

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6362277号
(P6362277)

(45) 発行日 平成30年7月25日(2018.7.25)

(24) 登録日 平成30年7月6日(2018.7.6)

(51) Int. Cl. F I
H04L 7/04 (2006.01) H04L 7/04 100

請求項の数 27 (全 178 頁)

| | | | |
|---------------|-------------------------------|-----------|---------------------|
| (21) 出願番号 | 特願2015-514293 (P2015-514293) | (73) 特許権者 | 500043574 |
| (86) (22) 出願日 | 平成25年3月14日 (2013.3.14) | | ブラックベリー リミテッド |
| (65) 公表番号 | 特表2015-525507 (P2015-525507A) | | BlackBerry Limited |
| (43) 公表日 | 平成27年9月3日 (2015.9.3) | | カナダ国 エヌ2ケー Oエー7 オンタ |
| (86) 国際出願番号 | PCT/CA2013/000232 | | リオ, ウォータールー, ユニバーシテ |
| (87) 国際公開番号 | W02013/177665 | | ィ アベニュー イースト 2200 |
| (87) 国際公開日 | 平成25年12月5日 (2013.12.5) | | 2200 University Ave |
| 審査請求日 | 平成27年1月27日 (2015.1.27) | | nue East, Waterloo |
| 審判番号 | 不服2017-569 (P2017-569/J1) | (74) 代理人 | ON N2K OA7, Canada |
| 審判請求日 | 平成29年1月16日 (2017.1.16) | | 100107489 |
| (31) 優先権主張番号 | 61/654, 287 | | 弁理士 大塩 竹志 |
| (32) 優先日 | 平成24年6月1日 (2012.6.1) | | |
| (33) 優先権主張国 | 米国 (US) | | |
| (31) 優先権主張番号 | 61/714, 156 | | |
| (32) 優先日 | 平成24年10月15日 (2012.10.15) | | |
| (33) 優先権主張国 | 米国 (US) | | |

最終頁に続く

(54) 【発明の名称】 マルチフォーマットオーディオシステムにおけるロック保証のための確率的方法に基づく汎用同期エンジン

(57) 【特許請求の範囲】

【請求項1】

第1の動作モードおよび第2の動作モードを有する統一バス通信プロトコルを使用して通信するマスタデバイスにスレーブデバイスを同期化する方法であって、

前記統一バス通信プロトコルに対して前記第1の動作モードを仮定することと、

前記第1の動作モードに従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することであって、前記第1の動作モードは、伝送されている同期化ビット間に第1の一定の間隙を有する、ことと、

前記動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、

前記第1の動作モードを仮定して同期化が取得されない場合、前記統一バス通信プロトコルに対して前記第2の動作モードを仮定し、前記第2の動作モードに従って、前記伝送されているデータ上で前記検索することおよび取得することを実行することであって、前記第2の動作モードは、伝送されている同期化ビット間に第2の一定の間隙を有し、前記第2の一定の間隙は、前記第1の一定の間隙とは異なる、ことと

を含み、

前記同期化パターンは、一定同期部分と、動的同期部分とを備え、前記検索することは、前記一定同期部分を検索することを含む、方法。

【請求項2】

前記一定同期部分は、前記第1の動作モードおよび前記第2の動作モードについて異な

る、請求項 1 に記載の方法。

【請求項 3】

前記動的同期部分は、2つの連続する同期化パターンの前記動的同期部分が異なるように、決定論的方法によって生成される、請求項 2 に記載の方法。

【請求項 4】

前記決定論的方法は、周期的冗長チェックカウンタ(CRC)を採用する、請求項 3 に記載の方法。

【請求項 5】

前記第 1 の動作モードは、ワードモードであり、前記第 2 の動作モードは、ビットストリームモードであり、または前記第 1 の動作モードは、ビットストリームモードであり、前記第 2 の動作モードは、ワードモードである、請求項 1 ~ 4 のいずれか一項に記載の方法。

10

【請求項 6】

ワードモードを前記動作モードとして仮定する場合、前記検索することは、前記同期化パターンが以前に場所を特定されていない限り、最大フレーム長までを含む、種々の開始タイムスロット位置に対して前記伝送されているデータ上で行われる、請求項 1 ~ 5 のいずれか一項に記載の方法。

【請求項 7】

ビットストリームモードを前記動作モードとして仮定する場合、前記検索することは、最大フレーム長までを含む、種々の開始タイムスロット位置に対して前記伝送されているデータ上で行われ、これは、前記同期化パターンが以前に場所を特定されていない限り、フレームチャンネルの最大数について繰り返される、請求項 1 ~ 6 のいずれか一項に記載の方法。

20

【請求項 8】

前記同期化パターンの場所が特定されると、前記検索することおよび取得することは、前記統一バス通信プロトコルによって使用されているフレーム長を決定するために、次の同期化パターンを検索することを含む、請求項 1 ~ 7 のいずれか一項に記載の方法。

【請求項 9】

前記少なくとも 1 つの同期化規則は、前記決定されたフレーム長が正しいことを検証するために、前記検索することおよび取得することを数回行うことを含む、請求項 8 に記載の方法。

30

【請求項 10】

前記検索することおよび取得することは、場所が特定されている一定同期部分に関連付けられた現在の動的同期部分を読み取ることを含み、前記少なくとも 1 つの同期化規則は、前記決定論的方法および前記現在の動的同期部分に基づいて、期待動的同期部分を計算することと、次の一定同期部分の場所を特定することと、誤った同期化の可能性を低減させるために、前記次の一定同期部分に関連付けられた動的同期部分を、前記期待動的同期部分と比較することとを含む、請求項 3 に記載の方法。

【請求項 11】

前記少なくとも 1 つの同期化規則は、誤った同期化の可能性をさらに低減させるために、前記計算すること、場所を特定することおよび比較することを 1 回以上繰り返すことを含む、請求項 10 に記載の方法。

40

【請求項 12】

前記少なくとも 1 つの同期化規則は、バストラフィック上の雑音が多い条件下で同期化の可能性を向上させるために、前記場所を特定することおよび取得することにおいてランダム成分を追加することを含む、請求項 1 ~ 11 のいずれか一項に記載の方法。

【請求項 13】

前記ランダム成分は、ある期間にわたって前記バストラフィックのパリティを読み取ることと、前記現在場所が特定されている同期化パターンが真のパターンであるか、または偽のパターンであるかを決定することに役立つように、前記パリティ情報を使用すること

50

とを含む、請求項 1 2 に記載の方法。

【請求項 1 4】

前記ランダム成分は、前記同期化パターンが 1 つ以上のエラーの下で場所を特定されている場合に使用される、請求項 1 2 に記載の方法。

【請求項 1 5】

前記第 1 の動作モードは、ビットストリームモードであり、クロックゲーティングは、前記同期化パターンのビットが存在すると仮定されないフレームチャネルからのデータをスキップするために使用される、請求項 1 ~ 1 4 のいずれか一項に記載の方法。

【請求項 1 6】

前記方法は、クロックゲーティング、および、ワードモードおよびビットストリーム動作モードの両方の下で前記同期化パターンを検索するための共通検索構造を採用する、請求項 1 ~ 1 4 のいずれか一項に記載の方法。

10

【請求項 1 7】

前記スレーブデバイスが同期化から外れた場合、前記スレーブデバイスは、以前の成功した同期化試行中に決定されたフレーム長およびフレーム形式を仮定し、同期化を再獲得しようとする、請求項 1 ~ 1 6 のいずれか一項に記載の方法。

【請求項 1 8】

前記次の一定同期部分に関連付けられた前記動的同期部分が、前記期待動的同期部分に合致しない場合、前記方法は、前記期待動的同期部分に合致する動的同期部分を伴う一定同期部分を検索し続けることをさらに含む、請求項 1 0 に記載の方法。

20

【請求項 1 9】

前記期待動的同期部分に合致する前記動的同期部分の場所を特定するために一定同期部分が検索されて場所を特定される数に制限が課せられる、請求項 1 8 に記載の方法。

【請求項 2 0】

前記合致を見出すことなく前記制限に達する場合、前記方法は、有効な同期部分として前記場所が特定された一定同期部分のうちのいずれかを選択することをさらに含む、請求項 1 9 に記載の方法。

【請求項 2 1】

同期化が達成された後、前記方法は、以前の場所が特定された同期化パターンからの決定されたフレーム長の距離の近傍にある、前記伝送されているデータの中の位置を検討することによって、同期化を維持することをさらに含む、請求項 1 ~ 2 0 のいずれか一項に記載の方法。

30

【請求項 2 2】

前記方法は、前記検索が最大フレーム長を超えるまで、期待動的同期部分と実際の動的同期部分との間の合致を見出さないことにかかわらず、前記現在の場所が特定された一定同期部分を維持することをさらに含み、前記検索が最大フレーム長を超えた時点で、現在の場所が特定された一定同期部分が破棄され、次の一定同期部分が前記現在の場所が特定された一定同期部分として使用されることを確実にし、前記合致に対する前記検索は、継続される、請求項 1 0 に記載の方法。

【請求項 2 3】

前記方法は、最大データ長内に位置する全ての有効な一定同期部分に対応する、前記動的同期部分の表を生成することと、実際のフレーム長を決定するために、連続位置の間の分離における反復に対する期待動的同期部分に合致する前記動的同期部分の位置を分析することと、同期化を取得するために前記実際のフレーム長を使用することとをさらに含む、請求項 1 に記載の方法。

40

【請求項 2 4】

第 1 の動作モードおよび第 2 の動作モードを有する統一バス通信プロトコルに従って通信する電子デバイスであって、前記デバイスは、
信号を送受信するためのインターフェースと、
前記インターフェースに結合されている多重化および同期エンジンと

50

を備え、

前記多重化および同期エンジンは、前記統一バス通信プロトコルに対して前記第1の動作モードを仮定することと、前記第1の動作モードに従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することであって、前記第1の動作モードは、伝送されている同期化ビット間に第1の一定の間隙を有する、ことと、前記動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、前記第1の動作モードを仮定して同期化が取得されない場合、前記統一バス通信プロトコルに対して前記第2の動作モードを仮定し、前記第2の動作モードに従って、前記伝送されているデータ上で前記検索することおよび取得することを実行することであって、前記第2の動作モードは、伝送されている同期化ビット間に第2の一定の間隙を有し、前記第2の一定の間隙は、前記第1の一定の間隙とは異なる、こととによって、前記電子デバイスを第2の電子デバイスと同期化するように構成され、

10

前記電子デバイスは、スレーブデバイスであり、前記第2の電子デバイスは、マスタデバイスであり、

前記同期化パターンは、一定同期部分と、動的同期部分とを備え、前記検索することは、前記一定同期部分を検索することを含む、デバイス。

【請求項25】

前記多重化および同期エンジンは、請求項1～23のいずれか一項に記載の方法を行うようにさらに構成されている、請求項24に記載の電子デバイス。

20

【請求項26】

コンピュータプログラムを記録したコンピュータ読み取り可能な記録媒体であって、前記コンピュータプログラムは、第1の動作モードおよび第2の動作モードを有する統一バス通信プロトコルを使用して通信するマスタデバイスにスレーブデバイスを同期化する方法を実装するようにマイクロプロセッサを適合させるための、前記スレーブデバイスの前記マイクロプロセッサ上で実行可能な複数の命令を備え、前記方法は、

前記統一バス通信プロトコルに対して前記第1の動作モードを仮定することと、

前記第1の動作モードに従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することであって、前記第1の動作モードは、伝送されている同期化ビット間に第1の一定の間隙を有する、ことと、

30

前記動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、

前記第1の動作モードを仮定して同期化が取得されない場合、前記統一バス通信プロトコルに対して前記第2の動作モードを仮定し、前記第2の動作モードに従って、前記伝送されているデータ上で前記検索することおよび取得することを実行することであって、前記第2の動作モードは、伝送されている同期化ビット間に第2の一定の間隙を有し、前記第2の一定の間隙は、前記第1の一定の間隙とは異なる、ことと

を含み、

前記同期化パターンは、一定同期部分と、動的同期部分とを備え、前記検索することは、前記一定同期部分を検索することを含む、コンピュータ読み取り可能な記録媒体。

40

【請求項27】

前記方法は、請求項1～23のいずれか一項に従ってさらに定義される、請求項26に記載のコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本明細書で説明される種々の実施形態は、概して、マルチフォーマットまたは統一通信プロトコルに同期化する方法、デバイス、およびシステムに関する。

【背景技術】

【0002】

50

コンピュータまたはプロセッサアーキテクチャでは、バスは、電子デバイス内のデバイス間でデータを転送するか、または電子デバイス間でデータを転送するサブシステムである。バスアーキテクチャはまた、互と通信し得るデバイスの各セットの間で別個の接続を有するよりもむしろ、複数のデバイスのための共通データ信号伝達経路で使用される。換言すると、バス構造は、1つ以上のスレーブデバイスが1つ以上のマスタデバイスと通信することを可能にするために使用されることができる。

【発明の概要】

【課題を解決するための手段】

【0003】

一側面では、本明細書で説明される少なくとも1つの実施形態は、第1および第2の動作モードを有する統一バス通信プロトコルを使用して通信するマスタデバイスにスレーブデバイスを同期化する方法を提供する。本方法は、統一バス通信プロトコルに対して第1の動作モードを仮定することと、第1の動作モードに従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することと、動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、第1の動作モードを仮定して同期化が取得されない場合、統一バス通信プロトコルのための第2の動作モードを仮定し、第2の動作モードに従って、伝送されているデータ上で検索することおよび取得することを実行することを含む。

10

【0004】

少なくとも1つの実施形態では、同期化パターンは、一定同期部分と、動的同期部分とを備え、検索することは、一定同期部分を検索することを含む。

20

【0005】

少なくとも1つの実施形態では、一定同期部分は、第1および第2の動作モードについて異なる。

【0006】

少なくとも1つの実施形態では、動的同期部分は、2つの連続する同期化パターンの動的同期部分が異なるように、決定論的方法によって生成される。

【0007】

少なくとも1つの実施形態では、決定論的方法は、周期的冗長カウンタ(CRC)を採用する。

30

【0008】

少なくとも1つの実施形態では、第1の動作モードは、ワードモードであり、第2の動作モードは、ビットストリームモードであり、または、第1の動作モードは、ビットストリームモードであり、第2の動作モードは、ワードモードである。

【0009】

少なくとも1つの実施形態では、ワードモードを動作モードとして仮定する場合、検索することは、同期化パターンが以前に場所を特定されていない限り、最大フレーム長までを含む、種々の開始タイムスロット位置に対して伝送されているデータ上で行われる。

【0010】

少なくとも1つの実施形態では、ビットストリームモードを動作モードとして仮定する場合、検索することは、最大フレーム長までを含む、種々の開始タイムスロット位置に対して伝送されているデータ上で行われ、これは、同期化パターンが以前に場所を特定されていない限り、フレームチャネルの最大数について繰り返される。

40

【0011】

少なくとも1つの実施形態では、同期化パターンの場所が特定されると、検索することおよび取得することは、統一バス通信プロトコルによって使用されているフレーム長を決定するために、次の同期化パターンを検索することを含む。

【0012】

少なくとも1つの実施形態では、少なくとも1つの同期化規則は、決定されたフレーム長が正しいことを検証するために、検索することおよび取得することを数回行うことを含

50

む。

【0013】

少なくとも1つの実施形態では、検索することおよび取得することは、場所が特定されている一定同期部分に関連付けられた現在の動的同期部分を読み取ることを含み、少なくとも1つの同期化規則は、決定論的方法および現在の動的同期部分に基づいて、期待動的同期部分を計算することと、次の一定同期部分の場所を特定することと、誤った同期化の可能性を低減させるために、次の一定同期部分に関連付けられた動的同期部分を、期待動的同期部分と比較することとを含む。

【0014】

少なくとも1つの実施形態では、少なくとも1つの同期化規則は、誤った同期化の可能性をさらに低減させるために、計算すること、場所を特定すること、および比較することを1回以上繰り返すことを含む。

10

【0015】

少なくとも1つの実施形態では、少なくとも1つの同期化規則は、バストラフィック上の雑音が多い条件下で同期化の可能性を向上させるために、場所を特定すること、および取得することにおいてランダム成分を追加することを含む。

【0016】

少なくとも1つの実施形態では、ランダム成分は、ある期間にわたってバストラフィックのパリティを読み取ることと、現在場所が特定されている同期化パターンが真のパターンであるか、または偽のパターンであるかを決定することに役立つように、パリティ情報を使用することとを含む。

20

【0017】

少なくとも1つの実施形態では、ランダム成分は、同期化パターンが1つ以上のエラーの下で場所を特定されている場合に使用される。

【0018】

少なくとも1つの実施形態では、第1の動作モードは、ビットストリームモードであり、クロックゲーティングは、同期化パターンのビットが存在すると仮定されないフレームチャネルからのデータをスキップするために使用される。

【0019】

少なくとも1つの実施形態では、本方法は、クロックゲーティング、およびワードモードおよびビットストリーム動作モードの両方の下で同期化パターンを検索するための共通検索構造を採用する。

30

【0020】

少なくとも1つの実施形態では、スレーブデバイスが同期化から外れた場合、スレーブデバイスは、以前の成功した同期化試行中に決定されたフレーム長およびフレーム形式を仮定し、同期化を再獲得しようとする。

【0021】

少なくとも1つの実施形態では、次の一定同期部分に関連付けられた動的同期部分が、期待動的同期部分に合致しない場合、本方法はさらに、期待動的同期部分に合致する動的同期部分を伴う一定同期部分を検索し続けることを含む。

40

【0022】

少なくとも1つの実施形態では、期待動的同期部分に合致する動的同期部分の場所を特定するために一定同期部分が検索されて場所を特定される数に制限が課せられる。

【0023】

少なくとも1つの実施形態では、合致を見出すことなく制限に達する場合、方法はさらに、有効な同期部分として場所が特定された一定同期部分のうちのいずれかを選択することを含む。

【0024】

少なくとも1つの実施形態では、同期化が達成された後、本方法はさらに、以前の場所が特定された同期化パターンからの決定されたフレーム長の距離の近傍にある、伝送され

50

ているデータの中の位置を検討することによって、同期化を維持することを含む。

【0025】

少なくとも1つの実施形態では、本方法はさらに、検索が最大フレーム長を超えるまで、期待動的同期部分と実際の動的同期部分との間の合致を見出さないことにかかわらず、現在の場所が特定された一定同期部分を維持することを含み、検索が最大フレーム長を超えた時点で、現在の場所が特定された一定同期部分が破棄され、次の一定同期部分が現在の場所が特定された一定同期部分として使用され、合致の検索は、継続される。

【0026】

少なくとも1つの実施形態では、本方法はさらに、最大データ長内に位置する全ての有効な一定同期部分に対応する、動的同期部分の表を生成することと、実際のフレーム長を決定するために、連続位置の間の分離における反復のために、期待動的同期部分に合致する動的同期部分の位置を分析することと、同期化を取得するために実際のフレーム長を使用することを含む。

【0027】

別の側面では、本明細書で説明される少なくとも1つの実施形態は、第1および第2の動作モードを有する統一バス通信プロトコルを使用して通信するマスタデバイスに同期化する方法を提供する。本方法は、統一バス通信プロトコルに対して第1および第2の動作モードを仮定することと、第1および第2の動作モードに従って伝送されているデータの中の1つ以上の場所で同期化パターンを同時に検索することと、第1および第2の動作モードのうちの1つのための少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、同期化が取得されない場合、統一バス通信プロトコルに従って、伝送されているデータ上で第1および第2の動作モードに従って、検索することおよび取得することを実行することを含む。

【0028】

少なくとも1つの実施形態では、本方法は、種々の方法の実施形態について上記で説明される特徴のうちの1つ以上を組み込み得る。

【0029】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、ビットストリームフレーム形式を使用して通信するマスタデバイスにスレーブデバイスを同期化する方法を提供する。本方法は、ビットストリームフレーム形式に従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することと、動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、同期化が取得されない場合、新たに伝送されているデータ上で検索することおよび取得することを繰り返し実行することを含む、ビットストリームフレーム形式では、同期化パターンは、他の伝送されているデータと一度に1ビットで多重化される。

