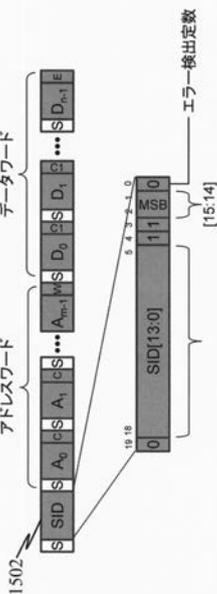
【 図 1 4 】

例示的なOCleモードからの終了インジケータ



【 図 1 5 】

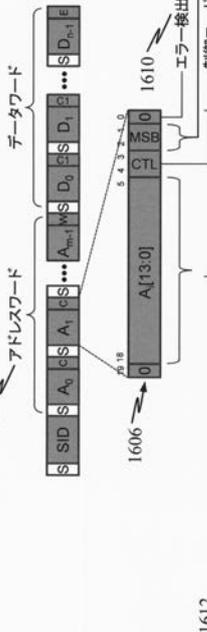
例示的なSIDワードフォーマット



ビットフィールド	値	説明
19	0	
18:5	SID[13:0]	16ビットのスレープID 14-LSB
4:3	11	11に固定され、SIDワードに固有
2:1	SID[15:14]	16ビットのスレープID 2-MSB
0	0	エラー検出定数

【 図 1 6 】

例示的なアドレスワードフォーマット

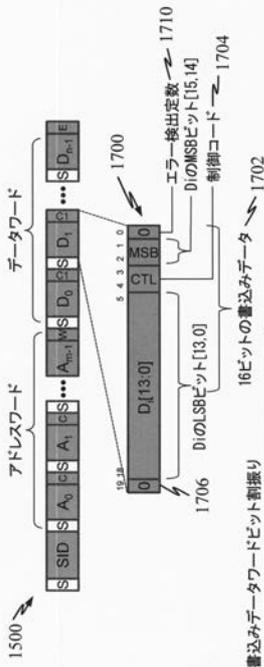


ビットフィールド	値	説明
19	0	
18:5	A[13:0]	16ビットのアドレス14-LSB
4:3	CTL[1:0]	制御コード(下記参照)
2:1	A[15:14]	16ビットのアドレス2-MSB
0	0	エラー検出定数

B[2:1]	シンボル	説明
00	C	次のアドレスワードに続く
01	W	書込み
10	RS	読取り指定ワードが後続する
11	-	禁止

【 図 1 7 】

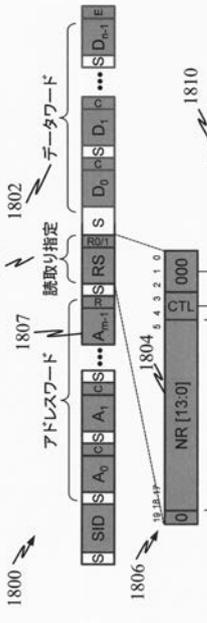
例示的な書込みデータワードフォーマット



ビットフィールド	値	説明
19	0	スベアビット
18:5	D[13:0]	16ビットの書込みデータ14-LSB
4:3	CTL[1:0]	制御コード(下記参照)
2:1	D[15:14]	16ビットの書込みデータ2-MSB
0	0	エラー検出定数

【 図 1 8 】

例示的な読取り指定ワードフォーマット



ビットフィールド	値	説明
19	0	
18:5	NR[13:0]	読み取るべきワードの数
4:3	CTL[1:0]	制御コード(下記参照)
2:0	000	エラー検出定数

CTL[1:0]	シンボル	説明
00	R0	同じアドレスからの読取りワード
01	R1	増分アドレスからの読取りワード
10	-	予約済み(または続く?)
11	-	禁止

- ・RS(読取り指定)ワードは、次の読取りデータワードシークエンスの特性を定義する
- ・NR[13:0]読み取るべきワードの数
- ・他の特性?
- ・NR[13:0]=0x000000のとき、読取りワード数は無制限であり、スレープがCTL=10(E)とともに読取りを終了するまで、読取りが続く。
- ・制御コードが「00」であるとき、同じアドレスからワードを読み取る。
- ・制御コードが「01」であるとき、Aから開始する増分アドレスからワードを読み取る