【0030】

少なくとも1つの実施形態では、本方法は、種々の方法の実施形態について上記で説明される特徴のうちの1つ以上を組み込み得る。

【0031】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、ワードフレーム形式を使用して通信するマスタデバイスにスレーブデバイスを同期化する方法を提供する。本方法は、ワードフレーム形式に従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することと、動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、同期化が取得されない場合、新たに伝送されているデータ上で検索することおよび取得することを繰り返し実行することを含む、ワードフレーム形式では、同期化パターンは、データの全フレームにおいて、連続ビットで伝送される。

【0032】

少なくとも1つの実施形態では、本方法は、種々の方法の実施形態について上記で説明

10

20

30

40

50

される特徴のうちの1つ以上を組み込み得る。

【0033】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、第1および第2の動作モードを有する統一バス通信プロトコルに従って通信する電子デバイスを提供する。本デバイスは、信号を送受信するためのインターフェースと、インターフェースに連結されている多重化および同期エンジンとを備え、多重化および同期エンジンは、統一バス通信プロトコルに対して第1の動作モードを仮定すること、第1の動作モードに従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索すること、動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得すること、および第1の動作モードを仮定して同期化が取得されない場合、統一バス通信プロトコルのための第2の動作モードを仮定し、第2の動作モードに従って、伝送されているデータ上で検索することおよび取得することを実行することによって、電子デバイスを第2の電子デバイスと同期化するように構成され、電子デバイスは、スレーブデバイスであり、第2の電子デバイスは、マスタデバイスである。

10

【0034】

少なくとも1つの実施形態では、本デバイスは、種々の方法の実施形態について上記で説明される特徴のうちの1つ以上を組み込み得る。

【0035】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、第1および第2の動作モードを有する統一バス通信プロトコルに従って通信する電子デバイスを提供する。本デバイスは、信号を送受信するためのインターフェースと、インターフェースに連結されている多重化および同期エンジンとを備え、多重化および同期エンジンは、統一バス通信プロトコルに対して第1および第2の動作モードを仮定すること、第1および第2の動作モードに従って伝送されているデータの中の1つ以上の場所で同期化パターンを同時に検索すること、第1および第2の動作モードのうちの1つのための少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得すること、同期化が取得されない場合、統一バス通信プロトコルに従って、伝送されているデータ上で第1および第2の動作モードに従って、検索することおよび取得することを実行することによって、電子デバイスを第2の電子デバイスと同期化するように構成され、電子デバイスは、スレーブデバイスであり、第2の電子デバイスは、マスタデバイスである。

20

30

【0036】

少なくとも1つの実施形態では、本デバイスは、種々の方法の実施形態について上記で説明される特徴のうちの1つ以上を組み込み得る。

【0037】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、ビットストリームフレーム形式に従って通信する電子デバイスを提供する。本デバイスは、信号を送受信するためのインターフェースと、インターフェースに連結されている多重化および同期エンジンとを備え、多重化および同期エンジンは、ビットストリームフレーム形式に従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索すること、動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得すること、同期化が取得されない場合、新たに伝送されているデータ上で検索することおよび取得することを実行することによって、電子デバイスを第2の電子デバイスと同期化するように構成され、ビットストリームフレーム形式では、同期化パターンは、他の伝送されているデータと一度に1ビットで多重化され、電子デバイスは、スレーブデバイスであり、第2の電子デバイスは、マスタデバイスである。

40

【0038】

少なくとも1つの実施形態では、本デバイスは、種々の方法の実施形態について上記で

50

説明される特徴のうちの1つ以上を組み込み得る。

【0039】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、ワードフレーム形式に従って通信する電子デバイスを提供する。本デバイスは、信号を送受信するためのインターフェースと、インターフェースに連結されている多重化および同期エンジンとを備え、多重化および同期エンジンは、ワードフレーム形式に従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索すること、動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得すること、同期化が取得されない場合、新たに伝送されているデータ上で検索することおよび取得することを繰り返し実行することによって、電子デバイスを第2の電子デバイスと同期化するように構成され、ワードフレーム形式では、同期化パターンは、データの全フレームにおいて、連続ビットで伝送され、電子デバイスは、スレーブデバイスであり、第2の電子デバイスは、マスタデバイスである。

10

【0040】

少なくとも1つの実施形態では、本デバイスは、種々の方法の実施形態について上記で説明される特徴のうちの1つ以上を組み込み得る。

【0041】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、第1および第2の動作モードを有する統一バス通信プロトコルを使用して通信するマスタデバイスにスレーブデバイスを同期化する方法を実装するようにマイクロプロセッサを適合させるための、スレーブデバイスのマイクロプロセッサ上で実行可能な複数の命令を備えているコンピュータ読み取り可能な媒体を提供し、本方法は、統一バス通信プロトコルに対して第1の動作モードを仮定することと、第1の動作モードに従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することと、動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、第1の動作モードを仮定して同期化が取得されない場合、統一バス通信プロトコルのための第2の動作モードを仮定し、第2の動作モードに従って、伝送されているデータ上で検索することおよび取得することを実行することを含む。

20

【0042】

少なくとも1つの実施形態では、コンピュータ読み取り可能な媒体は、種々の方法の実施形態について上記で説明される特徴のうちの1つ以上を組み込み得る。

30

【0043】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、第1および第2の動作モードを有する統一バス通信プロトコルを使用して通信するマスタデバイスにスレーブデバイスを同期化する方法を実装するようにマイクロプロセッサを適合させるための、スレーブデバイスのマイクロプロセッサ上で実行可能な複数の命令を備えているコンピュータ読み取り可能な媒体を提供し、本方法は、統一バス通信プロトコルに対して第1および第2の動作モードを仮定することと、第1および第2の動作モードに従って伝送されているデータの中の1つ以上の場所で同期化パターンを同時に検索することと、第1および第2の動作モードのうちの1つのための少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、同期化が取得されない場合、統一バス通信プロトコルに従って、伝送されているデータ上で第1および第2の動作モードに従って、検索することおよび取得することを実行することを含む。

40

【0044】

少なくとも1つの実施形態では、コンピュータ読み取り可能な媒体は、種々の方法の実施形態について上記で説明される特徴のうちの1つ以上を組み込み得る。

【0045】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、ビットストリームフレーム形式を使用して通信するマスタデバイスにスレーブデバイスを同期化する

50

方法を実装するようにマイクロプロセッサを適合させるための、スレーブデバイスのマイクロプロセッサ上で実行可能な複数の命令を備えているコンピュータ読み取り可能な媒体を提供し、本方法は、ビットストリームフレーム形式に従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することと、動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、同期化が取得されない場合、新たに伝送されているデータ上で検索することおよび取得することを繰り返し実行することとを含み、ビットストリームフレーム形式で、同期化パターンは、他の伝送されているデータと一度に1ビットで多重化される。

【0046】

少なくとも1つの実施形態では、コンピュータ読み取り可能な媒体は、種々の方法の実施形態について上記で説明される特徴のうちの1つ以上を組み込み得る。

【0047】

さらに別の側面では、本明細書で説明される少なくとも1つの実施形態は、ワードフレーム形式を使用して通信するマスタデバイスにスレーブデバイスを同期化する方法を実装するようにマイクロプロセッサを適合させるための、スレーブデバイスのマイクロプロセッサ上で実行可能な複数の命令を備えているコンピュータ読み取り可能な媒体を提供し、本方法は、ワードフレーム形式に従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することと、動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、同期化が取得されない場合、新たに伝送されているデータ上で検索することおよび取得することを繰り返し実行することとを含み、ワードフレーム形式では、同期化パターンは、データの全フレームにおいて、連続ビットで伝送される。

【0048】

少なくとも1つの実施形態では、コンピュータ読み取り可能な媒体は、種々の方法の実施形態について上記で説明される特徴のうちの1つ以上を組み込み得る。

例えば、本願発明は以下の項目を提供する。

(項目1)

第1および第2の動作モードを有する統一バス通信プロトコルを使用して通信するマスタデバイスにスレーブデバイスを同期化する方法であって、

前記統一バス通信プロトコルに対して前記第1の動作モードを仮定することと、

前記第1の動作モードに従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することと、

前記動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、

前記第1の動作モードを仮定して同期化が取得されない場合、前記統一バス通信プロトコルに対して前記第2の動作モードを仮定し、前記第2の動作モードに従って、前記伝送されているデータ上で前記検索すること、および取得することを実行すること

を含む、方法。

(項目2)

前記同期化パターンは、一定同期部分と、動的同期部分とを備え、前記検索することは、前記一定同期部分を検索することを含む、項目1に記載の方法。

(項目3)

前記一定同期部分は、前記第1および第2の動作モードについて異なる、項目1または項目2に記載の方法。

(項目4)

前記動的同期部分は、2つの連続する同期化パターンの前記動的同期部分が異なるように、決定論的方法によって生成される、項目2または項目3に記載の方法。

(項目5)

前記決定論的方法は、周期的冗長カウンタ(CRC)を採用する、項目4に記載の方法

10

20

30

40

50

。

(項目 6)

前記第 1 の動作モードは、ワードモードであり、前記第 2 の動作モードは、ビットストリームモードであり、または前記第 1 の動作モードは、ビットストリームモードであり、前記第 2 の動作モードは、ワードモードである、項目 1 ~ 5 のいずれか一項に記載の方法

。

(項目 7)

ワードモードを前記動作モードとして仮定する場合、前記検索することは、前記同期化パターンが以前に場所を特定されていない限り、最大フレーム長までを含む、種々の開始タイムスロット位置に対して前記伝送されているデータ上で行われる、項目 1 ~ 6 のいずれか一項に記載の方法。

10

(項目 8)

ビットストリームモードを前記動作モードとして仮定する場合、前記検索することは、最大フレーム長までを含む、種々の開始タイムスロット位置に対して前記伝送されているデータ上で行われ、これは、前記同期化パターンが以前に場所を特定されていない限り、フレームチャネルの最大数について繰り返される、項目 1 ~ 7 のいずれか一項に記載の方法。

(項目 9)

前記同期化パターンの場所が特定されると、前記検索すること、および取得することは、前記統一バス通信プロトコルによって使用されているフレーム長を決定するために、次の同期化パターンを検索することを含む、項目 1 ~ 8 のいずれか一項に記載の方法。

20

(項目 10)

前記少なくとも 1 つの同期化規則は、前記決定されたフレーム長が正しいことを検証するために、前記検索すること、および取得することを数回行うことを含む、項目 9 に記載の方法。

(項目 11)

前記検索すること、および取得することは、場所が特定されている一定同期部分に関連付けられた現在の動的同期部分を読み取ることを含み、前記少なくとも 1 つの同期化規則は、前記決定論的方法および前記現在の動的同期部分に基づいて、期待動的同期部分を計算することと、次の一定同期部分の場所を特定することと、誤った同期化の可能性を低減させるために、前記次の一定同期部分に関連付けられた動的同期部分を、前記期待動的同期部分と比較することとを含む、項目 4 に記載の方法。

30

(項目 12)

前記少なくとも 1 つの同期化規則は、誤った同期化の可能性をさらに低減させるために、前記計算すること、場所を特定すること、および比較することを 1 回以上繰り返すことを含む、項目 11 に記載の方法。

(項目 13)

前記少なくとも 1 つの同期化規則は、バストラフィック上の雑音が多い条件下で同期化の可能性を向上させるために、前記場所を特定すること、および取得することにおいてランダム成分を追加することを含む、項目 1 ~ 12 のいずれか一項に記載の方法。

40

(項目 14)

前記ランダム成分は、ある期間にわたって前記バストラフィックのパリティを読み取ることと、前記現在場所が特定されている同期化パターンが真のパターンであるか、または偽のパターンであるかを決定することに役立つように、前記パリティ情報を使用することとを含む、項目 13 に記載の方法。

(項目 15)

前記ランダム成分は、前記同期化パターンが 1 つ以上のエラーの下で場所を特定されている場合に使用される、項目 13 に記載の方法。

(項目 16)

前記第 1 の動作モードは、ビットストリームモードであり、クロックゲーティングは、

50

前記同期化パターンのビットが存在すると仮定されないフレームチャネルからのデータをスキップするために使用される、項目 1 ~ 15 のいずれか一項に記載の方法。

(項目 17)

前記方法は、クロックゲーティング、および、ワードモードおよびビットストリーム動作モードの両方の下で前記同期化パターンを検索するための共通検索構造を採用する、項目 1 ~ 15 のいずれか一項に記載の方法。

(項目 18)

前記スレーブデバイスが同期化から外れた場合、前記スレーブデバイスは、以前の成功した同期化試行中に決定されたフレーム長およびフレーム形式を仮定し、同期化を再獲得しようとする、項目 1 ~ 17 のいずれか一項に記載の方法。

10

(項目 19)

前記次の一定同期部分に関連付けられた前記動的同期部分が、前記期待動的同期部分に合致しない場合、前記方法は、前記期待動的同期部分に合致する動的同期部分を伴う一定同期部分を検索し続けることをさらに含む、項目 11 に記載の方法。

(項目 20)

前記期待動的同期部分に合致する前記動的同期部分の場所を特定するために一定同期部分が検索されて場所を特定される数に制限が課せられる、項目 19 に記載の方法。

(項目 21)

前記合致を見出すことなく前記制限に達する場合、前記方法は、有効な同期部分として前記場所が特定された一定同期部分のうちのいずれかを選択することをさらに含む、項目 20 に記載の方法。

20

(項目 22)

同期化が達成された後、前記方法は、以前の場所が特定された同期化パターンからの決定されたフレーム長の距離の近傍にある、前記伝送されているデータの中の位置を検討することによって、同期化を維持することをさらに含む、項目 1 ~ 21 のいずれか一項に記載の方法。

(項目 23)

前記方法は、前記検索が最大フレーム長を超えるまで、期待動的同期部分と実際の動的同期部分との間の合致を見出さないことにかかわらず、前記現在の場所が特定された一定同期部分を維持することをさらに含み、前記検索が最大フレーム長を超えた時点で、現在の場所が特定された一定同期部分が破棄され、次の一定同期部分が前記現在の場所が特定された一定同期部分として使用され、前記合致に対する前記検索は、継続される、項目 11 に記載の方法。

30

(項目 24)

前記方法は、最大データ長内に位置する全ての有効な一定同期部分に対応する、前記動的同期部分の表を生成することと、実際のフレーム長を決定するために、連続位置の間の分離における反復に対する期待動的同期部分に合致する前記動的同期部分の位置を分析することと、同期化を取得するために前記実際のフレーム長を使用することとをさらに含む、項目 2 に記載の方法。

(項目 25)

第 1 および第 2 の動作モードを有する統一バス通信プロトコルに従って通信する電子デバイスであって、前記デバイスは、

信号を送受信するためのインターフェースと、

前記インターフェースに連結されている多重化および同期エンジンと
を備え、

前記多重化および同期エンジンは、前記統一バス通信プロトコルに対して前記第 1 の動作モードを仮定することと、前記第 1 の動作モードに従って、伝送されているデータの中の 1 つ以上の場所で同期化パターンを検索することと、前記動作モードに対する少なくとも 1 つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、前記第 1 の動作モードを仮定して同期化が取得されない場合、前記

40

50

統一バス通信プロトコルに対して前記第2の動作モードを仮定し、前記第2の動作モードに従って、前記伝送されているデータ上で前記検索すること、および取得することを実行することとによって、前記電子デバイスを第2の電子デバイスと同期化するように構成され、

前記電子デバイスは、スレーブデバイスであり、前記第2の電子デバイスは、マスタデバイスである、デバイス。

(項目26)

前記多重化および同期エンジンは、項目2～24のいずれか一項に記載の方法を行うようにさらに構成されている、項目25に記載の電子デバイス。

(項目27)

第1および第2の動作モードを有する統一バス通信プロトコルを使用して通信するマスタデバイスにスレーブデバイスを同期化する方法を実装するようにマイクロプロセッサを適合させるための、前記スレーブデバイスの前記マイクロプロセッサ上で実行可能な複数の命令を備えているコンピュータ読み取り可能な媒体であって、前記方法は、

前記統一バス通信プロトコルに対して前記第1の動作モードを仮定することと、

前記第1の動作モードに従って、伝送されているデータの中の1つ以上の場所で同期化パターンを検索することと、

前記動作モードに対する少なくとも1つの同期化規則に従って、場所が特定された同期化パターンが検証された場合、同期化を取得することと、

前記第1の動作モードを仮定して同期化が取得されない場合、前記統一バス通信プロトコルに対して前記第2の動作モードを仮定し、前記第2の動作モードに従って、前記伝送されているデータ上で前記検索すること、および取得することを実行することと

を含む、コンピュータ読み取り可能な媒体。

(項目28)

前記方法は、項目2～24のいずれか一項に従ってさらに定義される、項目27に記載のコンピュータ読み取り可能な媒体。

【図面の簡単な説明】

【0049】

本明細書で説明される種々の実施形態のより良い理解のために、およびこれらの種々の実施形態がどのようにして実行に移され得るかをより明確に示すために、一例として、少なくとも1つの例示的実施形態を示す、添付図面が参照されるであろう。

【図1】図1は、可搬性電子デバイスの斜視図である。

【図2】図2は、可搬性電子デバイス的一部分のブロック図である。

【図3a】図3aは、バスシステムの例示的実施形態の概略図である。

【図3b】図3bは、マスタデバイスをバスに連結するためのインターフェース回路の実施例を示す、図3aのバスシステムの概略図である。

【図3c】図3cは、2ワイヤバスシステムの例示的実施形態のブロック図である。

【図4a】図4aは、スレーブデバイスが単一ワイヤバス上に「0110」を書き込む、バス上のトランザクションの例示的タイミング図である。

【図4b】図4bは、2本のワイヤを使用してバスが実装されるときに例示的タイミング図である。

【図5】図5は、同期化ワードのためのフィールドおよびビット割付の例示的実施形態の略図である。

【図6】図6は、ワードモードでのSワードのデータ同期化フィールドへのデータ動作を示す、例示的タイミング図である。

【図7】図7は、異なるコマンド動作に基づいてマスタデバイスおよびスレーブデバイスによって行われ得る、パリティおよび確認計算の実施例である。

【図8】図8は、コマンド動作(PING、READ、およびWRITE)、割り込みマスクビットの値、S0 DELAYビット、講じられるべき措置の種々の組み合わせを示す。

10

20

30

40

50

【図9 a】図9 aは、種々のコマンド動作のためのXコマンドワードの例示的实施形態のための種々のフィールドおよびビット割付を示す。

【図9 b】図9 bは、種々のコマンド動作のためのXコマンドワードの別の例示的实施形態のための種々のフィールドおよびビット割付を示す。

【図10】図10は、種々のコマンド動作のためのYコマンドワードの例示的实施形態のための種々のフィールドおよびビット割付を示す。

【図11 a】図11 aは、Xワードで設定され得る、関数および対応するビット設定の例示的リストを示す。

【図11 b】図11 bは、固定アドレスを使用する2つの端子を使用して、デバイス中の位置情報を符号化する実施例を示し、この構成はまた、レガシーシステムと適合性がある

10

。【図11 c】図11 cは、リングトポロジーに基づいて、固定アドレスまたは可変アドレス0 - Nのいずれかを使用して、デバイス中の位置情報を符号化する実施例を示し、この構成はまた、レガシーシステムと適合性がある。

【図11 d】図11 dは、標準レイアウトおよび完全位置情報を伴うデジタルマイクロホンのレイアウトの実施例を示す。

【図11 e】図11 eは、弱い出力ドライバおよびフィードバックを使用することによって、GND、VDD、およびFLOAT状態の間を区別することが可能である、I/Oセルを示す。

【図12】図12は、マスタデバイスまたはスレーブデバイスの能力情報を取得するためのFUNCTION CAPABILITYを起動した後にレジスタから読み取る実施例を示す。

20

【図13】図13は、統一バス通信プロトコルの例示的实施形態でサポートされることができる、種々のデータ形式の実施例を示す。

【図14】図14は、関数READ LAST ERRORの実行後にデータレジスタから読み取られた状態の実施例を示す。

【図15】図15は、統一バス通信プロトコルの例示的实施例で使用され得る、エラーコードの実施例を示す。

【図16 a】図16 aは、統一バス通信プロトコルの例示的实施形態のためのマスタデバイスまたはスレーブデバイスに使用されることができる、レジスタの定義の実施例である

30

。【図16 b】図16 bは、統一バス通信プロトコルの例示的实施形態のためのマスタデバイスまたはスレーブデバイスに使用されることができる、レジスタの定義の別の実施例である。

【図17】図17は、統一バス通信プロトコルの例示的实施形態でポートまたはデバイスのために設定されることができる、種々の電力消費レベルまたは電力管理モードの実施例である。

【図18】図18は、統一バス通信プロトコルを使用して2つのスレーブデバイスと通信する、マスタデバイスの例示的实施形態を示し、デバイスは、種々のチャンネルに割り付けられるポートを有する。

40

【図19】図19は、チャンネル間の例示的マッピング、およびポートのどのチャンネルが通信に使用されるかを説明するチャンネル選択フィールドを示す。

【図20】図20は、アドレスおよびREADおよびWRITE動作中のバンク選択に関する例示的動作を示す。

【図21】図21は、ワードモードでフレーム中で1回以上繰り返されている、単一のポートからのデータの実施例を示す。

【図22 a】図22 aは、所与のフレーム中の等時性、非同期、およびマルチフレーム転送の使用に応じて設定される、種々のフィールド値の実施例を示す。

【図22 b】図22 bは、図16 aのPCLKDフィールドを符号化するために使用されることができる、圧縮形式の実施例を示す。

50

【図22c】図22cは、所与のフレーム中の等時性、非同期、およびマルチフレーム転送の使用に応じて設定される、種々のフィールド値の代替実施形態の実施例を示す。

【図23】図23は、ワードモードで低遅延を達成するために、同一のポートからのチャネル間の分割を伴う複数のデータチャネルを使用するときのフレーム形式の実施例を示す。

【図24】図24は、複数のデータチャネルからのデータが、ワードモードでフレーム中で数回繰り返されるときのフレーム設定の実施例を示す。

【図25a】図25aは、統一バス通信プロトコルで使用されるフレームタイプに応じた、COMMAND SEPARATIONフィールドの定義の実施例を示す。

【図25b】図25bは、同期化のための時間を削減するためにビットストリームモードで使用されることができる、おくつかのサブフレーム長の例示的定義を示す。

【図25c】図25cは、統一バス通信プロトコルで使用されるフレームタイプに応じた、COMMAND SEPARATIONフィールドの定義の代替実施例を示す。

【図26】図26aおよび26bは、ビットストリームモードでHSTARTおよびVSTARTフィールドの異なる値を使用することによって達成されることができる、異なるフレーム形式の実施例を示す。

【図27】図27は、ビットストリームモードでの例示的实施形態のためのHSPACINGおよびVSPACINGフィールドのLSB、MSB、およびLSB+1ビットの定義を示す。

【図28a】図28aは、統一バス通信プロトコルを使用する、電流および電圧感知を伴うステレオシステムの例示的実施形態を示す。

【図28b】図28bは、図28aのステレオシステムに使用されることができる、ビットストリームフレーム形式の実施例を示す。

【図28c】図28cは、図28aのステレオシステムに使用されることができ、感知信号のためにより少ない帯域幅を使用する、ビットストリームフレーム形式の別の実施例を示す。

【図28d】図28dは、より少ない帯域幅が音声データと比較して制御データ上で使用されるように、オーディオデータと制御データとの間の帯域幅が変化させられる、ビットストリームモードでHSTART、VSTART、HSPACING、およびVSPACINGフィールドの値を設定することによって達成されることができる、ビットストリームフレーム形式の実施例を示す。

【図29a】図29aは、マスタデバイスに使用されることができる、レジスタの定義の例示的実施形態である。

【図29b】図29bは、マスタデバイスに使用されることができる、レジスタの定義の別の例示的実施形態である。

【図29c】図29cは、第1の段階のためのサンプリング率の定義の実施例である。

【図29d】図29dは、随意的な段階のためのサンプリング率の定義の実施例である。

【図29e】図29eは、第2の段階のためのサンプリング率の定義の実施例である。

【図29f】図29fは、第3の段階のためのサンプリング率の定義の実施例である。

【図29g】図29gは、複数のサンプル率比に好適であり得る、マルチフォーマットデシメータシステムの実施例である。

【図29h】図29hは、複数のサンプル率比に好適であり得る、マルチフォーマット補間器システムの実施例である。

【図29i】図29iは、FRAME CONTROLフィールドへの書き込みの解釈の実施例である。

【図30】図30は、例示的実施形態における、FRAME CONTROLフィールドに基づくWRITEおよびREAD動作、ならびにFRAME CONTROLフィールドの値に応じて監視されることができる、例示的動作の組み合わせを示す。

【図31】図31は、別の例示的実施形態における、FRAME CONTROLフィールドに基づくWRITEおよびREAD動作、ならびにFRAME CONTROLフィ

10

20

30

40

50

ールドの値に応じて監視されることができる、例示的動作の組み合わせを示す。

【図3 2 a】図3 2 aは、INTERFACE CONTROLレジスタの構成要素の例示的实施形態を示す。

【図3 2 b】図3 2 bは、INTERFACE CONTROLレジスタの構成要素の別の例示的实施形態を示す。

【図3 3 a】図3 3 aは、図2 9 aのレジスタ定義に対応する少なくとも1つの例示的実施形態において、どのようにしてMCLKDフィールドが符号化されることができるかという実施例を示す。

【図3 3 b】図3 3 bは、図2 9 bのレジスタ定義に対応する少なくとも1つの例示的実施形態において、バスに対するクロック信号を生成するために使用され得る、例示的周波数分割を示す。

【図3 4 a】図3 4 aは、IRQ MASKレジスタで使用されることができる、サブフィールドの例示的实施形態を示す。

【図3 4 b】図3 4 bは、IRQ MASKレジスタで使用されることができる、サブフィールドの別の例示的实施形態を示す。

【図3 5】図3 5は、FRAME DONE MASKフィールドを変更するための例示的タイミング図を示す。

【図3 6 a】図3 6 aは、INTERFACE STATUSレジスタの構成要素の例示的实施形態を示す。

【図3 6 b】図3 6 bは、INTERFACE STATUSレジスタの構成要素の別の例示的实施形態を示す。

【図3 7】図3 7は、スレーブ状態レジスタの例示的定義を示す。

【図3 8 a】図3 8 aは、スレーブデバイスのデバイスIDを符号化するために使用されることができる、コンパクトな符号化形式の例示的实施形態を示す。

【図3 8 b】図3 8 bは、スレーブデバイスのデバイスIDを符号化するために使用されることができる、一般的な符号化形式の例示的实施形態を示す。

【図3 9】図3 9は、単一の時間フレームに対するワードモードで使用される、一般的なワードフレームを使用する。

【図4 0】図4 0は、単一の時間フレームに対するワードフレーム形式の例示的实施形態を示す。

【図4 1】図4 1は、単一の時間フレームに対するワードフレーム形式の別の例示的实施形態を示す。

【図4 2】図4 2は、単一の時間フレームに対するビットストリームフレーム形式の例示的实施形態を示す。

【図4 3】図4 3は、単一の時間フレームに対する統一フレーム形式の例示的实施形態を示す。

【図4 4】図4 4は、単一の時間フレームに対する統一フレーム形式の別の例示的实施形態を示す。

【図4 5 a】図4 5 aおよび4 5 bは、ワードモードで動作するときの統一バス通信プロトコルに対するバス周波数、チャンネルの数およびタイプの例示的組み合わせの表を示す。

【図4 5 b】図4 5 aおよび4 5 bは、ワードモードで動作するときの統一バス通信プロトコルに対するバス周波数、チャンネルの数およびタイプの例示的組み合わせの表を示す。

【図4 6 a】図4 6 aおよび4 6 bは、ビットストリームモードで動作し、ビットストリームフレーム形式を使用するときの統一バス通信プロトコルに対するバス周波数、チャンネルの数、およびオーバーサンプリングレートの例示的組み合わせの表を示す。

【図4 6 b】図4 6 aおよび4 6 bは、ビットストリームモードで動作し、ビットストリームフレーム形式を使用するときの統一バス通信プロトコルに対するバス周波数、チャンネルの数、およびオーバーサンプリングレートの例示的組み合わせの表を示す。

【図4 7】図4 7は、ハイブリッドワードモードで動作するときの統一バス通信プロトコルに対するバス周波数、チャンネルの数およびタイプの例示的組み合わせの表を示す。

10

20

30

40

50

【図48a】図48a、48b、および48cは、ビットストリームモードで動作し、統一ビットストリームフレーム形式を使用するときの統一バス通信プロトコルに対するバス周波数、チャンネルの数およびタイプの例示的組み合わせの表を示す。

【図48b】図48a、48b、および48cは、ビットストリームモードで動作し、統一ビットストリームフレーム形式を使用するときの統一バス通信プロトコルに対するバス周波数、チャンネルの数およびタイプの例示的組み合わせの表を示す。

【図48c】図48a、48b、および48cは、ビットストリームモードで動作し、統一ビットストリームフレーム形式を使用するときの統一バス通信プロトコルに対するバス周波数、チャンネルの数およびタイプの例示的組み合わせの表を示す。

【図49a】図49aは、統一バス通信プロトコルにおいてデジタルワード形式でNビットを符号化するための一般的形式の例示的实施形態を示す。

10

【図49b】図49bは、携帯電話システムのためのビットストリーム符号化形式とともに使用されることができる、いくつかの一般的オーバーサンプリング係数の実施例を示す。

【図49c】図49cは、文字列符号化形式の可能な組み合わせの実施例を示す。

【図50a】図50aは、フラクショナルフローを実装するために使用されることができるアルゴリズムの例示的实施形態を示す。

【図50b】図50bは、例示的シナリオに対する図50aのアルゴリズムの計算の実施例を示す。

【図50c】図50cは、種々の再生シナリオに対するフラクショナルフローに使用されることができる種々の値の実施例を示す。

20

【図51a】図51aは、統一バス通信プロトコルを使用するディスプレイを伴う制御システムの例示的实施形態を示す。

【図51b】図51bは、統一バス通信プロトコルを使用する携帯電話システムの例示的实施形態を示す。

【図52a】図52aは、統一バス通信プロトコルを使用する住宅用安全システムの例示的实施形態を示す。

【図52b】図52bは、統一バス通信プロトコルを使用する家庭用娯楽システムの例示的实施形態を示す。

【図52c】図52cは、統一バス通信プロトコルを使用する家庭用娯楽システムの例示的实施形態を示す。

30

【図52d】図52dは、統一バス通信プロトコルを使用する計装システムの例示的实施形態を示す。

【図52e】図52eは、統一バス通信プロトコルを使用して通信することができる、電子キーの例示的实施形態を示す。

【図52f】図52fは、統一バス通信プロトコルを使用して通信することができる、メモリスティックの例示的实施形態を示す。

【図52g】図52gは、統一バス通信プロトコルを使用して通信することができる、加入者識別モジュール(SIM)カードの例示的实施形態を示す。

【図52h】図52hは、統一バス通信プロトコルを使用して通信することができる暗号化クレジットカードの例示的实施形態を示す。

40

【図52i】図52iは、統一バス通信プロトコルを使用する心拍数モニタシステムの例示的实施形態を示す。

【図53】図53は、統一バス通信プロトコルに従ってバスを動作させる方法の例示的实施形態を示す。

【図54】図54は、汎用同期化方法の例示的实施形態の略図である。

【図55a】図55aおよび55bは、それぞれ、真の同期化パターンが伝送されたデータ中で送信される第1のシナリオ、ならびに真および偽の同期化パターンが伝送されたデータ中で送信される第2のシナリオの略図である。

【図55b】図55aおよび55bは、それぞれ、真の同期化パターンが伝送されたデー

50

タ中で送信される第1のシナリオ、ならびに真および偽の同期化パターンが伝送されたデータ中で送信される第2のシナリオの略図である。

【図56】図56は、並列実装を利用する汎用同期化方法の別の例示的实施形態の略図である。

【図57】図57は、汎用同期化方法とともに使用されることができ、ワードモード同期化方法の例示的实施形態の略図である。

【図58a】図58aは、欠落した同期化パターンを取り扱うために使用され、かつ図7のワードモード同期化方法とともに使用され得る、方法の例示的实施形態の略図である。

【図58b】図58bは、現在の検索位置がフレームまたはチャンネルの最大長を超えるかどうかをチェックするために図57の方法とともに使用され得る、方法の例示的实施形態の略図である。

10

【図59】図59は、汎用同期化方法とともに使用されることができ、ビットストリームモード同期化方法の例示的实施形態の略図である。

【図60】図60は、汎用同期化方法の別の例示的实施形態の略図である。

【図61a】図61aは、欠落した同期化パターンをチェックするために使用されることができ、かつ図60の汎用同期化方法とともに使用され得る、別の方法の例示的实施形態の略図である。

【図61b】図61bは、フレームオーバーラン(すなわち、フレーム内で可能であるよりも長期間にわたる同期化パターンの欠如)をチェックするために使用されることができ、かつ図60の方法とともに使用され得る、方法の例示的实施形態の略図である。

20

【図62】図62は、図60の汎用同期化方法とともに使用され得る、ビットストリームモード同期化方法の別の例示的实施形態の略図である。

【図63】図63は、図62のビットストリームモード同期化方法とともに使用され得る、ビットストリーム更新方法の例示的实施形態の略図である。

【図64】図64は、迅速再同期化方法の例示的实施形態の略図である。

【図65】図65は、図60の汎用同期化方法とともに使用されることができ、クロックゲーティング方法の例示的实施形態の略図である。

【図66】図66は、無線通信で統一バス通信プロトコルの種々の実施形態をどのようにして使用されることができるといふ実施例を示す。

【図67】図67は、同期化ワードのためのフィールドおよびビット割付の別の例示的实施形態の略図である。

30

【図68】図68は、ワードモードでの図67のSワードのデータ同期化フィールドへのデータ動作を示す、別の例示的タイミング図である。

【図69a】図69aは、統一バス通信プロトコルの例示的实施形態に対するマスタデバイスまたはスレーブデバイスに使用されることができ、レジスタの定義の別の実施例である。

【図69b】図69bは、サブグループ当り使用されることができ、種々の最大チャンネルを示す実施例である。

【図69c】図69cは、1つのデバイスグループアドレスに割り当てられたデバイスを有する、システムの例示的实施形態の略図である。

40

【図69d】図69dは、3つのデバイスグループアドレスに割り当てられたデバイスを有する、システムの例示的实施形態の略図である。

【図70】図70は、Xワードで設定され得る、関数および対応するビット設定の別の例示的リストを示す。

【図71】図71は、デバイスをバスに連結するためにリングトポロジを使用するシステムの例示的实施形態であり、本システムは、統一バス通信プロトコルを使用する。

【図72】図72は、デバイスをバスに連結するためにパイプトポロジを使用するシステムの例示的实施形態であり、本システムは、統一バス通信プロトコルを使用する。

【図73】図73は、多くのデバイスをバスに連結するためにパイプ制御を使用するシステムの例示的实施形態であり、本システムは、統一バス通信プロトコルを使用する。

50

【図74】図74は、デバイスをバスに連結するために1次元交互リングトポロジーを使用するシステムの例示的实施形態であり、本システムは、統一バス通信プロトコルを使用する。

【図75】図75は、デバイスをバスに連結するために2次元リングトポロジーを使用するシステムの例示的实施形態であり、本システムは、統一バス通信プロトコルを使用する。

【図76】図76は、多くのデバイスをバスに連結するためにパイプ制御を使用するシステムの別の例示的实施形態であり、本システムは、統一バス通信プロトコルを使用する。

【図77a】図77aは、2Xオーバークロックデータスロットの実施例のタイミング図である。

10

【図77b】図77bは、オーバークロックデータスロットを利用するデバイスに使用され得る、タイミングパラメータ値の例示的实施形態を示す表である。

【図77c】図77cは、2倍のオーバークロッキングがある実施例に対するバス上のデータ転送およびタイミングの説明図である。

【図77d】図77dは、レジスタがオーバークロッキングを実装するための例示的設定、および関連節電の推定値の表である。

【図78a】図78aは、一度にデータの複数のビットを送信するためのタイミングおよび電圧レベルスキームを示す。

【図78b】図78bは、オーバークロッキングモードを使用して、デバイスがデータスロット中で複数のパックしたビットを送信するときに使用され得る、例示的符号化の表である。

20

【図79】図79は、MASTER INITIATED DEVICE WAKE UP FROM SLEEPモード中にスレーブデバイスによって見られるデータの実施例である。

【図80a】図80aは、統一バス通信プロトコルを使用してポートまたはデバイスのために設定することができる、種々の電力消費レベルまたは電力管理モードの代替実施形態の実施例である。

【図80b】図80bは、例示的UARTタイミング仕様の表である。

【図81a】図81aは、同期化に2つのスロット、データに1つのスロットを利用する、単一ワイヤ信号伝達システムの実施例である。

30

【図81b】図81bは、同期化に2つのスロット、データに4つのスロットを利用して、クロックおよびデータ情報の両方を転送する単一ワイヤ信号伝達システムの実施例である。

【図81c】図81cは、同期化に1つのスロット、データに4つのスロットを利用し、それによって、帯域幅を増加させ、電力消費を低減させる単一ワイヤ信号伝達システムの実施例である。

【発明を実施するための形態】

【0050】

ここで、概念の1つ以上の実施形態の実施例を提供するように、種々のシステムまたはプロセスを以下で説明する。以下で説明される実施形態は、請求項のうちのいずれも限定せず、請求項のうちのいずれかは、以下で説明されるものとは異なるプロセスまたはシステムを対象とし得る。請求された特徴は、以下で説明されるいずれか1つのシステムまたはプロセスの特徴の全てを有するシステムまたはプロセスに、あるいは以下で説明されるシステムまたはプロセスの複数または全てに共通する特徴に限定されない。以下で説明されるシステムまたはプロセスは、請求項のうちのいずれかに記載される実施形態ではない可能性がある。本書で請求されていない、以下で説明されるシステムまたはプロセスで開示される任意の概念が、別の保護機器、例えば、継続特許出願の主題であり得、出願者、発明者、または所有者らは、本書でのその説明によって、任意のそのような概念を放棄し、否定し、または一般公開に捧げることを意図しない。

40

【0051】

50

さらに、例証を簡単および明確にするために、適切であると見なされる場合、参照数字が、対応または類似する要素を示すように図の間で繰り返されることが理解されるであろう。加えて、本明細書で説明される実施形態の徹底的な理解を提供するために、多数の具体的詳細が記載される。しかしながら、これらの具体的詳細を伴わずに、本明細書で説明される実施形態が実践され得ることが、当業者によって理解されるであろう。他の場合において、周知の方法、手順、および構成要素は、本明細書で説明される実施形態を曖昧にしないよう詳細に説明されていない。また、本説明は、本明細書で説明される実施形態の範囲を限定すると見なされるものではない。

【 0 0 5 2 】

また、本明細書で使用される場合、連結されるという用語は、用語が使用される文脈に応じて、いくつかの異なる意味を有することができることに留意されたい。例えば、連結という用語は、機械的、電氣的、光学的、または通信的な意味合いを有することができる。例えば、いくつかの文脈では、連結という用語は、2つの要素またはデバイスを互に物理的に接続できるか、または例えば、ワイヤまたはケーブル等の物理的連結を介して、1つ以上の中間要素またはデバイスを通して互に接続できることを示す。いくつかの文脈では、連結という用語は、無線信号または光学信号等の他の手段を通して、2つの要素またはデバイスを接続できることを示す。いくつかの文脈では、連結という用語は、例えば、電気、光学、または無線信号等の信号を通して、要素またはデバイスが通信可能に互に接続され得ることを示す。さらに、「通信連結」という用語は、要素またはデバイスが、電氣的に、光学的に、または無線で、データを別の要素またはデバイスに送信するとともに、別の要素またはデバイスからデータを受信できることを示す。