【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)
 CCleワード=2222_2222_2222_2222 (0x81BF0) 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_2222	81b50	1000_0001_1011_1110_0000	000
T00*1	3131_3131_3130	2222_2222_2222_2221	81bef	1000_0001_1011_1110_1111	111
T00*3	3131_3131_3132	2222_2222_2222_2220	81bee	1000_0001_1011_1110_1110	110
T01*1	3131_3131_3121	2222_2222_2220_2210	81beb	1000_0001_1011_1110_1011	011
T01*3	3131_3131_3101	2222_2222_2200_2201	81bee9	1000_0001_1011_1110_1001	001
T02*1	3131_3131_3031	2222_2222_2102_2102	81bel	1000_0001_1011_1110_0001	001
T02*3	3131_3131_3231	2222_2222_2102_2102	81bdb	1000_0001_1011_1101_1011	011
T03*1	3131_3131_2131	2222_2222_1022_1022	81bc3	1000_0001_1011_1100_0011	011
T03*3	3131_3131_0131	2222_2222_0122_0122	81bb1	1000_0001_1011_1011_0001	001
T04*1	3131_3130_3131	2222_2221_0222_0222	81b69	1000_0001_1011_0110_1001	001
T04*3	3131_3132_3131	2222_2220_1222_1222	81b53	1000_0001_1011_0011_0011	011
T05*1	3131_3131_3131	2222_2210_2222_2222	81a5b	1000_0001_1010_0101_1011	011
T05*3	3131_3101_3131	2222_2201_2222_2222	81969	1000_0001_1001_1011_1001	001
T06*1	3131_3031_3131	2222_2102_2222_2222	81751	1000_0001_0111_0011_0001	001
T06*3	3131_2131_3131	2222_2012_2222_2222	8154b	1000_0001_0101_0100_1011	011
T07*1	3131_0131_3131	2222_1022_2222_2222	80db3	1000_0000_1101_1011_0011	011
T07*3	3131_3131_3131	2222_0122_2222_2222	80801	1000_0000_1000_0000_0001	001
T08*1	3130_3131_3131	2221_0222_2222_2222	7f139	0111_1111_0001_0011_1001	001
T08*3	3132_3131_3131	2220_1222_2222_2222	7e023	0111_1110_0000_0010_0011	011
T09*1	3121_3131_3131	2210_2222_2222_2222	79cbb	0111_1001_1011_1100_1011	011
T09*3	3101_3131_3131	2201_2222_2222_2222	76889	0111_0110_1000_1000_1001	001
T10*1	3031_3131_3131	2102_2222_2222_2222	69681	0110_1001_1011_1000_0001	001
T10*3	3231_3131_3131	2012_2222_2222_2222	601bb	0110_0000_0001_1011_1011	011
T11*1	2131_3131_3131	1022_2222_2222_2222	39a83	0011_1001_1010_1010_0011	011
T11*3	0131_3131_3131	0122_2222_2222_2222	1cd51	0001_1100_1101_0101_0001	001

【図 29】

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)
 CCleワード=2012_0120_0120_0120 (0x4AD8) 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_0120	4ada8	0100_1010_1101_1010_1000	000
T00*1	2030_2120_3020	1201_2012_0122_0122	4adaa	0100_1010_1101_1010_1010	010
T00*2	2030_2120_3023	1201_2012_0121_0121	4ada9	0100_1010_1101_1010_1001	001
T01*1	2030_2120_3031	1201_2012_0102_0102	4ada4	0100_1010_1101_1010_0100	100
T02*1	2030_2120_3121	1201_2012_0210_0210	4adae	0100_1010_1101_1010_1110	110
T03*1	2030_2120_2021	1201_2012_2220_2220	4ade7	0100_1010_1101_1110_0111	111
T03*2	2030_2120_1021	1201_2012_1020_1020	4adba	0100_1010_1101_1011_0110	010
T04*1	2030_2121_3021	1201_2010_2120_2120	4ad3c	0100_1010_1101_0011_1100	100
T05*1	2030_2130_3021	1201_2021_0120_0120	4ae4a	0100_1010_1101_0100_1010	010
T06*1	2030_2020_3021	1201_2222_0120_0120	4b44d	0100_1011_0100_0100_1101	101
T06*2	2030_2320_3021	1201_2102_0120_0120	4af8e	0100_1010_1111_0000_1110	110
T07*1	2030_3120_3021	1201_0212_0120_0120	4a244	0100_1010_0010_0100_0100	100
T08*1	2031_2120_3021	1202_1012_0120_0120	4bebe	0100_1011_1110_1011_1110	110
T09*1	2020_2120_3021	1222_2012_0120_0120	5610f	0101_0110_0001_0000_1111	111
T09*2	2010_2120_3021	1210_2012_0120_0120	4e0ea	0100_1110_0000_1110_1010	010
T10*1	2130_2120_3021	1021_2012_0120_0120	37a1c	0011_0111_1010_0001_1100	100
T11*1	3030_2120_3021	2101_2012_0120_0120	67afa	0110_0111_1010_1111_1010	010
T11*2	0030_2120_3021	0001_2012_0120_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)
 CCleワード=0120_1201_2012_0120 (0x18F38) 上で、
 単一シンボルエラーが発生するとき