【 0 0 5 3 】

また、本明細書で使用される場合、「マスタ」および「スレーブ」という用語は、純粹に技術的な観点から意図され、具体的には、1つの要素またはデバイスが命令または制御信号を別の要素またはデバイスに提供するホストデバイスまたは要素と、周辺デバイスまたは要素との間の技術的関係を表すことに留意されたい。したがって、本明細書で使用される場合、「マスタ」および「スレーブ」という用語の使用は、ここで与えられる技術的意味を超えるどんな意味合いをも持つように意図されていない。

【 0 0 5 4 】

また、通信プロトコルは、バス通信プロトコルとして説明され、要素と一緒に連結する物理的バスを示す、種々の例示的实施形態が提供されるが、本明細書で説明される通信プロトコルの種々の実施形態はまた、無線または光学インターフェース等の非物理的インターフェースを経由して実装されることができることに留意されたい。

【 0 0 5 5 】

本明細書で使用される場合、「実質的に」、「約」、「おおよそ」および「ほぼ」等の程度の用語が、最終結果が著しく変更されないような修正用語の合理的な量の偏差を意味することに留意されたい。これらの程度の用語は、この偏差が、それが修正する用語の意味を無効にしないであろう場合、修正用語の最大で±10%の偏差を含むものと解釈されることが可能である。また、本明細書で使用される場合、要素は、1つ以上の機能を行うように「構成」または「適合」されているとして説明され得る。概して、機能を行うように構成または適合されている要素は、その機能を行うために適しているか、その機能を行うように動作可能であるか、またはその機能を行うことが可能である。さらに、用語「current」は、本明細書において、「電流」（単位アンペアで測定される量）を意味するために使用されるが、これが、現在、今生じているイベントを指す場合があり得、特別な場合におけるその意味は、文脈から概ね明らかである。

【 0 0 5 6 】

さらに、以下の節において、実施形態の異なる側面が、より詳細に定義される。定義される各側面は、それとは反対に明確に示されない限り、任意の他の側面と結合され得る。特に、好ましい、または有利であるとして示される任意の特徴は、好ましい、または有利であるとして示される少なくとも1つの他の特徴と結合され得る。好ましい、または有利で

10

20

30

40

50

あり得る特徴または構成要素は、必ずしも不可欠ではない。

【0057】

発明を実施するための形態は、モバイルデバイスの一般的説明から始まり、次いで、異なるタイプのデータ形式をサポートすることができる一方で、より少ない数のワイヤを用いてデバイスを連結するために使用されることができる、統一バス通信プロトコルの例示的实施形態を続けて説明する。したがって、統一バス通信プロトコルは、異なる形式で数値データを生成するが、同一のバスに連結されることができる、デバイス用の統一インターフェースとしての役割を果たす。

【0058】

統一バス通信プロトコルは、マスタデバイスとスレーブデバイスとの間、2つのマスタデバイスの間、2つのスレーブデバイスの間、およびマスタデバイスと1つより多くのスレーブデバイスとの間等の少なくとも2つのデバイス間の通信を可能にする。デバイス間のデータ通信は、概して、特に指定がない限り、連続クロックを利用し、通信は、特に指定がない限り、双方向であり得る。

10

【0059】

数値データは、オーディオデータ、または電流、電圧、圧力、温度等の測定データであり得る。数値データは、最初に、ビットストリーム形式またはデジタルワード形式でデバイスによって生成されることができる。デジタルワードは、2進コード化ワード、および浮動小数点ワード（すなわち、符号なし、または2の補完ワード）を対象とするように意図されている。

20

【0060】

ワードモードでは、デジタルワードは、別の形式に変換されることなく、バスに沿って伝送されることができる。ビットストリームモードでは、データは、ビットストリームフレーム形式、または一種のビットストリームフレーム形式である統一フレーム形式のいずれかを使用して、パッケージ化される。ビットストリームフレーム形式では、ビットストリームが、フレームチャンネルを使用して、異なるビットストリームデータチャンネルから1度に1ビット伝送される。ビットストリームデータという用語の代わりに、パルス密度変調データ（PDM）データという用語が時として使用されることに留意されたい。

【0061】

データチャンネルという用語は、概して、ある形式でデータを生成または受信するチャンネルを指す。したがって、ワードデータチャンネルは、デジタルワードデータを備え、ビットストリームデータチャンネルは、ビットストリームデータを備える。

30

【0062】

フレームチャンネルという用語は、フレームチャンネルが多重化され、フレーム中であるタイムスロットが与えられる、ビットストリームフレームまたは統一ビットストリームフレーム中のデータのチャンネルを定義するために使用される。

【0063】

統一フレーム形式では、少なくとも1つのワードデータチャンネルに対するデジタルワードが、ビットストリームに変換され、1度に1ビット伝送される、少なくとも1つの仮想フレームチャンネルがある。統一フレーム形式では、1つ以上のビットストリームデータチャンネルからビットストリームデータを受信する、少なくとも1つのビットストリームフレームチャンネルもあり得る。ビットストリームフレーム形式および統一フレーム形式は、制御フレームチャンネル中にコマンドワードのビットストリームを含み、これは、削減された数の端子の使用により費用削減をもたらす。これらのフレーム形式を、本説明において以降でさらに詳細に説明する。

40

【0064】

ビットストリームデータは、一連のビットとして伝送され、パルス密度変調（PDM）データと称されることもある、データである。それらは、典型的には、デルタシグマ変換器の出力等のオーバーサンプリングされたシステムの出力として生成される。オーバーサンプルレートは、サンプリングクロックとデシメーション後の最終出力サンプルレートと

50

の間の比（または補間については逆も同様）を示すために使用される。ビットストリームは、電気通信およびオーディオ用途においてデータを転送するため、および（例えば、スーパーオーディオCD（SACD）上で）記憶形式として使用される。

【0065】

デジタルワードは、デジタル形式でアナログ値を表すために使用される、ある分解能を有する0および1のシーケンスである。例えば、16ビットデジタルワードは、各ビットが0または1であり得る、16ビットを有する。デジタルワードは、パルス符号変調（PCM）と呼ばれる方法を使用して生成されることができ、その場合、デジタルワードは、PCMワードまたは（オーディオ用途については）オーディオワードと称される。PCMワードは、CD、Blu-ray（登録商標）プレーヤ、DVDプレーヤ、コンピュータ、スマートフォン、ラップトップ、タブレット等を含む、種々のデバイスでのデジタルオーディオに使用される。

10

【0066】

バス上で通信されるデータは、同期化データ、制御データ、数値データ、およびクロック信号を含む。電力もまた、バスを経由して伝送されることができる。統一バス通信プロトコルは、バスに使用され得るワイヤの数の削減を可能にし、概して、詳細に説明されるように、単一ワイヤバスまたは2ワイヤバス上で実装されることができる。単一ワイヤバスは、同期化データ、制御データ、数値データ、クロック信号、および電力を伝送するために、1本のワイヤを使用する。2ワイヤバスは、同期化データ、制御データ、数値データ、および電力を伝送するために第1のワイヤを、クロック信号を伝送するために第2のワイヤを使用することができる。したがって、バスおよび関連統一バス通信プロトコルは、ピンまたはワイヤの数が限定される動作、または雑音排除性に関する高い信頼性が所望される動作に使用され得る。本明細書で説明されるバスおよび関連統一バス通信プロトコルは、特に指定がない限り、同一のクロック信号によって全て同期化され、通信リンクを提供するために、フレーム形式に従って特定の瞬間にバスを放電する、可変数のデバイスの連続動作を可能にする。システムクロック周波数が低い実施形態では、バス上の帯電を維持するように、1つ以上のバスホルダを追加することができる。

20

【0067】

発明を実施するための形態の第2の部分は、マスタデバイスと、または本明細書で説明される統一バス通信プロトコルの種々の実施形態のうち少なくとも1つを利用するバスと同期化するためにスレーブデバイスによって使用されることができる、同期化方法の種々の例示的实施形態を提供する。同期化方法は、概して、ワードモードで順次に順番付けられ得る、一連のデータビット（すなわち、1および0）である、特定の同期化パターンを検索するための技法を伴い、または同期化パターンのビットは、他のデータビットと多重化または連動され得るが、依然としてビットストリームモードで、同一の順番で伝送される。

30

【0068】

一般に、同期化方法のうち少なくとも1つは、バスが初期化され、ある期間にわたって動作した後に、スレーブデバイスがバスにアタッチすることを有利に可能にするために使用され得る。これは、ホットプラグインと称される。したがって、スレーブデバイスは、この情報を伝えられることなく、どのような動作モードおよびフレーム形式がバス通信プロトコルに使用されているかを自力で決定することができ、これは、有利であり、概して、従来のバス通信プロトコルに対して行われていない。

40

【0069】

バスアーキテクチャでは、多くの場合、デバイスが、例えば、使用されている間にバスを使用する必要があるときを、ならびに、数値データおよびコマンド命令の伝送のため等の使用の性質を信号伝達するための機構がある。しかしながら、バス制御は、多数の非同期プロセスが効率的にバスを共有しようとしているとき、ならびにこれらのプロセスがビットストリームデータおよびデジタルワードデータ等の異なるデータ形式を使用するとき、極めて複雑になり得る。例えば、従来のバスシステムでは、異なるインターフェース

50

が、種々の種類のデータをサポートするために必要とされる。実施例として、PCMインターフェースは、ビットストリームデータではなくデジタルワードデータをサポートし、ビットストリームインターフェースは、デジタルワードデータをサポートしない。いくつかの用途では、センサが、バイナリワードを使用するナイキスト型変換器（例えば、温度センサ、加速度計、およびディスプレイ）を使用し、いくつかの用途では、ビットストリームインターフェースが、ビーム形成、能動雑音消去、または低遅延制御用途等の低遅延制御用途に使用されるため、多くの場合、両方のタイプのデータを使用するシステムがある。従来のバスシステムは、デジタルワードデータを通信する1つのインターフェース（4本のバスワイヤを必要とするMCBSPインターフェース等）、ビットストリームデータを通信する別のインターフェース（さらに4本のバスワイヤを必要とする少なくとも1つのMCBSPインターフェース等）、および制御データ用の別のインターフェース（さらに2本のバスワイヤを必要とするI²C等）を使用し、合計で少なくとも3つのインターフェースならびに少なくとも10本のバスワイヤおよび端子を使用する。これは、関与する構成要素に対する費用および余分な空間要件を追加するとともに、電力消費を増加させる。さらに、そのようなシステムが外部デバイスとともに使用されるとき、複数のコネクタが使用され、より広い空間要件をもたらす。

【0070】

したがって、本明細書で説明される少なくとも1つの実施形態では、本明細書で説明される統一バス通信プロトコルの一側面は、より少数のワイヤ、したがって、より少数の端子を使用しながら、ビットストリームデータ、デジタルワードデータ、および制御データを取り扱うことができる統一インターフェースの提供である。低遅延および高速過渡応答等のビットストリーム処理の利点は、デジタルワード形式の使用を回避することによって維持されることができる。したがって、本明細書で説明される統一インターフェースの少なくとも1つの実施形態の側面は、ビットストリーム処理のための同一の待ち時間（すなわち、1ビット待ち時間）を達成するために、ビットストリームデータを多重化できることである。さらに、デジタルワードデータは、1度に1ビット伝送されることができ、各ビットがビットストリームデータと多重化され、デジタルワードデータインターフェースの使用を排除する。加えて、同期化データおよびコマンドデータ等の制御データは、ビットストリーム形式に変換され、統一バス通信プロトコルを使用して多重化データの一部として組み込まれることができ、これは、外部インターフェース（例えば、I²C）の使用を排除し、また、ロボットプログラム可能マルチフォーマットデータインターフェースも提供する。したがって、統一バス通信プロトコルの少なくとも1つの実施形態の側面は、PCM/TDM/I²Cストリームをトンネリングするため、および低遅延でビットストリームを伝送するためのサポートを含む。

【0071】

クロックデータはまた、（1ワイヤバス実施形態では）制御データに組み込まれることもでき、またはそれは、（2ワイヤバス実施形態では）別個のワイヤを使用して提供されることができる。別のワイヤ上でクロック信号を伝送することにより、スレーブデバイスの実装が単純化されることを可能にし、また、クロックジッタへの影響も低減させる。

【0072】

したがって、統一バス通信プロトコルは、システムの具体的要件に応じて、1本または2本のワイヤを使用して実装されることができる（例えば、ヘッドセットインターフェースとして、または一般的なバスシステムとして）。しかしながら、代替実施形態では、本明細書で説明される統一バス通信プロトコルは、2本より多くのワイヤを有するバスとともに使用されることができるとに留意されたい。バスが1本より多くのワイヤを使用する実施形態は、より高い帯域幅、および物理層でのいくつかの単純化を可能にすることができる。

【0073】

デジタルワードデータをビットストリームデータと多重化するために、本明細書で説明される少なくとも1つの実施形態では、統一インターフェース中の1つ以上のフレームチ

10

20

30

40

50

チャンネルを仮想フレームチャンネルとして割り付けることができ、デジタルワードデータは、データチャンネルに対するデータの適正な順番を維持しながら、ビットストリームデータとともに1度に1ビット多重化される(すなわち、インターリーブまたはインターレースされる)。また、特定の実施形態に応じて、帯域幅制御と、異なるサンプリングレートでサンプリングされるデータを組み合わせることとのうちの少なくとも1つを可能にする、ビットストリームデータを使用するフレームチャンネルとデジタルワードデータを使用するフレームチャンネルとの間のプログラム可能な組み合わせがある実施形態もあり得る。制御データを他のタイプのデータとともに1度に1ビット多重化またはインターレースすることができるように、統一インターフェースの少なくとも1つの実施形態では、制御チャンネルも割り付けられる。

10

【0074】

本明細書で説明される少なくとも1つの実施形態では、統一バス通信プロトコルの側面は、例えば、より効率的に、電気通信用途(例えば、19.2および13.0MHz)で一般的に使用されているもの等の種々のクロック周波数およびサンプルレートをサポートすることをより容易にする、変動フレーム長の使用である。さらに、統一バス通信プロトコルは、SLIMbusバス通信プロトコルのために選択され、12.288MHz用途に適している、768ビットのフレーム長等の他の一般的に使用されているフレーム長に適応することができる。統一バス通信プロトコルによって提供される制御機能性は、I²Cバス通信プロトコルより幅広い機能性を有し、例えば、少なくともいくつかの実施形態では、オーディオに使用されるもの等の特別な機能をサポートする。統一バス通信プロトコルの少なくとも1つの実施形態は、制御データと同時に、低遅延を伴うビットストリームデータの等時性転送をサポートする。

20

【0075】

さらに、本明細書で説明される少なくとも1つの実施形態では、バス構造は、統合、電力消費、および経済的理由で低ゲート総数が有利である、センサ用途およびオーディオ用途で使用するために好適である、低複雑性を有する。低ゲート総数は、本明細書で説明される統一バス通信プロトコルの種々の実施形態に従って通信および同期化するために、同期エンジンおよび関連ハードウェアを使用する、マスタおよびスレーブデバイスの両方について達成されることができる。低ゲート総数はまた、集積回路(IC)実装において、より低いドライバ複雑性、より小さいシリコン面積、およびより低い費用ももたらす。さらに、統一バス通信プロトコルがI²Cトンネリングに適しているため、レガシーシステムは、わずかな努力でサポートされ続けることができる。

30

【0076】

本明細書で説明される少なくとも1つの実施形態では、統一バス通信プロトコルの側面は、スレーブデバイスの状態の連続監視のためのサポートである。これは、いかなるハンドシェイキングも伴わずに、スレーブデバイスのホットプラグインおよび除去を可能にする。これはまた、可能な限りバスを妨げないために、エラーの場合にスレーブデバイスがバスから自ら退場することも可能にする。したがって、スレーブデバイスが統一バス通信プロトコルを使用してバスを停滞させることは可能ではなく、これは、I²Cバス通信プロトコルの場合には当てはまらない。さらに、エラーの検出および補正は、本明細書で説明される統一バス通信プロトコルの少なくともいくつかの実施形態では、SMbus通信プロトコルと比較してはるかに速く実装されることができる。例えば、SMbusは、エラーを検出する前に約35~40ミリ秒を要し得る。停滞したバスの検出は、内部タイムアウト機能によって起動される(バス上にアクティビティがない場合)。この状況は、本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態を使用して、はるかに速く発見されることができる。いくつかの最悪のシナリオでは、スレーブデバイスがバスとの同期化に戻るため約50~100ミリ秒を要し得るが、典型的には、本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態については、約1~10ミリ秒を要するであろう。

40

【0077】

50

ここで図1を参照すると、可搬性電子デバイス10の斜視図が示されている。本実施例では、可搬性電子デバイス10は、携帯電話またはスマートフォン等の移動通信デバイスである。しかしながら、本明細書で説明される実施形態は、電話に限定されず、異なるデータ形式用の統一インターフェースを提供するバス通信プロトコルから利益を享受することができる、任意の電子デバイスに拡張されることができることを理解されたい。そのような電子デバイスの実施例は、概して、携帯電話無線オーガナイザ、携帯情報端子、コンピュータ、ラップトップ、ハンドヘルド無線通信デバイス、無線対応ノートブックコンピュータ、タブレットコンピュータまたは電子書籍リーダ、電子セキュリティデバイス、無線インターネットアプライアンス等の任意の可搬性電子デバイスを含む。他の実施例が、本説明の全体を通して提供される。移動式である、本明細書に記載される電子デバイスは、概して、可搬性であり、したがって、バッテリー電源型であり、限られた処理能力を有し得、その場合、本明細書で説明される統一バス通信プロトコルの実施形態のうちの少なくとも1つ等の統一データインターフェースを提供するバス通信プロトコルを使用することが有益である。

10

【0078】

可搬性電子デバイス10は、表示画面14と、キーボード/キーパッド16と、一式のボタン18と、トラックパッドまたはトラックボール等のユーザ入力デバイス20とを含む、本体12を有する。ユーザ入力デバイス20は、ジョイスティック、スクロールホイール、ローラホイール、マウスまたはタッチパッド等、または別のボタンとして提供されることもできる、ユーザ動作型ポインティングまたは入力デバイスを表すことが理解されるであろう。可搬性電子デバイス10は、当業者に周知であるため、示されていない、または説明されていない他の部品を含む。可搬性電子デバイス10はまた、ジャックを受け取るための少なくとも1つのポートも含むが、これは図1に示されていない。

20

【0079】

ここで図2を参照すると、可搬性電子デバイス10の一部分のブロック図が示されている。可搬性電子デバイス10はさらに、可搬性電子デバイス10内に統合されるヘッドセットまたはヘッドホンインターフェースチップ等のチップ32に接続される、コントローラ30を含む。コントローラ30は、プロセッサまたは専用回路を使用して実装されることができる。チップ32は、ビデオケーブルまたはヘッドセットケーブル等のケーブル40に関連付けられるジャック38を受け取るためのポート36と統合される、スイッチマトリクスおよびジャック構成検出部分34を含む。スイッチマトリクス34は、ジャック38内の対応するワイヤまたはライン44aから44dを用いて信号を受信および伝送するための複数の個々の入力および出力ポート42を含む。

30

【0080】

ジャック38内のワイヤ44aから44dは、オーディオおよびビデオライン等の信号ラインを表す。一式の個々のライン、典型的には、4本のラインは、他の数のワイヤを伴う他のジャック構成が考慮されるが、可搬性電子デバイス10と、ヘッドセット等、ケーブルの他方の側に位置するデバイスとの間の通信を可能にする。一実施形態では、ワイヤ44aおよび44bは、一对のオーディオラインであり得、ワイヤ44cは、接地ラインであり得、ワイヤ44dは、マイクロホンラインであり得る。しかしながら、ラインのうち1本のみが通信に使用され、それによって、単一ワイヤバス実施形態のための通信バスとしての機能を果たす。残りのラインは、他の機能性に使用され得る。典型的には、1本のラインが、接地に使用されるであろう一方で、残りの2本のラインは、ヘッドホン出力に使用され得る。別の構成では、1本のラインがバス通信に使用され得、1本が接地に使用され得、第3のラインが電力に使用され得、最後のラインが、ビデオ信号の伝送のため、または他の機能性のための別個のクロックラインとして等、他の目的で残される。他の実施形態では、単一ワイヤバスが、デバイス間のデジタルまたはアナログ伝送で使用され得る。代替として、2ワイヤバス実施形態は、以下でさらに詳細に説明されるように、2つのチップの間で通信を提供するために有用である。

40

【0081】

50

ここで図3 aを参照すると、バスシステム50の例示的实施形態の概略図が示されている。バスシステム50は、可搬性電子デバイスまたは可搬性電子デバイス内のヘッドセットインターフェースチップ等のマスタデバイス52と、ヘッドセット等のスレーブデバイス54とを含む。1つだけのスレーブデバイス54が示されているが、複数のスレーブデバイスがマスタデバイス52との通信のためにバスシステム50に連結され得ることが理解されるであろう。したがって、スレーブデバイス54について本明細書で提供される説明は、概して、バスシステム50に接続される他のスレーブデバイスに適用されることができる。

【0082】

マスタデバイス52またはスレーブデバイス54は、ベースバンドプロセッサまたは移動処理ユニットに接続される、集積回路間(I²C)インターフェース56を含み得る。マスタデバイス52またはスレーブデバイス54はまた、デジタルオーディオデータ用のI²Cインターフェースを含み得る。I²Cインターフェース56およびシリアルインターフェース58への入力または入力信号60は、外部クロック信号60a(EXT CLK)、I²Cクロック信号60b(I²C CLK)、およびI²Cデータ信号60c(I²C DAT)を含み得るが、それらに限定されない。低速シリアルインターフェース58の出力は、同軸ケーブル等のケーブル62を介して、スレーブデバイス54に接続される。上記で説明されるように、ケーブル62内のワイヤまたはラインのうちの1本は、スレーブデバイス54とマスタデバイス52との間の通信に使用されるバス64を提供し、単一ワイヤバスとして見なすことができる。バス64は、ピンまたはワイヤの数が限定される場合に、または雑音排除性に関する高い信頼性のために使用され得る。他の実施形態では、2ワイヤバスを使用することができる。

【0083】

単一ワイヤバス実施形態では、単一のワイヤが、単一のバスサイクルでのクロックおよびデータの両方の伝送を含むが、それに限定されない、複数の機能を組み合わせ得る。動作時に、I²Cインターフェース56は、バス64を経由してデータを取り出し、スレーブデバイス54に送信する。他の実施形態では、バス64を経由した通信は、ベースバンドプロセッサまたは別の処理ユニットへの接続を介して制御され得る。したがって、マスタデバイス52の制御は、I²Cインターフェース56を介したものの、別の制御インターフェースを通したものの、または限定されないが、PING、READ、WRITE、およびFUNCTIONコマンドまたは動作等の種々の動作をマスタデバイス52に行わせる、ベースバンドプロセッサまたは他の処理ユニットへの接続の一部としてのものであり得る。本明細書で説明される少なくとも1つの実施形態では、各フレームが1つだけの動作によって定義されるため、PING、READ、WRITE、およびFUNCTIONコマンドは、同一のフレーム中で起こらないであろう。これらの動作を、本説明の全体を通してさらに詳細に説明する。

【0084】

マスタデバイス52は、スレーブデバイス54または複数のスレーブデバイスがバス64に同期化されることを可能にするために、フレーミング情報を生成する。一実施形態では、フレーム長は、コマンドデータの各ブロックの開始間の分離距離を提供する、8ビットレジスタによって決定される。別の実施形態では、測定単位は、4ビットに等しいニブルである。フレーム長は、コマンドパターンによって決定され、一実施形態では、ワードモードで、28ニブルのデフォルト同期分離値を伴う48ビットであり得、それによって、384ビットのフレーム長をもたらす。

【0085】

動作時に、同期化(同期)信号、情報の制御信号、データ、クロック信号、および電力が、バス64を経由してマスタデバイス52とスレーブデバイス54との間で伝送される。したがって、バス64は、いくつかの外部デバイスの連続動作を可能にし、全てのデバイスは、同一のクロック信号によって同期化される。クロック信号は、シグマデルタ変換器等の内部回路のため、または複雑な論理回路の連続動作のために、サンプリングクロッ

10

20

30

40

50

クとして使用されることができ、他の実施形態では、以下でさらに詳細に説明されるように、バス64上にアクティビティがないときに電力を節約するように、クロック信号が静的低論理レベルまたは静的高論理レベル等の定常レベルで保たれる場合があり得る。

【0086】

バス64を経由して通信する構成要素のサイズは、本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態を使用することによって減少させられ得る。例えば、図2の可搬性電子デバイス10で実装される場合、統一バス通信プロトコルは、バス64が単一のワイヤを経由してクロック信号およびデータを伝送することを可能にし、これは、チップ32中で占有されるピンの数を削減し、それによって、他のピンポート42が、他の機能性に使用されること、またはプリント回路基板(PCB)レベルでチップの全ピン総数、シリコン面積、または費用を削減することを可能にする。

10

【0087】

ここで図3bを参照すると、マスタデバイス52をバス64に連結するためのインターフェース回路70の実施例を示す、バスシステム50の概略図が示されている。この概略図は、主にバスの単一ワイヤバージョンに対するものであるが、また、2ワイヤバージョンにおけるデータラインに使用することもできる。スレーブデバイス54をバス64に連結するために、同様の回路を使用することができる。インターフェース回路70は、充電トランジスタ72と、放電トランジスタ74と、電力トランジスタ76と、抵抗器R1とを含む。充電トランジスタ72は、バス64上で論理「1」をアサートするために使用され、放電トランジスタ74は、バス64上で論理「0」をアサートするために使用される。電力トランジスタは、通常は直列終端抵抗器R1を通して提供される電流よりも多くの電流が使用されるときに、迅速にスレーブデバイス(例えば、バッテリー)を充電するために使用され得る。この配列では、バス64は、信号伝達速度が、電力消費、雑音排除性、および帯域幅の間の妥協である、オープンコレクタおよびオープンドレインタイプ等の時間制約型実装に悩まされない。電力トランジスタ76は、随意的であり、デバイスを充電し、通信がない時に外部デバイスに電力供給するために、いくつかの実施形態に含まれ得る。電力トランジスタ76は、レジスタによって制御され、フレームの開始時に更新され得る。

20

【0088】

この特定の例示的実施形態では、充電トランジスタ72および電力トランジスタ76が、PMOSトランジスタである一方で、放電トランジスタ74は、NMOSトランジスタである。充電トランジスタ72のソースノードは、供給電圧VSSに連結され、充電トランジスタ72のゲートノードは、低速シリアルインターフェース58の第1のピンに連結され、充電トランジスタ72のドレインノードは、放電トランジスタ74のドレインノードに連結される。放電トランジスタ74のゲートノードは、低速シリアルインターフェース58の第2のピンに連結され、放電トランジスタ74のドレインノードは、接地に連結される。電力トランジスタ76のゲートノードは、低速シリアルインターフェース58の第3のピンに連結され、電力トランジスタ76のドレインノードは、供給電圧VSSに連結され、電力トランジスタ76のソースノードは、低速シリアルインターフェース58の第4のピンおよびバス64の第1のノードに連結される。抵抗器R1の1つのノードは、充電トランジスタ72のソースノードおよび放電トランジスタ74のソースノードに連結され、抵抗器R1の第2のノードは、バス64の第1のノードに連結される。例示的実施形態では、抵抗器R1の抵抗は、インターフェース回路70の出力抵抗が、概して、約75オームであり、供給電圧VSSが3Vであり得るように、選択されることができ、

30

40

【0089】

充電トランジスタ72に接続される低速シリアルインターフェース58のノードは、充電トランジスタ72が全バスサイクルの開始時にバス64を充電し得るように、低い電圧を印加する。その後、バス64は、充電トランジスタ72を動作停止することによって、浮動したままにされる。次いで、マスタデバイス52は、バス64を放電するであろう高い電圧を放電トランジスタ74のゲートに印加することによって、放電トランジスタ74

50

を起動することによってバス64を駆動し得る。抵抗器R1の後のバス64の接続は、バス64からデータを受信するために使用される(場合によっては、向上した設定/ホールドタイミングのマージンのために遅延が含まれ得る)。電力トランジスタ76に連結される低速シリアルインターフェース58のノードは、抵抗器R1を通して利用可能であるバス64により大きい電力を供給するために使用される。これは、トランジスタ72が起動された後に最初に全電力を印加すること(これが共鳴を減少させるであろう)等のある措置が講じられない限り、いくらかの初期共鳴につながり得る。

【0090】

一実施形態では、バス64は、低・高・浮動周期パターンを使用して実装されるが、代替として、高・低・浮動パターンとして実装され得る。データの実際の転送が、浮動期間中に起こる一方で、低・高(または高・低)期間は、電力を伝達するため、およびクロック同期化に使用される。スレーブデバイス54またはマスタデバイス52が論理ゼロを信号伝達したい場合、その特定のデバイスは、バス64を以前と同一の状態で、すなわち、バス64上に同一の値を伴って浮動したままにするであろう。スレーブデバイス54またはマスタデバイス52は、浮動期間中にバス64を放電し、それによって、バス64の状態を反対の論理レベルに変化させることによって、論理1を信号伝達し得る。信号伝達スキームの実施例は、スレーブデバイス54が「0110」を書き込む、バス64上のトランザクションのタイミング図を示す、図4aに示されている。外部クロック信号は、典型的には、マスタデバイス52に提供される(代替的な場合では、マスタデバイス52がクロック信号を生成し得る)。マスタデバイス52またはスレーブデバイス54が3状態モードになるとき、これは、例証目的で図4aに鎖線として描かれている。実践では、出力電圧は、バス64上に残された残りの電荷によって決定されるであろう。しかしながら、バス64が放電されると、何度デバイスがこの放電を試行しようとしても、さらなる電荷がないであろう。したがって、バス64が放電されると、後続のデバイスによるバス64の放電が検出されず、バス衝突が可能ではないので、この信号伝達スキームは、複数のデバイスが同時にバス64を経由して信号伝達または通信しようとするときに、バス競合または混雑の低減または回避を可能にする。

【0091】

図4aの実施例によって見る事ができるように、バス64が4つの時間間隔において時間多重化されるように、タイミングがバス64上で実装される。代替実施形態では、別の数の時間間隔を使用することが可能であり得る。図4aの実施例については、第1の時間間隔では、マスタデバイス52は、クロック周期の最初の25%(すなわち、4分の1)の間、バス64をアクティブローに駆動する。第2の時間間隔では、マスタデバイス52は、クロック周期の第2の25%の間、バス64をアクティブハイに駆動する。低・高クロック遷移は、連続サンプリングクロック信号として使用され得る。第3および第4の時間間隔では、マスタデバイス52は、スレーブデバイス54またはマスタデバイス52がバスをローにプルし、「1」を信号伝達しない限り、残りのクロック周期にわたってバス64を浮動状態のままにする。例えば、第3の間隔では、スレーブデバイス54は、例えば、クロック周期の50~62%の間で始まって、バス64をアクティブローにプルし得る。バス64は、典型的には、クロック周期が完了する前に完全に安定化させられ得る。バス値は、クロック周期の終了間際で、例えば、クロック周期の88~100%で、第4の間隔においてサンプリングされ、その後、マスタデバイス52は、バス64を再度アクティブローに駆動し始めるであろう。一実施形態では、クロックサイクルの最初の半分が、クロック信号に使用され得る一方で、クロックサイクルの第2の半分は、データ伝送に使用され得る。しかしながら、例えば、クロック信号については67%、次いで、データ伝送については33%等の、クロックサイクルの第1の割合がクロック信号に使用され、クロックサイクルの第2の割合がデータ伝送に使用される、クロックサイクルの他の分割が考慮される。

【0092】

図4aの実施例では、バス64は、最初に、クロックサイクルの開始時にアクティブロ

10

20

30

40

50

ーである。特定の期間後、バス64は、マスタデバイス52によってアクティブハイに駆動される。スレーブデバイス54が「0」を送信することを希望すると、スレーブデバイス54は、バス64に何も行わない。次いで、「0」の値がマスタデバイス52によって読み取られ、マスタデバイス52の前のスレーブデバイス54が、次のクロックサイクルに先立ってバス64をアクティブローに駆動する。第2のクロックサイクル中に、マスタデバイス52は、特定の期間後にバス64をアクティブハイに駆動する。スレーブデバイス54が「1」を送信することを希望すると、スレーブデバイス54は、バス64をアクティブローにプルするか、または駆動し、次いで、値は、マスタデバイス52およびスレーブデバイス54によって読み取られる。バス64がアクティブローであると、マスタデバイス52は、バス64をアクティブローに駆動しないが、図4aに示されるように、その機能を果たし得る。第3のクロックサイクルでは、特定の期間後に、マスタデバイス52は、バス64をアクティブハイに駆動する。スレーブデバイス54が「1」を送信することを希望すると、スレーブデバイス54は、バス64をアクティブローにプルするか、または駆動し、次いで、値は、マスタデバイス52およびスレーブデバイス54によって読み取られる。第4のクロックサイクルの開始時に、バス64はアクティブローであり、特定の期間後、マスタデバイス52によってアクティブハイに駆動される。スレーブデバイス54が「0」を送信することを希望すると、値が読み取られた後まで、バス64上にアクティビティがなく、次いで、マスタデバイス52は、次のクロックサイクルに備えてバス64をアクティブローに駆動する。

【0093】

バスタイミングは、マスタデバイス52がバス64をアクティブローに駆動する期間中に、スレーブデバイス54がバス64をアクティブローにプルし続けることを可能にする。したがって、スレーブデバイス54の出力ドライバの解放タイミングは、あまり厳密ではない。しかしながら、スレーブデバイス54は、ライン遅延を含んで、マスタデバイス52がバス64をアクティブハイに駆動するときに、浮動出力を生成する。浮動期間がアクティブロー期間およびアクティブハイ期間より長い理由は、スレーブデバイス54がマスタデバイス52のタイミングに依存するので、スレーブデバイス54がマスタデバイス52と同一の時間量でバス64をアクティブローに駆動することを可能にし、起動が開始するとき、その瞬間に対してある公差を許可するためである。さらに、バス64上に遅延がある。

【0094】

バス64がマスタデバイス52によってローにプルされるときタイミングは、間違っただデータがマスタデバイス52またはスレーブデバイス54によってサンプリングされないように、特定の期間後である。したがって、スレーブデバイス54は、マスタデバイス52がバス64をハイにプルするときに浮動出力を有する。

【0095】

マスタデバイス52またはスレーブデバイス54が3状態になるとき、これは、たとえ実際の電圧がバス64に書き込まれた以前の値によって決定されようとも、鎖線によって表される。バス64が浮動する期間は、非常に短いため、バス64に連結された寄生静電容量により、安定して保たれ得るか、またはシステム50内に少なくとも1つのバスホルダを含むことによって、より長い期間にわたって安定して保たれ得る。したがって、バス64上の電荷および電圧は、静的にアンロードされるか、または高インピーダンスによってロードされる場合に、安定していると見なされ得る。

【0096】

スレーブデバイス54は、ある時間間隔でバス64を起動し、これは、PLL、遅延ロックスループ(DLL)、または単純遅延回路を使用して、ローカルクロックを実装することによって行われることができる。遅延回路は、固定電流によって交互に充電され、次いで、バス64が起動およびサンプリングされるべき点を決定するためにコンデンサ上の充電電圧を使用している、2つのコンデンサを備え得る。これら2つのコンデンサは、1つの固定クロックエッジから次の固定クロックエッジまで充電され、このサイクルを開始す

10

20

30

40

50

る前に放電され得る。情報が転送されていない長い期間中、クロック信号を高いレベルまでゲート制御することによって、バス64をアイドル状態にさせることが可能である。この場合、バス64を再起動するために、わずかな遅延が生じるであろう。マスタデバイス52がタイミングを制御するため、概して、いかなるアナログ制御遅延も使用されない。

【0097】

一般に、浮動または3状態期間中のバス64の状態の読み取りが一意的に定義されるように、十分に小さいバス静電容量、および十分に高いクロック周波数が使用される。バスシステム50がプリント回路基板上で実装される場合、いくつかの実施形態では、データエラーから保護するように十分大きい電荷を提供するために、小型コンデンサが使用され得る。この場合、小型であるが有限のバス静電容量が使用されるか、または代替として、安定したバス値を維持するようにバスホルダを使用して（例えば、背面連結された2つのインバータを使用して）実装される。この方法は、わずかな漏出電流がバス64上に残された任意の電荷をゆっくりと放電し得るが、バスホルダが値を安定して保つであろう、バス64の低周波数動作のために選択され得る。バスホルダは、通常、その動作に関連付けられる電力効率損失を制限するように、マスタデバイス52で実装されるであろうが、また、データ完全性目的でスレーブデバイス54に実装されることもできる。バス64が単一のワイヤを使用して実装されるときの実示的实施形態のタイミング特性が、表1に示されている。

【0098】

【表1】

表1:単一ワイヤバスのタイミング特性の実施例

| パラメータ | 最小 | 通常 | 最大 | 単位 |
|--------------------------------|---------|----|---------|-------|
| バス周波数(1.8mケーブル)= EXTCLK/分割器 | 1 | 6 | 12 | MHz |
| デューティサイクルクロック | 45 | 50 | 55 | % |
| 設定時間(スレーブデバイス) | | | 5 | ナノ秒 |
| ホールド時間(スレーブデバイス) | | | 5 | ナノ秒 |
| スルーレート(マスタデバイス)、充電 | VDD*0.1 | | VDD*0.5 | V/ナノ秒 |
| スルーレート、放電 | VDD*0.1 | | VDD*0.5 | V/ナノ秒 |

バス64上の最大妨害の推定は、2MHzのより低いクロック周波数が使用される場合（通常は6MHzクロック周波数が推奨される）に行われることができる。10 μ Aの最大漏出電流、1 μ 秒の期間、および100pFのライン静電容量を仮定すると、最大電圧降下は、約0.05Vに等しい。クロック周波数がさらに増加させられるとき、雑音排除性が増加する。マスタデバイス52の出力インピーダンスは、75 Ω 伝送線に合致する約75 \pm 12 Ω である。伝送線の端部で受け取られたとき、反射が、スルーレートにほとんど影響を及ぼさずに、即時に電圧を完全レベルまで上昇させるであろう。これらの特性があると、雑音排除性は、I²Cタイプ実装より良好であろう。長さ2メートルのケーブルについては、ケーブル遅延の推定値は、約10ナノ秒である（約20cm/ナノ秒の遅延）。バス64を実装するための75オームケーブルの使用は、50オームケーブルより長さにつき低い静電容量、およびより低い電気損失という利点を有する。単一のワイヤを使用するときのバス64の実示的实施形態の電気的特性が、表2に示されている。

【0099】

【表 2】

表2:単一のワイヤバスの電氣的特性の実施例

| パラメータ | 最小 | 通常 | 最大 | 単位 |
|---|------|------|------|-------|
| VDD | 1.62 | | 1.98 | ボルト |
| VL, in | | | 0.65 | ボルト |
| VH, in | 1.40 | | | ボルト |
| Iload(バス停止、同期オフ、電カトランジスタオン)、Vdrop=300mV (POWER MODE) | | | 30 | mA |
| バスホルダ出力インピーダンス (マスタデバイス) | 600 | 1200 | 1800 | Ω |
| 75pF以下の負荷を伴う(マスタデバイスにおける)スルーレート制限 | 0.3 | 0.6 | 1.0 | V/ナノ秒 |
| ドライバの出力インピーダンス | 63 | 75 | 87 | Ω |