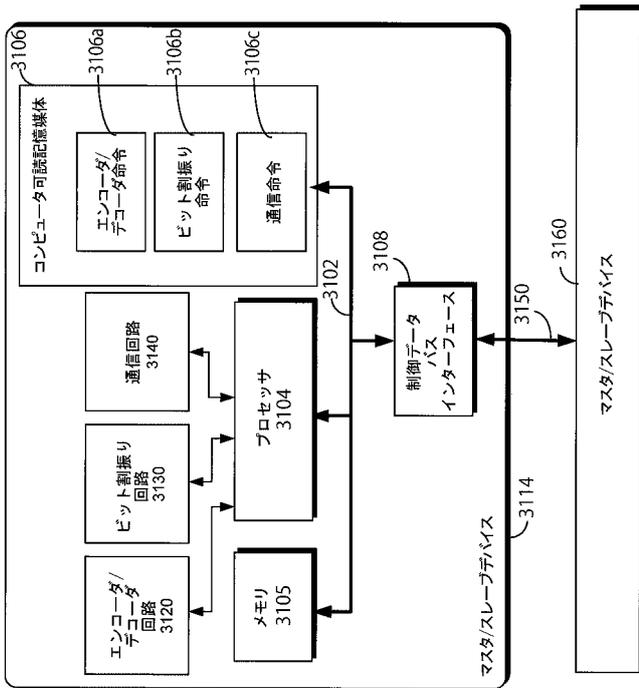
エラー箇所	S11_S0 (シンボル)	T11_T0 (遷移)	Hex	ビット[19:0]	ビット[2:0]
Good	0132_3101_3231	0120_1201_2012_0120	18f38	0001_1000_1111_0011_0000	000
T00*1	0132_3101_3230	0120_1201_2011_0120	18f37	0001_1000_1111_0011_0111	111
T00*3	0132_3101_3232	0120_1201_2010_0120	18f36	0001_1000_1111_0011_0110	110
T01*1	0132_3101_3201	0120_1201_2021_0120	18f3a	0001_1000_1111_0011_1010	010
T02*1	0132_3101_3031	0120_1201_2102_0120	18f3e	0001_1000_1111_0011_1110	110
T02*3	0132_3101_3131	0120_1201_2222_0120	18f4d	0001_1000_1111_0100_1101	101
T03*1	0132_3101_0231	0120_1201_0212_0120	18f14	0001_1000_1111_0001_0100	100
T04*1	0132_3102_3231	0120_1202_0121_0120	18f6e	0001_1000_1111_0110_1110	110
T05*2	0132_3121_3231	0120_1210_2012_0120	18fda	0001_1000_1111_1101_1010	010
T05*3	0132_3131_3231	0120_1222_2012_0120	1916f	0001_1001_0001_0110_1111	111
T06*3	0132_3201_3231	0120_1021_2012_0120	18b6c	0001_1000_1011_0110_1100	100
T07*3	0132_0101_3231	0120_2101_2012_0120	194ea	0001_1001_0100_1110_1010	010
T08*2	0130_3101_3231	0121_0201_2012_0120	1a04e	0001_1010_0000_0100_1110	110
T08*3	0131_3101_3231	0122_2201_2012_0120	1cb05	0001_1100_1011_0000_0101	101
T09*3	0102_3101_3231	0102_1201_2012_0120	128b4	0001_0010_1000_1011_0100	100
T10*3	0232_3101_3231	0210_1201_2012_0120	228fe	0010_0010_1000_1111_1110	110
T11*2	2132_3101_3231	1020_1201_2012_0120	35c8a	0011_0101_1100_1000_1010	010
T11*3	3132_3101_3231	2220_1201_2012_0120	70dd7	0111_1101_1101_1101_0111	111