2ワイヤ実装を有するバス上で統一バス通信プロトコルを実装するとき、クロックとデータ情報とを別個のワイヤ(すなわち、クロックラインおよびデータライン)上で伝送することができる。この場合、遅延回路は、スレーブデバイス54の内側で使用されなくてもよく、両方のクロックエッジを使用する内部タイミング回路は、より良好なジッタ性能を有するであろう。このバス構造は、ピン要件があまり厳格ではない、PCB上の実装に適している。実施形態では、信号伝達するために使用される方法は、NRZI信号伝達であり得、すなわち、論理ゼロが、バス64上のレベルを維持することによって信号伝達される一方で、論理1は、バス64のレベルを変化させることによって信号伝達される。

【0100】

ここで図3cを参照すると、2ワイヤバスシステムの例示的实施形態が示されている。本実施例では、マスタデバイス1は、低速シリアルインターフェースを通して、I²Cインターフェース、TDMインターフェース、ビットストリームインターフェース、およびI²Sインターフェースに連結される。次いで、マスタデバイス1は、本明細書で説明される統一バス通信プロトコルの実施形態のうち少なくとも1つに従って動作し得る、2ワイヤバスを介して、スレーブデバイス1、スレーブデバイス2、およびマスタデバイス2に連結される。2ワイヤバスは、クロック情報を搬送する第1のワイヤと、データ、同期化、およびコマンド情報を搬送する第2のワイヤとを有する。本実施例では、スレーブデバイス1および2ならびにマスタデバイス2の各々は、低速シリアルインターフェースを介してバスに連結される。スレーブデバイス1はまた、第2のビットストリームインターフェースにも連結される。スレーブデバイス2はまた、第2のI²Sインターフェースにも連結される。マスタデバイス2はまた、第2のI²Cインターフェースおよび第2のTDMインターフェースにも連結される。

【0101】

別個のクロックおよびデータラインを伴う2ワイヤバスが使用されるとき、クロック信号は、DATAライン上で信号伝達されるような情報が有効であるか、または有効ではないときについてのタイミング情報を提供するにすぎないであろう。場合によっては、情報は、単一のクロックエッジ上で、または両方のクロックエッジ上で転送され得る。場合によっては、DATAライン上の情報は、クロックラインの電力消費を削減するために、単一のクロックサイクル内で2回以上変化し得る。

【0102】

ここで図4bを参照すると、2本のワイヤを使用してバス64が実装されるとき例示

10

20

30

40

50

的タイミング図が示されている。2本のワイヤとともにプロトコルを使用するとき、1つのクロックエッジ上のみ（立ち上がりまたは立ち下りのいずれか）で、または両方のクロックエッジ上でデータを転送することが可能である。両方のクロックエッジを使用することの利点は、より高い帯域幅およびより低い電流消費を可能にすることである。しかしながら、インターフェースの物理的特性が選択されると、バス半径に制約がある。1つだけのクロックエッジが使用される場合、クロック周波数が低減される限り、非常に長いバス半径を使用することが可能である。しかしながら、2つのクロックエッジが使用される場合、マスタデバイス52の近くに位置するスレーブデバイスは、マスタデバイス52からより遠く離れて位置するスレーブデバイスからサンプリングされる前にデータを出力し始め得る。この問題を解決する1つの方法は、以前のデータスロット中で3状態にさせられていたデバイスに対してデータがバス64で有効にされる前の時間が、デバイスがデータ状態から3状態になるために要する時間より長いことを確実にすることである。さらに、データがクロックエッジ上で即時にバス64から時間記録されている間、バス64へのデータの出力クロッキングを少し遅延させることによって、タイミングのマージンが向上させられ得る。いくつかの実施形態では、高い周波数、または大型バス直径での動作を可能にするために、タイミングのマージンは、クロック周波数に依存するように調整され得る。例として、データは、より低い動作周波数のために、クロック遷移後に、より長時間にわたって安定して保たれることができる。他の場合において、タイミングのマージンの固定パラメータを使用し、2つのスレーブデバイスが異なる時間にクロック信号を受信するときの問題（マスタデバイス52の近くに位置する1つのスレーブデバイスに、マスタデバイス52から遠く離れて位置するスレーブデバイスからの結果を上書きさせ得る）を回避するために、大型バス直径を伴うシステムについて、1つおきのチャンネルが空である（効果的に3状態にさせられている）ビットストリームモードを使用することが決定され得る。代替実施形態では、スレーブデバイス54をバス64に接続するために、例えば、リング構造またはパイプ構造等の異なる構造を使用することができる。これらの構造は、図71から76に関してより詳細に説明される。いくつかの実施形態では、マスタデバイス52は、パイプまたはリング構造への直接接続を含み得る。他の実施形態では、ハブデバイスが、1ワイヤまたは2ワイヤ多重ドロップバスと代替的なパイプまたはリングトポロジーとの間のインターフェースの役割を果たし得る。

【0103】

この2ワイヤバスの実装は、 I^2C 実装および同様のオープンコレクタ構造実装と比較して電力消費を削減し、 I^2C および同様のオープンコレクタ構造実装の場合のように、帯域幅と電力消費との間の妥協を推断しない。場合によっては、統一バス通信プロトコルを使用するバス64の2ワイヤ実施形態は、SLIMbusより40%低い電力消費、および2倍の帯域幅をもたらす。さらに、いくつかの実施形態では、統一バス通信プロトコルを実装するために、SLIMbusで使用されるのと同じ物理層を使用することができる。

【0104】

少なくともいくつかの実施形態では、バスホルダをマスタデバイス52で実装することができる。これは、スレーブデバイスがほとんど存在しない状況および多くのスレーブデバイスが存在する状況で、予測可能な雑音性能をもたらす。バス64に接続されたデバイスに電力が印加されないとき、たとえ電力を伴わないデバイスがバス64に依然として接続されていたとしても、バス64を動作させることが依然として可能であるべきである（例えば、電力が印加されていないときに、（静電放電）ESD構造がバス64を停滞させるべきではない）。2本のワイヤを使用するときのバス64の例示的实施形態の電気的特性が、表3に示されている。

【0105】

10

20

30

40

【表 3】

表3:2ワイヤバスの電氣的特性の実施例

| パラメータ | 最小 | 通常 | 最大 | 単位 |
|---------------------------------|---------|------|---------|-------|
| バス周波数(0.30cmPCB)= EXTCLK/分割器 | 0 | 10 | 20 | MHz |
| クロック半期(高または低であるが 両方ではない) | 23 | | | ナノ秒 |
| 設定時間 | | | 2 | ナノ秒 |
| ホールド時間 | | | 2 | ナノ秒 |
| バスホルダ(マスタデバイスのみ) | 600 | 1200 | 1800 | オーム |
| スルーレート | VDD*0.1 | | VDD*0.5 | V/ナノ秒 |
| 出力を無効にする時間、 T_{D-Z} | | | 5 | ナノ秒 |
| 出力を有効にする時間、 T_{Z-D} | 6 | | 12 | ナノ秒 |
| サンプリング前のデータライン上の 内部遅延 | 1 | 2 | 3 | ナノ秒 |

電力消費に関して、データが2ワイヤバス実施形態について1つのクロックエッジ上で変化すると仮定される場合、容量充電による動的電力消費は、2ワイヤバス実施形態と比較して、単一ワイヤバス実施形態について、より低くなるであろう。伝送されたビットにつき使用される状態変化が、単一ワイヤバス実施形態については2となるであろう(すなわち、信号が上昇または下降する)一方で、2ワイヤバス実施形態は、上昇および下降するクロック信号、および平均で1回おきに变化するデータ信号により、平均して $2 + (1/2)$ の信号変化を使用するであろう。単一ワイヤバス実施形態は、信号がデューティサイクル変調され得るため、いかなる余分な電力消費も伴わずに(すなわち、ラインの余分な充電がない)、追加のデータ情報がクロック信号中で搬送されることを可能にする。これは、端子の数を削減するため、および信号遷移の数を削減することによって電力消費を削減するために、クロックおよびデータラインが単一のラインに組み込まれるときの例である。他の実施形態では、単一ワイヤバスはまた、クロックサイクルにつきより多くのビットを転送するために、レベル変調され得る。しかしながら、データが2ワイヤバス実施形態の両方のクロックエッジ上で変化する場合には、2ワイヤバス実施形態が、伝送される全ビットに対して $1 + (1/2)$ の状態変化(すなわち、1クロック変化および $1/2$ データ信号変化の平均)のみを使用するであろうため、それは、より電力効率的となるであろう。

【0106】

電力効率が、バス64にアタッチされたデバイスの数とともに直線的に劣化するため、可能な限り少ないデバイスをバス64に接続させることが、電力に関してより効率的なはずであることに留意されたい。複数のデバイスが単一のバスラインにアタッチされ得る場合、結果として生じる電力消費を低下させるように、各デバイスの入力静電容量を低減させることが有利である。代替として、デバイスのうちのいくつかが1つのバスに連結され、他のデバイスが別のバスに連結され、マスタデバイス等の両方のバスに共通するデバイスがあるように、1つより多くのバスが使用される実施形態があり得る。

【0107】

一般に、スレーブデバイス54は、PING動作中に信号を能動的にプルダウンすることによって、マスタデバイス52と通信することを希望することを示すことができる。次いで、この場合、READ、WRITE、またはFUNCTION動作を介して、スレーブデバイス54とマスタデバイス52との間でデータを伝送することができる。マスタデ

バイス 5 2 はまた、独力で R E A D、W R I T E、または F U N C T I O N 動作を開始することもできる。通信を開始するために、マスタデバイス 5 2 およびスレーブデバイス 5 4 は、以下でさらに詳細に説明されるように、互に同期化される。

【 0 1 0 8 】

統一バス通信プロトコルの例示的实施形態では、フレームは、3つの制御ワード、すなわち、同期化ワード(すなわち、Sワード)、および、スレーブデバイスへおよびスレーブデバイスからのコマンドを通信するための2つのコマンドワード(すなわち、XワードおよびYワード)を含むように定義される。制御ワードは、長さ16ビットであり得、概して、全フレーム中に送信される。他の実施形態では、制御ワードの長さは、何らかの他の長さ、例えば、8ビットであり得、フレーム中の制御ワードの数は、3とは異なり得る。しかしながら、制御ワードは、動作モード、すなわち、ワードモードおよびビットストリームモードに応じて、フレーム中に異なって伝送されることができる。例えば、ワードモードでは、異なるオーバーサンプル率およびクロック周波数のサポートを可能にする、制御ワードの間の距離が変化し得る。ビットストリームモードでは、データビットは、各制御ワードにおいて個々のビットの間にインターリーブされ、このデータインターリーブの1つの利益は、複数の低遅延ビットストリームデータチャネルのサポートである。

10

【 0 1 0 9 】

例示的实施形態では、マスタデバイス 5 2 は、状態監視(すなわち、P I N G 動作)、レジスタを読み取ること(すなわち、R E A D 動作)、レジスタに書き込むこと(すなわち、W R I T E 動作)、および関数を実行すること(すなわち、F U N C T I O N 動作)を含む、いくつかの制御機能を果たす。動作のタイプは、マスタデバイス 5 2 によって決定される。いくつかの実施形態では、マスタデバイス 5 2 は、システムコントローラ(図示せず)によって制御される。これらの場合、マスタデバイス 5 2 とシステムコントローラとの間のインターフェースは、I²Cバスを使用して、またはこれら2つのシステム構成要素を1つのデバイス(すなわち、プロセッサ)に組み込むことによって実装されることができる。

20

【 0 1 1 0 】

スレーブデバイス 5 4 と通信するために、マスタデバイス 5 2 は、スレーブデバイス 5 4 (または複数のデバイス)が、それらの内部タイミングまたはクロックをマスタデバイス 5 2 と、および結果としてバス 6 4 と同期化するために使用し得る、同期化データを有する同期化(すなわち、同期またはS)ワードを伝送する。一般に、マスタデバイス 5 2 がフレーム中で伝送する第1の制御ワードは、Sワードである。したがって、Sワードは、フレームを開始するために使用されることができる。伝送される第1のビットは、S 1 4 ビット等が後に続く、S 1 5 ビットである。ワードモードにおいて、ビットが互の直後に伝送されるであろう一方で、ビットストリームモードにおいて、これらのビットの伝送の間に一定の間隙が生じるであろう。入出力(I/O)方向(例えば、r/w)が、マスタデバイス 5 2 に関して与えられる。

30

【 0 1 1 1 】

ここで図 5 を参照すると、Sワードのフィールドおよびビット割付の例示的实施形態の略図が示されている。本実施例では、Sワードは、4つのフィールド、すなわち、D A T A S Y N C H R O N I Z A T I O N フィールド、9ビット C O N S T A N T S Y N C S Y M B O L フィールド、D Y N A M I C S Y N C S Y M B O L フィールド、および I N T E R R U P T フィールドから成る。他の実施形態は、Sワードに他のフィールド、または図 5 に示されるフィールドより多いまたは少ないビットを使用し得る。これらのフィールドは、レジスタの1つ以上のビットを使用して実装されることができる。代替実施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。

40

【 0 1 1 2 】

Sワードの D A T A S Y N C H R O N I Z A T I O N フィールドは、パリティビット

50

PAR(S15)と、確認ビットACK(S14)とを含む。パリティビットの計算は、新しいパリティビットを書き込む前に、最後のビットまでを含んで書き込まれる最後のパリティビットを含む、バス64上で遭遇する全てのトラフィックに基づく(すなわち、パリティが完全フレームについて計算され得る)。パリティビットは、フレーム中の1に等しいビットの総数が偶数である場合に1であるように設定され、そうでなければ、パリティビットは、ゼロであるように設定される。電源投入時のマスタデバイス52の初期パリティ値は、ゼロである。少なくとも1つの実施形態では、マスタデバイス52は、バス64にアタッチされる任意のデバイスが、信号完全性問題またはデータ衝突を検出できることを確実にするように、全ての動作(PING、READ、WRITE、およびFUNCTION)に対してパリティ情報を書き込み得る。

10

【0113】

確認ビットは、最後のフレーム中のバス64上のトラフィックに基づいて計算される。スレーブデバイス54は、スレーブデバイス54へのREAD、WRITE、およびFUNCTION動作中にデータ伝送が破損された場合に、論理低を信号伝達するべきである。パリティにエラーがない場合、スレーブデバイス54は、論理1で応答するであろう。これは、I/O WRITE OPERATIONS中の追加セキュリティのために有用である。スレーブデバイス54は、エラーが発生した場合、例えば、パリティビットが間違っていた、またはバスエラーが発生したことを把握しているので、スレーブデバイス54は、過去の動作に基づいて確認ビットを設定する。スレーブデバイスがバス64上に存在しない場合、問題を示すバス信号伝達スキームの性質によって、論理低がバス64からマスタデバイス52に返されるであろう。以前の動作がPING動作であった場合、パリティビットおよび過去のフレームのアクティビティ全体に基づいてエラーを検出した任意のスレーブデバイスは、このビットをアクティブにし得る(すなわち、論理1を書き込む)。これは、確認ビットがPING動作中に効果的に反転させられることを意味する(論理ゼロはエラーなしを意味する)。エラーが検出された場合、マスタデバイス52は、READ、WRITE、およびFUNCTION動作後の欠落したACKビットについてはIO ERRORビット、PING動作中のNACKについてはBUS ERRORをアクティブにし得る。ワードモードでのSワードのDATA SYNCRHONIZATIONフィールドへの動作を示すタイミング図である、図6に示される実施例では、マスタデバイス52は、タイムスロット0-95中のアクティビティに基づいて、タイムスロット96に偶数パリティを書き込む。スレーブデバイス54は、タイムスロット97中で応答する。READ、WRITE、およびFUNCTION動作中に、スレーブデバイス54は、エラーの場合に応答する(すなわち、NACKによって)のみであろう。

20

30

【0114】

ここで図7を参照すると、異なるコマンド動作に基づいてマスタデバイス52およびスレーブデバイス54によって行われる、パリティおよび確認計算の例示的实施形態が示されている。パリティにエラーがない場合、アドレス指定されているデバイスが、論理1で応答する(すなわち、アクションをアクティブにする)であろう。これは、I/O WRITE動作中の追加セキュリティのために有用である。マスタデバイス52からスレーブデバイス54へのWRITE動作中にパリティエラーがある場合、スレーブデバイス54は、パリティエラーがあれば内部でWRITE動作を完了せず、マスタデバイス52は、IO ERROR割り込みでバスコントローラに応答するであろう。スレーブデバイス54からのREAD動作中にパリティエラーがある場合、マスタデバイス52がIO ERROR IRQを生成するであろう一方で、スレーブデバイス54は、マスタデバイス52がアクティブACKを受信しなければ何もしないであろう。デバイスレジスタおよび最後に書き込まれたワードを読み取ることによって、どのデバイスが問題になっているかを識別することが可能である。新しいREAD、WRITE、またはFUNCTION動作を行う前に、一実施形態では、現在のIO動作が完了するまで待機すること、およびIO ERRORビットの状態をチェックすることが可能である。この方法論は、よりロバストなエラー性能を提供する。

40

50

【 0 1 1 5 】

したがって、統一バス通信プロトコルの少なくとも1つの実施形態の側面は、フレーム内の伝送される全てのデータを監視し、伝送されるデータのパリティを計算し、動作を行うためのコマンドを受信したバス64にアタッチされた1つ以上のスレーブデバイスからのREAD/WRITE/FUNCTION動作の確認をチェックすることによって、データ完全性サポートを提供することである。マスタデバイス52は、監視し、パリティを計算し、およびチェックするアクションを行うことができる。

【 0 1 1 6 】

いくつかの動作中、例えば、PING動作中に、特に1つのデバイスもアドレス指定されない。したがって、任意のデバイスがパリティエラーを検出することができる。次いで、マスタデバイス52は、フレームの読み取りを完了すると、パリティビットに書き込み、パリティエラーを検出する任意のデバイスは、NACK（否定応答）と同一の有効な機能を有する、論理1で応答するであろう。これは、1ワイヤおよび2ワイヤ実装の両方におけるバス64が接続または機能性を可能にするので、複数のデバイスが同時に応答することを可能にする。一般に、特定のデバイスまたはデバイス群へのREADおよびWRITE動作のように扱われ得る関数データを使用するコマンドを除いて、FUNCTION動作は、PING動作のように扱われ得る。

【 0 1 1 7 】

したがって、統一バス通信プロトコルの少なくとも1つの実施形態の側面は、フレーム内の伝送される全てのデータを監視し、伝送されるデータのパリティを計算し、動作を行うためのコマンドを受信した、バスにアタッチされた1つ以上のスレーブデバイスからのREAD、WRITE、またはFUNCTION動作以外の動作に対する否定応答メッセージをチェックすることによって、データ完全性サポートを提供することである。マスタデバイス52は、監視し、パリティを計算し、およびチェックするアクションを行うことができる。

【 0 1 1 8 】

少なくともいくつかの実施形態では、フレーム内のほとんどの誤った同期化位置を迅速に排除するために、CONSTANT SYNC SYMBOLフィールド（S13：S5）がスレーブデバイス54によって使用されることができる一方で、現在の位置が確かにフレームの開始であることを検証するために、DYNAMIC SYNC SYMBOLフィールド（S4：S1）が使用される（本明細書で使用される命名法S13：S1は、ビットS13、S12、S11、S10、S9、S8、S7、S6、S5、S4、S3、S2、およびS1を示すことに留意されたい）。このアプローチは、迅速なロック時間を得て、スレーブデバイス54が、バス64上のランダムな静的データによって生成される誤った開始にロックすることを防止するために役立つために使用される。例えば、次の9ビット{S13：S5} = “101100011”は、スレーブデバイス54によってランダムバスデータ上で迅速な初期同期化を取得するために使用されることができる定数である。したがって、Sワードの一定同期部分は、フレーム中の誤ったSワードの可能性を低減させるために、スレーブデバイス54によって使用される。例えば、Sワードであるように見えるデータが、他のスレーブデバイスによってバス64を経由して伝送され得るが、マスタデバイス52によって伝送されていないので、それは、誤っており、接続されていないスレーブデバイスによって無視されるべきである。したがって、Sワードの一定部分（すなわち、CONSTANT SYNC SYMBOLフィールド）の値の選択は、Sワードとしてのその使用が損なわれないように、デジタルワード、オーディオデータ、またはビットストリームデータにおいて通常発生しないビットのパターンであるようなものである。

【 0 1 1 9 】

DYNAMIC SYNC SYMBOLフィールド（S4：S1）は、例えば、スレーブデバイス54によって同期化にも使用される、ジェネレータ多項式 $X^4 + X + 1$ 、 $X^8 + X^5 + X^4 + 1$ 、または $X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$ を用

10

20

30

40

50

いて、周期的冗長チェック（CRC）ジェネレータ等の疑似ランダムカウンタによって実装されることができ、疑似ランダム値を備える。例えば、DYNAMIC SYNC SYMBOLフィールドは： $\{S4, S3, S2, S1\} = \{S3, S2, S1, (S4 \oplus S3)\}$ （最上位ビット（MSB）が最初）のように更新されることができ、本実施例では、DYNAMIC SYNC SYMBOLフィールドは、リセット時に「1111」に初期化される。「0000」条件は使用されず、遭遇した場合には次のサイクル中に「1111」状態をもたらすものとする。16進数でのシーケンス全体は、以下の通り、すなわち、 $\{F, E, C, 8, 1, 2, 4, 9, 3, 6, D, A, 5, B, 7\}$ である。

【0120】

少なくともいくつかの実施形態では、疑似ランダム値はまた、同期化にも使用されることができ、これは、より迅速なロック時間を可能にし、スレーブデバイス54がバス64上のランダムデータによって生成される誤ったワードにロックする可能性を低減させるであろう。少なくとも1つの実施形態では、スレーブデバイス54は、バス64にロックする前に、疑似ランダムパターンを複数回検証することができる。少なくとも1つの実施形態では、スレーブデバイス54は、バス64上で通信することを可能にする前に、同期生成サイクル全体を検証するまで待機することができる。マスタデバイス52との同期化を取得したことを決定するまで、スレーブデバイスが検出に使用し得る同期化パターンの数は、誤った同期化のエラーのどれほど低い確率が容認可能であるかという問題である。例えば、ロックが容認される前に、スレーブデバイス54がN個のフレームを使用する場合には、これがランダムデータによって起こる可能性は $1/5^N$ であり、これは、非常に小さい確率であり、一定同期部分は、通常は一樣に離間され、同期化中にも見出されるので、実践ではさらに低い。したがって、同期化（すなわち、ロック）が安全と見なされることを宣言する前に、より少ないまたは多数の同期化パターン（したがってフレーム）が、異なる用途で使用され得る。代替実施形態では、スレーブデバイス54は、バス64にロックするために7つのフレームを使用し得る。他の実施形態では、異なる数のビットを一定同期部分および動的同期部分に使用することができる。代替実施形態では、DYNAMIC SYNC SYMBOLフィールドのコンテンツを生成するために、他の技法を使用することができる。

【0121】

例えば、デジタルワードデータの伝送、ビットストリームデータの伝送等のためのバス64の使用に応じて、CONSTANT SYNC SYMBOLフィールドおよび/またはDYNAMIC SYNC SYMBOLフィールドの値を生成するために、異なる値またはスキームが使用され得る。実施例として、ワードに対する一定部分は、バス64がワードモードで使用されるときには $0 \times B25$ （16進数）であり得、ワードに対する一定部分は、バス64がビットストリームモードで使用されるときには $0 \times FFE$ であり得る。

【0122】

INTERRUPTフィールド（IRQS(S0)）は、優先的な割り込み制御を実装するためにワード中に含まれ、多くの使用シナリオに対する割り込み要求への高速応答を可能にすることができる。スレーブデバイス54は、フレーム中にマスタデバイス52への割り込みを生成することができる。スレーブデバイス54は、即時注意を求める場合、INTERRUPTフィールド中で論理「1」レベルを信号伝達し、これは、この例示的实施形態ではワードの最後のビット中である。IRQビットは、一般的な割り込み制御として使用され、任意のフレーム中に任意のスレーブデバイスによってアクティブにされることができる。例えば、スレーブデバイス54が送信すべき重要な状態メッセージを有する場合、IRQSビットは、READ、WRITE、またはFUNCTION動作を遅延させるために使用されることができ、これは、INTERRUPTフィールド中のビットをアクティブにすることによって行われる。いかなるスレーブデバイスもこのビットをアクティブにしていない場合（論理「1」がアクティブにすることを意味する）、ま

10

20

30

40

50

たは I R Q S 中断が I R Q マスクビット B 4 によって無効にされている場合 (S 0 D E L A Y E N A B L E、図 2 9 a および 2 9 b 参照)、R E A D、W R I T E、または F U N C T I O N 動作が継続するであろう。この遅延は、スレーブデバイス 5 4 が「1 0」または「1 1」に等しい状態レベルを有する場合にアクティブにされる(これは以下でさらに詳細に説明される)。したがって、スレーブデバイス 5 4 は、フレーム中のこの I R Q S ビットに対するタイムスロット中に、その状態レジスタ(スレーブ状態ビットを保持する)の M S B のコンテンツをバス 6 4 上にコピーすることができる。スレーブ状態ビットは、X および Y ワードの説明でさらに議論される。X および Y ワードの部分中にスレーブ状態データを含むことにより、他のスレーブデバイスからのデータが収集されることを可能にする一方で、スレーブデバイスのうちの 1 つがエラーを有する場合にバス 6 4 を無効にしない。一方で、I R Q S ビットの値が非アクティブである場合、または I R Q が I R Q マスクビットによって決定されるように許可されない場合(その場合、バス 6 4 に連結されたバスの状態は、P I N G 動作を使用してポーリングされる)、R E A D、W R I T E、または F U N C T I O N トランザクション/動作は、継続することを可能にされる。図 8 は、コマンド動作(P I N G、R E A D、W R I T E、および F U N C T I O N)、割り込みマスクビットの値、および講じられるべき措置の種々の組み合わせを示す。

【 0 1 2 3 】

フレーム内の制御動作のタイプ(すなわち、P I N G、R E A D、W R I T E、または F U N C T I O N 動作)は、この例示的实施形態では X ワードの最初の 3 つのビットによって決定される。これらのビットの後には、特定の動作によって使用される情報が続く。P I N G 動作の場合、レジスタバンク選択ビットが、スレーブデバイス 5 4 用の構成モードを決定するために使用され得、オーディオ用途については、これは、異なるオーディオモード間のシームレスな転送に使用することができるオーディオモードと称されることができ、この転送は、結果として生じるオーディオで不可聴である。データ同期化ビットは、スレーブデバイス 5 4 中の少しのハードウェアを用いて、奇数サンプルレートおよび非同期転送をサポートするように含まれ得る。後続のビットは、ペアでグループ化され、バス 6 4 にアタッチされた全てのデバイスの現在の状態を読み取るために使用される。他の実施形態では、別のマスタデバイスによってアクティブにすることができる R E Q U E S T F O R B U S ビット、およびバス 6 4 へのアクセスを許可する R E L A S E B U S ビットを含むことによって、マルチマスタ動作に対するサポートが達成される。このタイプの動作では、両方のマスタデバイスが同時に X および Y ワードでコマンドを書き込むことができるため、真のマルチマスタ動作が可能である。バス衝突がある場合、ゼロを書き込むが 1 をリードバックするデバイスが、アービトレーションを失うであろう。P I N G 動作を「0 0 0」に割り付けることによって、真のマルチマスタ動作をバス 6 4 の連続監視と組み合わせることが可能である(すなわち、P I N G 動作を行うマスタデバイスは、別のマスタデバイスが R E A D / W R I T E または F U N C T I O N を要求する場合にアービトレーションを失うであろう)。

【 0 1 2 4 】

R E A D、W R I T E、および F U N C T I O N 動作は、情報をバス 6 4 にアタッチされた 1 つの特定のデバイスに送信する。固有のスレーブアドレスに割り付けられたビット数は、この例示的实施形態では 4 ビットである。他の実施形態は、スレーブアドレスにより多いまたは少ないビットを使用することができる。実際のアドレスは、アドレス衝突が起こることを防止するように、すなわち、(通常は最大 1 1 個のデバイスが使用されるが)多くても 1 6 個のデバイスが同時にバス 6 4 にアタッチされている限り、電源投入後に動的に割り当てられることができる。4 ビットの後には、レジスタアドレスおよびデータ情報が続く。アドレスおよびデータフィールド長は、概して、動作のタイプ(8 または 1 6 ビットデータ)に依存し、Y ワードに及ぶであろう。マスタデバイス 5 2 は、現在のデータ転送に基づいて計算したパリティを書き込み、スレーブデバイス 5 4 は、動作が成功したことを確認する。マスタデバイス 5 2 およびスレーブデバイス 5 4 の両方が、他方のデバイスが正しいデータの受信に失敗したかどうかを知るであろうため、このパリティ検

10

20

30

40

50

証のためのアプローチは、余分なセキュリティを与える。

【 0 1 2 5 】

F U N C T I O N動作は、種々の目的で使用されることができる。例えば、少なくとも1つの実施形態では、F U N C T I O N動作は、ある状態にスレーブデバイス54を設定するため、特定のデバイスについての情報を要求するため、または試験のために使用される。

【 0 1 2 6 】

ここで図9aを参照すると、その中に、種々のコマンド動作のためのXコマンドワードに対する種々のフィールドおよびビット割付が示されている。これらのフィールドは、レジスタの1つ以上のビットを使用して実装されることができる。代替実施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。読み取り(r)および書き込み(w)方向は、マスタデバイス52に対して定義され、すなわち、(w)は、データフローの方向(すなわち、転送または伝送)が、マスタデバイス52からスレーブデバイス54であることを意味し、(r)は、(w)動作と比較してデータフローの反対方向を意味する。Xワードの構造は、どのような動作が現在進行中であるかに依存するであろう。図9aは、いくつかの例示的組み合わせを提供する。本実施例では、Xコマンドワードは、COMMANDフィールドと、2つのDATA SYNCHRONIZATIONフィールドと、SLAVE DEVICEフィールドと、BANK SELECTフィールドと、ADDRESSフィールドと、SLAVE STATUSフィールドと、FUNCTIONフィールドとを備える。フィールドのうちのいくつかは、説明されるように、あるコマンド動作中にのみ使用される。FUNCTION動作が以下で議論されるときに、FUNCTIONフィールドを説明する。一般に、最初の3つのビット(X15-13)が、READ、WRITE、PING、FUNCTION、またはOTHERコマンド等のマスタデバイス52によって発行されているコマンドを示す、コマンドフィールドの一部である一方で、残りのビットの構造は、選択されたコマンドに依存するであろう。図9bは、使用され得るXワードの代替実施形態を提供する。この符号化は、BANKビットの復号および起動におけるいくつかの複雑性を犠牲にして、任意の動作モード(PING/READ/WRITE/FUNCTION)で非同期、分数、またはマルチフレームデータ転送を行うことが可能であるという利点を有する。

【 0 1 2 7 】

COMMANDフィールド(X15:X13)に関して、少なくともいくつかの実施形態では、コマンドは、この例示的実施形態では8または16ビットREAD/WRITE、PING動作、またはFUNCTION動作であり得る(より多いまたは少ないビットが他の実施形態で使用され得る)。一般に、PINGコマンドは、マスタデバイス52がREAD、WRITE、またはFUNCTIONコマンドを受信し、それを現在処理していない限り、常にアクティブである。

【 0 1 2 8 】

いくつかの実施形態では、X12ビットは、PING動作中に使用されるBANKビットまたはBANKフィールドであり得る。PINGコマンドが発行されるとき、マスタデバイス52は、データ転送の内部タイミングのために、どのレジスタバンクが全てのデバイスに使用されるべきかを示すためにBANKビットを使用する。オーディオ用途では、BANKフィールドは、2つのレジスタバンクの間で切り替えて、オーディオモードを変更しながら再生中にいかなる誤作動も回避するために使用されることができる。一般に、BANKフィールドは、フレーム中のデータ伝送に使用することができる、異なるデータモードに対する異なるフレーム形式を特定する、異なるレジスタバンクの間で切り替えるために使用されることができる。異なるデータモードは、異なるオーディオシナリオに対して、異なるサンプリングレート、異なるチャンネル選択、異なるポート選択、ビットストリームモードまたはワードモードを使用すること等を特定することができる。例えば、1つのデータモードでは、データチャンネルからのデータは、フレーム中のタイムスロットの

10

20

30

40

50

第1のセット中に配置されるように定義されることができ、第2のデータモードでは、同一のデータチャネルからのデータは、フレーム中のタイムスロットの第2のセット中に配置されるように定義することができる。PING動作を使用しないフレームに、BANKフィールドの以前の値が、レジスタ選択に使用され得る。スレーブデバイス54は、アドレス指定されているとき、どのレジスタバンク上でREAD、WRITE、またはFUNCTION動作を行うかを選択するために、BANKフィールドを使用する。スレーブデバイス54は、1つより多くのデータ動作モードで変化する、全てのレジスタに対する少なくとも1つのシャドウレジスタを使用し得る。それは、内部レジスタおよび/または内部プログラムの間で選択するために使用され得る。エラーに対するロバスト性を提供するために、BANKフィールドは、変化が起こる前に、同一の値を伴って、2回等の1回以上読み取られることができる。

10

【0129】

少なくとも1つの実施形態では、エラーに対するロバスト性を提供するために、BANKSELECTフィールドは、変化が起こる前に、少なくとも2つの連続フレーム中のこのフィールド内に同一の値があることを確実にするために、2回等の1回以上読み取ることができ、すなわち、本実施例では、レジスタバンクの変化は、少なくとも2つのフレームを要するであろう。

【0130】

SLAVEDEVICEフィールド(X11:X8)は、READ、WRITE、またはFUNCTION動作中に使用される。SLAVEDEVICEフィールドは、バス64にアタッチされるスレーブデバイスのアドレスを特定する。全てのスレーブデバイスは、電源投入後にゼロのアドレスを使用して、マスタデバイス52と通信し始め、その後、それらは、マスタデバイス52によって新しいアドレスを割り当てられるであろう。アドレス0-12を伴うスレーブデバイスは、完全状態情報をマスタデバイス52に通信することができる。この例示的实施形態では、アドレス15は、バス64にアタッチされるデバイスの全てへの情報のブロードキャストのために保留される。このブロードキャストは、例えば、全てのスレーブデバイスをリセットするように1つのコマンドを使用するために有用であり得る。

20

【0131】

DATA SYNCHRONIZATIONフィールド(X11-10中のDS0-DS1)は、いくつかのフレームにわたってデータを同期化するために使用され得、PING動作中に伝送される。スレーブデバイス54は、データが伝送され始めるときを選択する、内部カウンタの開始を同期化するために、このフィールドを選択し得る。スレーブデバイス54がデータ同期化ビットDS0またはDS1を使用する場合、それが1に設定される場合、スレーブデバイス54は、その内部カウンタをリセットし(通常のフレーム開始時に起こり得る)、そうではなく、DS0またはDS1ビットが0に設定される場合、これらの内部カウンタはリセットされない。これは、例えば、フラクショナルフローに対して、いくつかのフレームに最良適合するデータの伝送を同期化するために有用である。

30

【0132】

ビットREQUEST(REQ)BUSACCESS(X9)は、PING動作中に使用される。このビットをアクティブにする(それを1に等しく設定する)ことによって、別のマスタデバイスは、バス64へのアクセスを要求することができる。

40

【0133】

ビットRELEASE(REL)BUS(X8)は、PING動作中に使用される。現在のマスタデバイスがこのビットを起動する場合(反転論理については論理値0)、別のマスタデバイスが、コマンドをバス64に送信し始めることを可能にされる。現在のマスタデバイス52は、そうするように選択する場合、コマンドを同時に送信し得る。データアービトレーションは、2つのマスタデバイス間に競合がある場合、どちらが特定のコマンドに対する要求を獲得するかを決定するであろう。別のマスタデバイスがRELEASEBUSビットを無視し、このフィールドが非アクティブ(例えば、論理1)である

50

ときにコマンドワードに書き込み始める場合、これは、バスエラーと見なされるであろう。

【 0 1 3 4 】

代替実施形態では、X 1 2 - X 9 ビットは、上記で説明されるようにデータ同期化 (D S 0 - D S 3) に使用されることができる。さらに、X 8 ビットを上記で説明されるように B A N K ビットとして使用することができる (例えば、図 9 b 参照)。

【 0 1 3 5 】

A D D R E S S フィールド (X 7 : X 0) は、R E A D、W R I T E 動作、およびいくつかの F U N C T I O N 動作中に使用され、スレーブデバイス 5 4 の内側で特定のレジスタアドレスを選択するために使用される、下位 8 ビットアドレスである。1 6 ビット動作が使用される場合、上位 8 ビットは、ゼロであると仮定され得る。

【 0 1 3 6 】

S L A V E S T A T U S フィールド (X 7 : X 0) は、P I N G 動作中に使用され、スレーブデバイス S T 1 1 から S T 8 の部分に対する状態を提供する (残りの状態情報は、説明されるように Y ワードで提供される)。この例示的实施形態では、全てのスレーブデバイスは、2 ビットから成る状態レジスタを含む。状態レジスタの値は、マスタデバイス 5 2 が所与のスレーブデバイスに講じるべき措置を決定するために使用される。各スレーブデバイスは、X および Y ワードで各スレーブデバイスに対する専用タイムスロットに書き込むことによって、状態情報で応答する。本実施例では、状態ビットの総数は 2 4 であり、スレーブデバイスは、最初に最上位状態ビットを伝送する。状態ビットは、割り込みマスクビットの値に応じて、マスク割り込みを生成し得る。電源投入リセットイベント後に、全てのスレーブデバイスは、それらがバス 6 4 にちょうどアタッチされたことを信号伝達するように、スレーブアドレスゼロで応答し始めるであろう。他の実施形態では、スレーブデバイスの状態レジスタは、使用される、より多くの状態レベルがある場合に、2 つより多くのビットを有し得る。

【 0 1 3 7 】

この例示的实施形態では、2 ビットを使用して、スレーブデバイス 5 4 は、4 つの異なる状態レベルを有することができる。「0 0」の状態レベルは、最低状態レベルであり、スレーブデバイス 5 4 が最初にバス 6 4 にアタッチされるが、バス 6 4 にロックされない (すなわち、マスタデバイス 5 2 に同期化されない) ときのスレーブデバイス 5 4 の内側の初期値である。論理 1 のみが、(この例示的实施形態で使用される信号伝達スキームにより) バス 6 4 の物理的動作に影響を及ぼすため、この状態値は、バス 6 4 を不変のままにするであろう。「0 1」の状態レベルは、スレーブデバイス 5 4 がバス 6 4 にロックしていることを信号伝達し、スレーブデバイス 5 4 がバス 6 4 に接続するとき、およびそこから接続解除するときを検出するために使用されることができる。これは、スレーブ状態レベルを読み取り、A T T A C H M E N T ビットを使用して割り込みを有効にすることによって監視され得る。この値は、P I N G 動作に影響を及ぼす。「1 0」の状態レベルは、バス 6 4 にロックした後、スレーブデバイス 5 4 への注意を要求するため (例えば、バッファレベルまたはエラー条件に起因して) に使用される。この条件は、A T T E N T I O N 割り込みマスクビットを有効にすることによって監視され得る。この状態レベルは、S 0 および S 0 D E L A Y I R Q マスクビットが同時にアクティブ (すなわち、「1」) である場合に、R E A D、W R I T E、または F U N C T I O N 動作を遅延させるであろう。「1 1」の状態レベルは、スレーブデバイス 5 4 が、限定されないが、スレーブデバイス 5 4 の温度が高すぎることを、またはスレーブデバイス 5 4 に危機的な電圧不足または過電圧があることを等の迅速に対処される必要があり得る緊急事態を有することを信号伝達する。この条件は、A T T E N T I O N I R Q マスクビットを有効にすることによって監視され得る。この状態レベルもまた、S 0 ビットおよび S 0 D E L A Y I R Q マスクビットが同時にアクティブである (すなわち、論理レベル「1」に設定される) 場合に、R E A D、W R I T E、または F U N C T I O N 動作を遅延させるであろう。