【図 30】

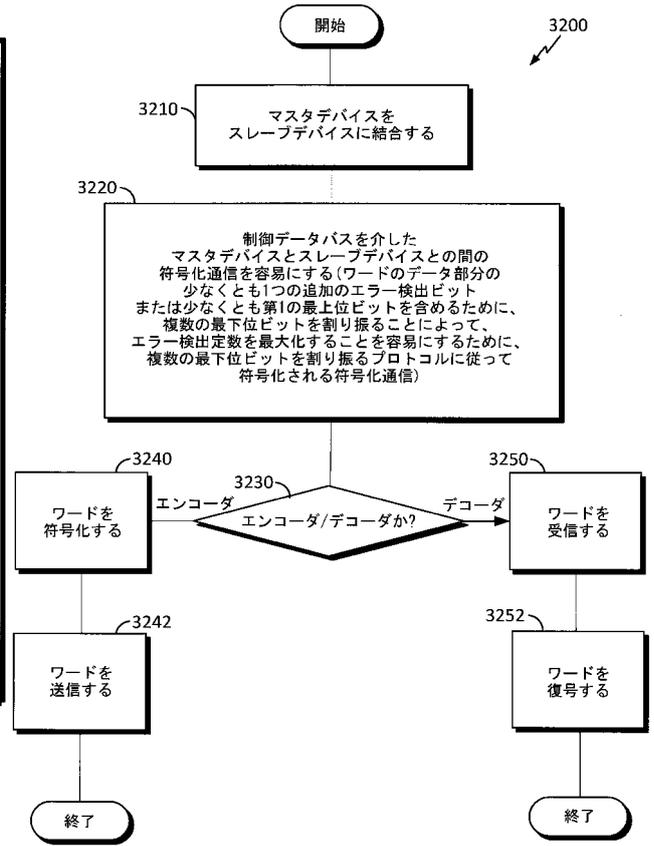
例示的なクロック消失なしのシンボルエラー(シンボルスリップなし)
 CCleワード=2012_0120_1201_0120 (0x5ED08) 上で、
 単一シンボルエラーが発生するとき

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_0120	5ed08	0101_1110_1101_0000_1000	000
T00*2	3231_0132_3103	2012_0120_1200_0120	5ed07	0101_1110_1101_0000_0111	111
T00*3	3231_0132_3102	2012_0120_1202_0120	5ed09	0101_1110_1101_0000_1001	001
T01*2	3231_0132_3121	2012_0120_1210_0120	5ed0a	0101_1110_1101_0000_1010	010
T01*3	3231_0132_3131	2012_0120_1222_0120	5ed0f	0101_1110_1101_0000_1111	111
T02*3	3231_0132_3201	2012_0120_1021_0120	5edcf	0101_1110_1100_1111_1100	100
T03*3	3231_0132_0101	2012_0120_2101_0120	5ed1a	0101_1110_1101_0001_0101	010
T04*2	3231_0130_3101	2012_0121_0201_0120	5ed3e	0101_1110_1101_0011_1110	110
T04*3	3231_0131_3101	2012_0122_2201_0120	5ed51	0101_1110_1101_1100_0101	101
T05*3	3231_0102_3101	2012_0102_1201_0120	5ebc4	0101_1110_1011_1100_0100	100
T06*3	3231_0232_3101	2012_0210_1201_0120	5eeee	0101_1110_1110_1110_1110	110
T07*2	3231_2132_3101	2012_1020_1201_0120	5f2ba	0101_1111_0010_1011_1010	010
T07*3	3231_3132_3101	2012_2220_1201_0120	600f7	0110_0000_0000_1111_0111	111
T08*3	3232_0132_3101	2010_2120_1201_0120	5cadc	0101_1100_1010_1101_1100	010
T09*3	3201_0132_3101	2021_0120_1201_0120	6204a	0110_0010_0000_0100_1010	010
T10*2	3031_0132_3101	2102_0120_1201_0120	686ce	0110_1000_0110_1100_1110	110
T10*3	3131_0132_3101	2222_0120_1201_0120	8073d	1000_0000_0111_0011_1101	101
T11*3	0231_0132_3101	0212_0120_1201_0120	25264	0010_0101_0010_0110_0100	100

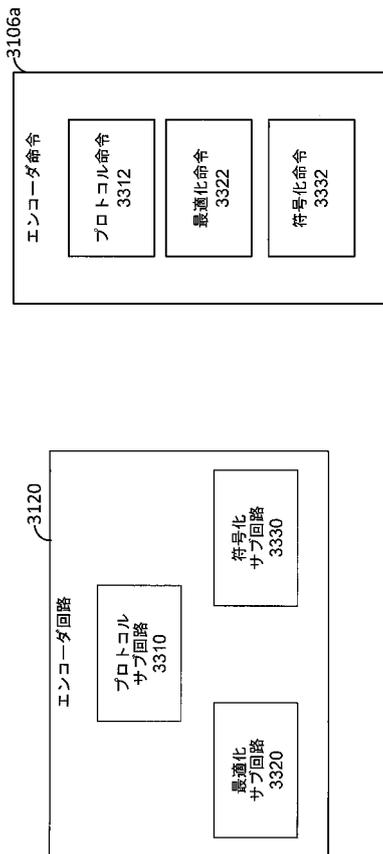
【図 3 1】



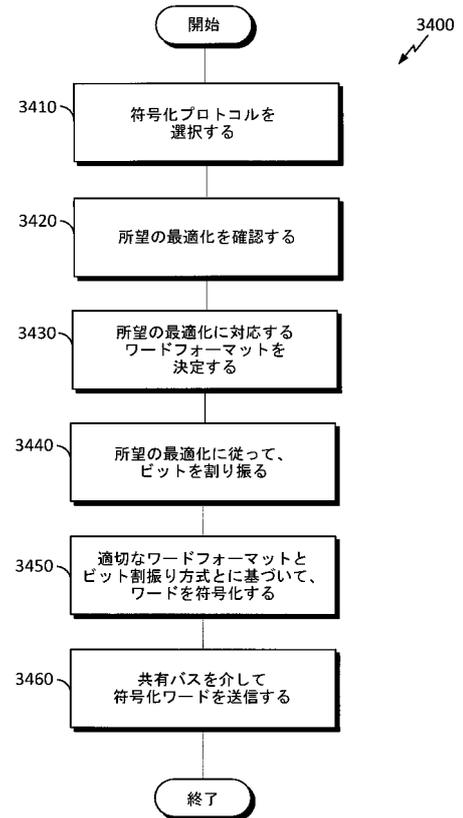
【図 3 2】



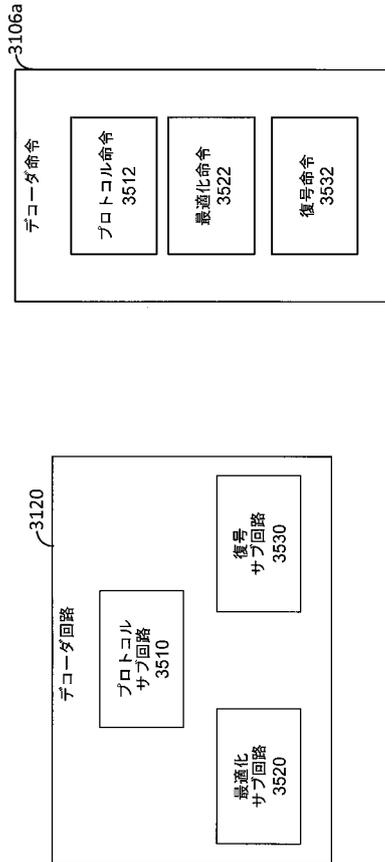
【図 3 3】



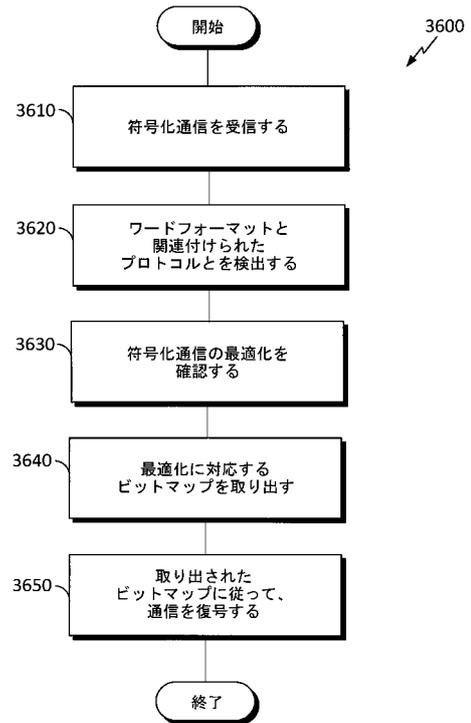
【図 3 4】



【 図 3 5 】



【 図 3 6 】



【 手続補正書 】

【 提出日 】平成28年9月1日 (2016.9.1)

【 手続補正 1 】

【 補正対象書類名 】特許請求の範囲

【 補正対象項目名 】全文

【 補正方法 】変更

【 補正の内容 】

【 特許請求の範囲 】

【 請求項 1 】

2本以上の線を有する制御データバスに結合されたプロセッサを備え、

前記プロセッサが、前記制御データバスを介したマスタデバイスとスレーブデバイスとの間のワードの符号化通信を容易にするように構成され、前記ワードが、桁を備える遷移番号に変換され、各桁が、前記制御データバス上で送信される現在のシンボルの値に基づいて、前記制御データバス上で送信されるべきシンボルのシーケンス内の次のシンボルを取得するために使用され、

前記符号化通信が、エラー検出定数を最大化することを容易にするために、前記符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化され、前記プロトコルが、前記ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、前記複数の最下位ビットを割り振り、

前記エラー検出定数が既知の固定値を有し、

前記シンボルのシーケンス内のシンボルのいずれかの送信中に導入されたエラーが、前記エラー検出定数の値の変化を生じる、デバイス。

【 請求項 2 】

前記制御データバスが2ラインバスである、請求項1に記載のデバイス。

【請求項 3】

前記プロトコルがカメラ制御インターフェース拡張(CCIe)プロトコルである、請求項1に記載のデバイス。

【請求項 4】

前記デバイスが、

ビット割振り方式に従ってビットを割り振るように構成されたビット割振り回路であって、前記ビット割振り方式が、前記符号化通信の前記複数の最下位ビットを割り振る、ビット割振り回路と、

ワードの符号化を容易にするように構成されたエンコーダ回路であって、

前記プロトコルに関連付けられた前記ワードのワードフォーマットを決定するように構成されたプロトコルサブ回路と、

前記符号化通信を生成するために、前記ワードフォーマットと前記ビット割振り方式とに従って、ワードを符号化するように構成された符号化サブ回路と

を備えるエンコーダ回路と、

前記制御データバスを介して前記符号化通信を送信するように構成された通信回路とをさらに備える、請求項1に記載のデバイス。

【請求項 5】

前記符号化サブ回路が、シンボルにトランスコーディングされた、符号化された3進数として、ワードを符号化するように構成される、請求項4に記載のデバイス。

【請求項 6】

前記エンコーダ回路が、前記ワードフォーマットと前記ビット割振り方式とを介して実施するための最適化を確認するように構成された最適化サブ回路をさらに備える、請求項4に記載のデバイス。

【請求項 7】

前記最適化サブ回路が、第1のビット割振り方式を有するエラー検出最適化に従うワードの符号化と、第2のビット割振り方式を有するデータ最適化に従うワードの符号化との間で切り替えることを容易にするように構成される、請求項6に記載のデバイス。

【請求項 8】

前記符号化サブ回路が、前記複数の最下位ビットが固定数の3ビットを備えるデータ最適化に従って、ワードを符号化するように構成され、前記ビット割振り回路が、最下位ビットをエラー検出のために、第2の最下位ビットを前記ワードの前記データ部分の前記第1の最上位ビットのために、および第3の最下位ビットを前記ワードの前記データ部分の第2の最上位ビットのために割り振ることによって、前記データ最適化を容易にするように構成される、請求項6に記載のデバイス。

【請求項 9】

前記符号化サブ回路が、前記複数の最下位ビットが固定数の3ビットを備えるエラー検出最適化に従って、ワードを符号化するように構成され、前記ビット割振り回路が、最下位ビット、第2の最下位ビット、および第3の最下位ビットの各々をエラー検出のために割り振ることによって、前記エラー検出最適化を容易にするように構成される、請求項6に記載のデバイス。

【請求項 10】

前記制御データバスを介して前記符号化通信を受信するように構成された通信回路と、前記符号化通信の復号を容易にするように構成されたデコーダ回路とをさらに備える、請求項1に記載のデバイス。

【請求項 11】

前記デコーダ回路が、

前記プロトコルに関連付けられた前記ワードのワードフォーマットを検出するように構成されたプロトコルサブ回路と、

前記符号化通信の最適化と、前記最適化に対応するビット割振り方式とを確認するように構成された最適化サブ回路と、

前記ワードフォーマットと前記ビット割振り方式とに従って、前記符号化通信を復号するように構成された復号サブ回路と
を備える、請求項10に記載のデバイス。

【請求項12】

マスタデバイスをスレーブデバイスに結合するステップと、
制御データバスを介した前記マスタデバイスと前記スレーブデバイスとの間のワードの符号化通信を容易にするステップであって、前記ワードが、桁を備える遷移番号に変換され、各桁が、前記制御データバス上で送信される現在のシンボルの値に基づいて、前記制御データバス上で送信されるべきシンボルのシーケンス内の次のシンボルを取得するために使用されるステップと
を含み、

前記符号化通信が、エラー検出定数を最大化することを容易にするために、前記符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化され、前記プロトコルが、前記ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、前記複数の最下位ビットを割り振り、
前記エラー検出定数が既知の固定値を有し、
前記シンボルのシーケンス内のシンボルのいずれかの送信中に導入されたエラーが、前記エラー検出定数の値の変化を生じる、方法。

【請求項13】

前記制御データバスが2ラインバスである、請求項12に記載の方法。

【請求項14】

前記プロトコルがカメラ制御インターフェース拡張(CCIe)プロトコルである、請求項12に記載の方法。

【請求項15】

前記プロトコルに関連付けられた前記ワードのワードフォーマットを決定するステップと、

ビット割振り方式に従ってビットを割り振るステップであって、前記ビット割振り方式が、前記符号化通信の前記複数の最下位ビットを割り振るステップと、

前記符号化通信を生成するために、前記ワードフォーマットと前記ビット割振り方式とに従って、前記ワードを符号化するステップと、

前記制御データバスを介して前記符号化通信を送信するステップと
をさらに含む、請求項12に記載の方法。

【請求項16】

前記符号化するステップが、シンボルにトランスコーディングされた、符号化された3進数として、ワードを符号化するステップを含む、請求項15に記載の方法。

【請求項17】

前記ワードフォーマットと前記ビット割振り方式とを介して実施するための最適化を確認するステップをさらに備える、請求項15に記載の方法。

【請求項18】

第1のビット割振り方式を有するエラー検出最適化に従うワードの符号化と、第2のビット割振り方式を有するデータ最適化に従うワードの符号化との間で切り替えるステップをさらに含む、請求項17に記載の方法。

【請求項19】

前記符号化するステップが、前記複数の最下位ビットが固定数の3ビットを備えるデータ最適化に従って、ワードを符号化するステップを含み、前記割り振るステップが、最下位ビットをエラー検出のために、第2の最下位ビットを前記ワードの前記データ部分の前記第1の最上位ビットのために、および第3の最下位ビットを前記ワードの前記データ部分の第2の最上位ビットのために割り振ることによって、前記データ最適化を容易にするステップを含む、請求項17に記載の方法。

【請求項20】

前記符号化するステップが、前記複数の最下位ビットが固定数の3ビットを備えるエラー検出最適化に従って、ワードを符号化するステップを含み、前記割り振るステップが、最下位ビット、第2の最下位ビット、および第3の最下位ビットの各々をエラー検出のために割り振ることによって、前記エラー検出最適化を容易にするステップを含む、請求項17に記載の方法。

【請求項21】

前記制御データバスを介して前記符号化通信を受信するステップと、
前記符号化通信を復号するステップと
をさらに含む、請求項12に記載の方法。

【請求項22】

前記プロトコルに関連付けられた前記ワードのワードフォーマットを検出するステップと、

前記符号化通信の最適化と、前記最適化に対応するビット割り振り方式とを確認するステップと、

前記ワードフォーマットと前記ビット割り振り方式とに従って、前記符号化通信を復号するステップと

をさらに含む、請求項21に記載の方法。

【請求項23】

マスタデバイスをスレーブデバイスに結合するための手段と、

制御データバスを介した前記マスタデバイスと前記スレーブデバイスとの間のワードの符号化通信を容易にするための手段であって、前記ワードが、桁を備える遷移番号に変換され、各桁が、前記制御データバス上で送信される現在のシンボルの値に基づいて、前記制御データバス上で送信されるべきシンボルのシーケンス内の次のシンボルを取得するために使用される手段と

を備え、

前記符号化通信が、エラー検出定数を最大化することを容易にするために、前記符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化され、前記プロトコルが、前記ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、前記複数の最下位ビットを割り振り、

前記エラー検出定数が既知の固定値を有し、

前記シンボルのシーケンス内のシンボルのいずれかの送信中に導入されたエラーが、前記エラー検出定数の値の変化を生じる、デバイス。

【請求項24】

前記プロトコルに関連付けられた前記ワードのワードフォーマットを決定するための手段と、

ビット割り振り方式に従ってビットを割り振るための手段であって、前記ビット割り振り方式が、前記複数の最下位ビットを割り振る手段と、

前記符号化通信を生成するために、前記ワードフォーマットと前記ビット割り振り方式とに従って、前記ワードを符号化するための手段と、

前記制御データバスを介して前記符号化通信を送信するための手段と
をさらに備える、請求項23に記載のデバイス。

【請求項25】

前記ワードフォーマットと前記ビット割り振り方式とを介して実施するための最適化を確認するための手段をさらに備える、請求項24に記載のデバイス。

【請求項26】

第1のビット割り振り方式を有するエラー検出最適化に従うワードの符号化と、第2のビット割り振り方式を有するデータ最適化に従うワードの符号化との間で切り替えるための手段をさらに備える、請求項25に記載のデバイス。

【請求項27】

1つまたは複数の命令を記憶している非一時的機械可読記憶媒体であって、前記1つまた

は複数の命令が、少なくとも1つのプロセッサによって実行されたときに、前記少なくとも1つのプロセッサに、

マスタデバイスをスレーブデバイスに結合すること、および

制御データバスを介した前記マスタデバイスと前記スレーブデバイスとの間のワードの符号化通信を容易にすることであって、前記ワードが、桁を備える遷移番号に変換され、各桁が、前記制御データバス上で送信される現在のシンボルの値に基づいて、前記制御データバス上で送信されるべきシンボルのシーケンス内の次のシンボルを取得するために使用されること
を行わせ、

前記符号化通信が、エラー検出定数を最大化することを容易にするために、前記符号化通信の複数の最下位ビットを割り振るプロトコルに従って符号化され、前記プロトコルが、前記ワードのデータ部分の少なくとも1つの追加のエラー検出ビットまたは少なくとも第1の最上位ビットを含めるために、前記複数の最下位ビットを割り振り、

前記エラー検出定数が既知の固定値を有し、

前記シンボルのシーケンス内のシンボルのいずれかの送信中に導入されたエラーが、前記エラー検出定数の値の変化を生じる、非一時的機械可読記憶媒体。

【請求項 28】

前記1つまたは複数の命令が、前記少なくとも1つのプロセッサによって実行されたときに、前記少なくとも1つのプロセッサに、

前記プロトコルに関連付けられた前記ワードのワードフォーマットを決定すること、

ビット割り振り方式に従ってビットを割り振ることであって、前記ビット割り振り方式が、前記複数の最下位ビットを割り振ること、

前記符号化通信を生成するために、前記ワードフォーマットと前記ビット割り振り方式とに従って、前記ワードを符号化すること、および

前記制御データバスを介して前記符号化通信を送信すること

を行わせる命令をさらに備える、請求項27に記載の非一時的機械可読記憶媒体。

【請求項 29】

前記1つまたは複数の命令が、前記少なくとも1つのプロセッサによって実行されたときに、前記少なくとも1つのプロセッサに、

前記複数の最下位ビットが固定数の3ビットを備えるデータ最適化に従って、ワードを符号化すること、ならびに

最下位ビットをエラー検出のために、第2の最下位ビットを前記ワードの前記データ部分の前記第1の最上位ビットのために、および第3の最下位ビットを前記ワードの前記データ部分の第2の最上位ビットのために割り振ることによって、前記データ最適化を容易にすること

を行わせる命令をさらに備える、請求項28に記載の非一時的機械可読記憶媒体。

【請求項 30】

前記1つまたは複数の命令が、前記少なくとも1つのプロセッサによって実行されたときに、前記少なくとも1つのプロセッサに、

前記複数の最下位ビットが固定数の3ビットを備えるエラー検出最適化に従って、ワードを符号化すること、ならびに

最下位ビット、第2の最下位ビット、および第3の最下位ビットの各々をエラー検出のために割り振ることによって、前記エラー検出最適化を容易にすること

を行わせる命令をさらに備える、請求項28に記載の非一時的機械可読記憶媒体。

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No PCT/US2015/018202

A. CLASSIFICATION OF SUBJECT MATTER INV. H03M13/09 ADD.		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) H03M		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	"I2C - Inter-IC Communications, Lectures 28, October 26-29, 2012", internet article, 29 October 2012 (2012-10-29), XP002740959, Retrieved from the Internet: URL:http://ece.uidaho.edu/ee/classes/ECE34 0/Lecture_Notes/L27/I2C.pdf [retrieved on 2015-06-16] the whole document page 2 ----- -/--	1-30
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 16 June 2015		Date of mailing of the international search report 26/06/2015
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Rydyger, Kay

INTERNATIONAL SEARCH REPORT

International application No PCT/US2015/018202

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>Steve Corrigan: "Introduction to the Controller Area Network (CAN) - Application ReportSLOA101AAugust 2002, Revised July 2008", internet article, 30 July 2008 (2008-07-30), XP002740955, texas Instruments Retrieved from the Internet: URL:http://www.ti.com/lit/an/sloa101a/sloa101a.pdf [retrieved on 2015-06-16] the whole document paragraph [3.1.1]</p> <p style="text-align: center;">-----</p>	1-30
X	<p>"Tradeoffs when considering SPI or I2C?", internet article, 1 April 2012 (2012-04-01), XP002735900, Retrieved from the Internet: URL:http://electronics.stackexchange.com/questions/29037/tradeoffs-when-considering-spi-or-i2c [retrieved on 2015-02-12] the whole document</p> <p style="text-align: center;">-----</p>	1-27
X	<p>I-AN CHEN ET AL: "An error-correction scheme with Reed-Solomon codec for CAN bus transmission", INTELLIGENT SIGNAL PROCESSING AND COMMUNICATIONS SYSTEMS (ISPACS), 2011 INTERNATIONAL SYMPOSIUM ON, IEEE, 7 December 2011 (2011-12-07), pages 1-5, XP032114635, DOI: 10.1109/ISPACS.2011.6146059 ISBN: 978-1-4577-2165-6 the whole document</p> <p style="text-align: center;">-----</p>	1-27
X	<p>US 7 707 349 B1 (KEITHLEY DOUGLAS G [US]) 27 April 2010 (2010-04-27) figure 4 the whole document</p> <p style="text-align: center;">-----</p>	1-27

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2015/018202

Patent document cited in search report	Publication date	Patent family member(a)	Publication date	
US 7707349	B1	27-04-2010	US 7707349 B1	27-04-2010
			US 8046518 B1	25-10-2011

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US