【 0 1 3 8 】

10

20

30

40

50

ここで図10を参照すると、種々のコマンド動作のためのYコマンドワードに対する種々のフィールドおよびビット割付が示されている。Yワードの構造もまた、どのような動作が現在進行中であるかに依存するであろう。一般に、Yワードは、マスタデバイス52と1つ以上のスレーブデバイス54との間で、データおよびおそらくアドレス情報を通信するために使用される。図10は、いくつかの例示的の組み合わせを提供する。この例示的の実施形態では、Yコマンドワードは、ADDRESSフィールドと、WRITE DATAフィールドと、READ DATAフィールドと、FUNCTIONフィールドとを備える。フィールドのうちのいくつかは、説明されるように、あるコマンド動作中にのみ使用される。これらのフィールドは、レジスタの1つ以上のビットを使用して実装されることができる。代替実施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。FUNCTIONフィールドについては、FUNCTION動作が議論されるときに説明する。

10

【0139】

ADDRESSフィールド(ADDR7-0)は、スレーブデバイス54に対する16ビットアドレスを形成するために、Xワードでアドレスフィールドと一緒に使用され得る、8ビットアドレスである。

【0140】

WRITE DATAフィールド(WD15:0)は、スレーブデバイス54中のレジスタに書き込まれるべき、マスタデバイス52からのデータを含む。16ビットWRITE動作については、ワードWD15:0全体が使用される。8ビットWRITE動作については、WD7:0のみが使用される。レジスタWD15:0は、マスタデバイス52の外部制御によって設定されることができる。

20

【0141】

READ DATAフィールド(RD15:0)は、スレーブデバイス54からマスタデバイス52への読み取られるべきデータを含む。16ビットREAD動作については、ワードRD15:0全体が使用される。8ビットREAD動作については、RD7:0のみが使用される。レジスタRD15:0は、マスタデバイス52の外部制御によって読み取られることができる。

30

【0142】

ここで図11aを参照すると、Xワードでの関数および対応するビット設定の例示的リストが示されている。他の実施形態では、他の関数および/またはビット設定を使用することができる。例えば、図11aに記載される関数の異なる部分的組み合わせを使用する、他の実施形態があり得る。これらの実施形態は、他の関数を追加し、ある関数を除去し、またはいくつかの関数の追加および他の関数の除去の両方を行うことができる。FUNCTION動作は、システム機能性の制御およびデバイス制御のために有用である、特別なコマンドの実行を開始するために使用され得る。いくつかの関数は、ブロードキャストをサポートし、すなわち、全てのポート(例えば、スレーブデバイス)が、このコマンドによる影響を受ける。XワードでのFUNCTIONフィールド(X7:X4)の上位ビットは、図11aで定義される全ての関数について0000に等しい。関数が実装されていない場合、スレーブデバイス54は、マスタデバイス52からの対応する要求を確認(ACK)するべきではない。マスタデバイス52が複数のスレーブデバイスに書き込んでいる場合、スレーブデバイスは、この例示的実施形態では、代わりに否定応答(NACK)を使用することができる。

40

【0143】

REQUEST ID関数(X3:X0=0000)の実行は、システムで使用されるマスタデバイス52を一意的に識別するため、または、マスタデバイス52がこれらのスレーブデバイスを区別し、それらの各々に固有の個々のスレーブ番号(例えば、1-14)を割り当てることができるようにバス64にアタッチされた複数のスレーブデバイスを区別するために使用される。これは、同時にバス64を介してマスタデバイス52に接続

50

されるいくつかのスレーブデバイスに対するサポートを可能にし、アドレス衝突問題（典型的にはI²Cバスプロトコルで見られる）を回避する。各スレーブデバイスは、製造業者部品IDを有する。この名称は、製造業者コードおよび部品番号から成るであろう。少なくとも1つの実施形態では、製造業者コードは、個々のデバイスアドレスをバス64に接続される各スレーブデバイスに割り当てるために、アービトレーション中に使用される。同一の部品番号を有するスレーブデバイスの状況は、以下で説明されるように、位置情報を追加することによって対処することができる。

【0144】

リセットイベントがバス64上で起こった後、全てのスレーブデバイスは、スレーブアドレスゼロ（すなわち、レジスタアドレス0x0000）で設定情報を探ることから始めるであろう。マスタデバイス52は、次いで、本説明において以降で説明される動的アドレス割当プロセスで説明されるように、アドレスゼロに書き込み、このアドレスから読み取ることによって、アドレスアービトレーションプロセスを開始するであろう。単一のマスタデバイス52のみがあるので、位置フィールドは、マスタデバイス52から読み取るときに「0000」に等しいであろう。

【0145】

電源投入リセットイベント後、全てのスレーブデバイスは、それらのデバイスアドレスをゼロに設定し、マスタデバイス52は、スレーブデバイスと通信してアドレスをそれらに割り当てる。スレーブデバイス54は、バス64にロックした後、PING動作中に「01」以上の状態を示し、マスタデバイス52は、新しいデバイスがバス64にちょうどアタッチされたことを発見するであろう。マスタデバイス52は、REQUEST ID関数をアクティブにすることによって、アービトレーションプロセスを開始するであろう。スレーブデバイス54は、この要求を受信すると、スレーブデバイス54に関連付けられた内部識別（すなわち、固有の名称）を符号化するために使用される内部名称文字カウンタを有効にする。この瞬間にバス64にロックされていないスレーブデバイスは、起動されず、このアービトレーションサイクル中に応答しないであろう。さらに、いかなるスレーブデバイスもバス64に現在接続されていない場合、マスタデバイス52は、フレームを完成させ続け、次いで、次のフレームサイクルでさらなるPING動作を発行する。

【0146】

後に、マスタデバイス52は、スレーブデバイスゼロからのレジスタアドレスゼロから読み取るであろう。これは、特定の実装に応じて、16ビットまたは8ビットREAD動作を使用して行われることができる。返される値は、スレーブデバイスアドレスゼロを伴ってバス64にアタッチされた任意のスレーブデバイスの最高値部品ID（すなわち、固有の名称）の部品番号であろう。マスタデバイス52は、固有のアドレスが割り当てられそうである全てのスレーブデバイスの固有の名称の第1のビットを読み取る。次いで、各スレーブデバイスの固有の名称の第1のビットが再検討される。それらが全て同一である、すなわち、全て「1」または全て「0」である場合、マスタデバイス52は、スレーブデバイス名の次のビットに進むか、またはそれをポーリングする。「0」および「1」の組み合わせがある場合には、マスタデバイス52は、バス64の動的性質によりバス値を決定するであろうため、「1」を有する固有の名称のみを伴って継続する。換言すると、バス64にアタッチされた1つより多くのスレーブデバイスがある場合、論理0を書き込もうとするが、論理1をリードバックする、任意のスレーブデバイスは、バックオフし、次のアービトレーションサイクルが始まることを待つであろう。マスタデバイス52は、部品IDが完全であることをスレーブデバイス54が示すまで、アドレスゼロから読み取り続ける。これは、ゼロバイト、すなわち、0x00によって示されることができる。次いで、マスタデバイス52は、データビット3:0を設定することにより、所望のデバイス番号をレジスタアドレスゼロに書き込むことによって、個々のデバイスアドレスをこのスレーブデバイスに割り当てる。マスタデバイス52は、現在のデバイスアドレスに書き込み、レジスタアドレスゼロでビット3:0を所望の値に設定することによって、スレーブデバイスのデバイスアドレスを以降で変更することができる。IDを符号化するための

10

20

30

40

50

みに4バイトを使用する、短い符号化形式を使用することができる。スレーブデバイスが、読み取りシーケンスを完了する前に、アドレスゼロからの読み取り以外のI/O要求を受信する場合、バックオフしてプロセスを終了させ得る。

【0147】

動的アドレス割当プロセスは、「アルファベット順」で、マスタデバイス52がデバイスの固有の名称を再検討することを可能にし、より高い数値の名称値を伴うスレーブデバイスは、より低い数値の名称値を伴うスレーブデバイスの前または後のスレーブアドレスを割り当てられる。スレーブデバイス名の数値は、2進数として読み取られている値を用いた検索に基づくソーティングから決定される。単一の「1」値のみがある場合には、マスタデバイス52は、この固有の名称に関連付けられるスレーブデバイスを第1のデバイスとして割り当て、それに固有のスレーブ値1-14を割り当てる。少なくともいくつかの実施形態では、デバイス値1-11のみが、全てのアタッチされたデバイスの連続監視を可能にするために使用されるが、デバイス値12-14は、デバイスグループ化のために保留される。次いで、それは、続けてバス通信に戻り、アドレスを必要とする次のスレーブデバイスに遭遇し、第1のスレーブデバイスとは異なるスレーブデバイスアドレスを割り当ててであろう。このアドレス割当は、バス64にアタッチされた、割り当てられていないスレーブデバイスがなくなるまで継続する。各スレーブデバイスが固有の名称を有するので、マスタデバイス52が、どのスレーブデバイスを現在のデバイスアドレスに関連付けるべきかを把握していないいかなる状況も生じないであろう。スレーブデバイスへの後続のREAD、WRITE、およびFUNCTION動作では、割り当てられたスレーブアドレス1-14は、スレーブデバイスを識別するために使用されるであろう。

【0148】

2つ以上のスレーブデバイスが同一のチップを含む場合、これらのデバイスは、異なるPOSITION情報でプログラムされるか、または、位置に対して符号化すべき追加の端子を有し得る。単純な実施例として、単一の入力端子が、左右のチャンネルの間で情報を提供することができる。1つの代わりに2つの端子を追加することによって、これらの端子の各々をGND、VDD、またはFLOAT(すなわち、接続されていない、または3状態、Z)に接続することにより、最大9つの異なる位置を割り当てることが可能である(図11b-11e参照)。両方の端子は、最初に、入力端子と見なされる。図11dでは、出力端子が浮動している(すなわち、オープン出力端子)場合には、1が入力端子に基づいてアドレスを送信することができる。出力浮動条件は、図11eに示されるような回路によって決定されることができる。典型的には、値Xが内部状態マシンによって駆動されるであろう一方で、TESTは外部端子であり、Yは試験結果である。このI/O回路は、9つ全ての組み合わせを取得するために、SEL_INおよびSEL_OUT端子の両方に使用され得る。他の実施形態では、FLOAT(Z)条件に対する試験が除去され得、リング構造が依然として利用され得る。高値プルダウン抵抗器が、入力端子TESTへの接続を伴わずに電力消費を削減するように含まれ得るが、電力消費が問題ではない場合は随意的であり得る。入力Xに続くインバータからの出力が、典型的には、アクティブである場合には、代替として、2k抵抗器が値を増加させられ、100k抵抗器が省略され得る。デジタルマイクロホン用のレガシーシステムとも適合性がある、例示的ピン構成が図11dに示されている。図11bおよび11cに示されるような例示的符号化は、いかなるプログラミングも伴わず、かつ物理的レイアウトの変更を伴わずに、少なくとも9つの位置、および現在のレガシーシステムとの適合性を可能にする。図11cに示されるように符号化されるリングトポロジーは、第1のデバイスを固定入力値に接続し、全ての後続のデバイスは、入力端子SEL_INを前のデバイスのSEL_OUT端子に接続するであろう。起動中、デバイスは、第1の値(0)から、N-1と番号付けされるであろう最後のデバイス(すなわち、リングトポロジーにおいてN個のデバイスが接続されている)まで増分することによって、それら自体を自動構成するであろう。このトポロジーは、9つより多くの位置を取得することができるという利点を有するが、単一のデバイスが故障した場合に、列挙も失敗するであろうという不利点を有する。したがって、そ

10

20

30

40

50

れは、本実施例では0 - 8の固定アドレスを使用する（他の実施形態では他の範囲を使用することができる）図11bの構成と比較して、よりエラーの傾向があり、あまりロバストではない。両方の場合において、デバイスは、最初に、内部位置の列挙が完了した後に、バス64にアタッチされるべきである。各デバイスが、典型的には、前のデバイスから情報を受信するか、またはそれがリング内の第1のデバイスであることを確認するまで待機するため、このプロセスは、リングトポロジーについて、より長くかかるであろう。

【0149】

実施形態では、スレーブデバイス54がハードウェアエラー状態になり、リセットされる場合、そのスレーブアドレスは、ゼロのデフォルト値に設定される。スレーブデバイス54がマスタデバイス52との同期化を失う場合、その出力を無効にし、マスタデバイス52との同期化を再獲得しようとする。スレーブデバイス54は、その現在のデバイスアドレスを保ち、最後に取得されたコマンド分離値（本書において以降で説明される）を使用して検索し始めるべきである。同期化を再獲得した後、スレーブデバイス54は、そのデバイス状態を「10」に設定することによって、バスエラーを信号伝達し得る。次いで、マスタデバイス52は、エラータイプについてスレーブデバイス54に尋ね、それは、デバイスアドレスゼロ、レジスタアドレスゼロにおいて、SLAVE_SYNC_ERRORまたは「0xFF」で応答し得る。スレーブデバイス54が、読み取りシーケンスを完了する前に、アドレスゼロからREAD動作以外のIO要求を受信する場合、バックオフしてプロセスを終了させるべきである。実施形態では、スレーブデバイス54が動作中に同期化を失い、再同期化することができない場合には、マスタデバイス52は、PING動作の使用によって2つのフレーム内でこのイベントについて知り、次いで、マスタデバイス52の内側でSTATUSレジスタの内側の値を変更し、ATTACHMENTビットを設定するであろう。

【0150】

要約すると、スレーブデバイス54は、以下の状況下で、すなわち、電源投入リセット後、スレーブデバイス54がバス64にロックされていないとき、スレーブデバイス54がマスタデバイス52との同期化を失ったとき、別のスレーブデバイスがアービトレーションを獲得したとき、およびスレーブデバイス54が各自の名称IDの出力を完了したときに、スレーブアドレスの出力を非アクティブにする。

【0151】

スレーブデバイス54は、各自のレジスタをリセットし、マスタデバイス52がFUNCTION_REQUEST_IDを開始し、マスタデバイス52によって書き込まれたデバイスアドレスがスレーブデバイス54に等しいとき、そのIDの読み取りを準備する。別のスレーブデバイスが、このイベントが起こった後にバス64にロックする場合、REQUEST_IDコマンドを受信していないので、たとえそのデバイスアドレスがマスタデバイスアドレスに合致したとしても、アービトレーションを開始するべきではない。したがって、電源投入リセット後、またはスレーブデバイスが自らをバス64にアタッチした直後、スレーブデバイスは、マスタデバイス52からこのコマンドを受信していない限り、アービトレーションに参加することを可能にされるべきではない。

【0152】

HARD RESET関数（X3：X0 = 0001）の実行は、スレーブデバイス全体をリセットするために使用され得、これは、電源投入リセット動作に類似する。

【0153】

SOFT RESET関数（X3：X0 = 0010）の実行は、以前のプログラムされたデバイスアドレスおよび同期化エンジンを除いて、全ての機能性をリセットするために使用され得る。一般に、スレーブデバイス用の内部レジスタおよび機能性は、スレーブデバイスがすでにバス64にロックされており、すでにアドレスを割り当てられていることを除いて、電源投入リセットイベント後と同一であろう。これは、スレーブデバイスがバス64にすでにアタッチされ、ゼロとは異なるアドレスがすでに割り当てられていることを除いて、スレーブデバイスは、電源投入リセットイベント後、そうしていたであろうよ

10

20

30

40

50

うに挙動するであろうことを意味する。これは、スレーブデバイスが再度プログラムされることが出来る前に、スレーブデバイスの内側の全てのレジスタをリセットし、スレーブデバイスをバス64にロックさせ、次いで、アドレスを割り当てる場合よりも速く、内部または外部エラーを補正するために使用することができる。

【0154】

REQUEST CAPABILITY INFORMATION関数(X3:X0=0101)の実行は、マスタデバイス52またはスレーブデバイス54の能力についての情報を得るために使用され得る。これは、ポートを自動的に設定するために使用されることが出来る。この関数は、REQUEST ID関数のように、レジスタ0およびレジスタ1から値を読み取ることによって機能する。返される情報は、アドレス指定されているポートに対応する。ポートに接続されたチャンネルからの全ての情報が読み取られたとき、代わりに、情報の終了を示すゼロが返されることが出来る。この例示的实施形態では、ポートが連続的に定義される。

10

【0155】

マスタデバイス52またはスレーブデバイス54の能力情報を取得するレジスタ0x0000の実施例が、図12に挙げられている。レジスタ0x0000から読み取ると、スレーブデバイスの内側のポートの能力が与えられるであろう。ポートの能力を読み取ると、ゼロ終端文字列が返されるであろう。この文字列は、このポートについての情報を含むであろう。ポートの能力は、それがどの方向にあるか(すなわち、入力ポートまたは出力ポート)、それがいくつのチャンネルを有するか、およびこれらのチャンネルに対するデータ形式がどのようなものであるかを伝えるであろう。全てのポートが読み取られた後、次の読み取りは、値0x00を返すであろう。DATAレジスタ0x0000(ならびに0x0001)は、主に、データ(場合によっては大量のデータであり得る)を転送するために使用され、例えば、文字列、ADDRESS ID、較正データ等を読み取るために使用される。アドレス0x0000が、DATAレジスタの下位バイトである一方で、アドレス0x0001は、上位バイトである。別のレジスタ0x0C-0x0Dは、アドレスを転送するために使用されることができ、ADDRESSレジスタと呼ばれる。アドレスは、マスタデバイス52からの要求に回答して転送される。

20

【0156】

DATAレジスタ0x0000は、概して、TYPEフィールドと、CHANNEL INFOフィールドと、DATA FORMATフィールドとを含む。これらのフィールドは、レジスタの1つ以上のビットを使用して実装されることが出来る。代替実施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。

30

【0157】

TYPEフィールド(B7)は、データ転送の方向を示すために使用され得、例えば、入力動作(すなわち、バス64からデータを読み取る)を示すために0を使用することができ、出力動作(すなわち、データをバス64に出力するか、または書き込む)を示すために1を使用されることが出来る。入力および出力の両方をサポートするチャンネルを2つの別個のポートに接続することができ、この例示的实施形態では、これらのポートのうちの一方は、入力ポートとして定義され、これらのポートのうちの他方は、出力ポートとして定義される。少なくともいくつかの実施形態では、1つのポートに接続された全てのチャンネルが、同一のデータ形式および同一のサンプルレートを共有する。

40

【0158】

CHANNEL INFOフィールド(B6:B4)は、このポートに関連付けられるチャンネルの数を定義するために使用され得る。各ポートは、1つ以上のチャンネルに関連付けられ得る。CHANNEL INFOフィールドは、PORTフィールド値によって定義されるポート数とともに、所与のスレーブデバイスからのチャンネルの総数を返すであろう。実施形態では、PORTフィールド値は、2進マイナス1で符号化されることができ

50

、その場合、「000」は、1つのチャンネルまたはチャンネル1を意味し、「101」は、6つのチャンネルを意味する等である。CHANNEL INFOフィールドに書き込むことによって、能力に対する次のREAD動作が、GENERAL INFOフィールドのそのチャンネルについての情報を生じるであろう。実施形態では、ポートにつき使用されるチャンネルの数は、8であり得、ポート内の各チャンネルに対するサンプルレートは、同一である。複数のサンプルレートが使用される場合、チャンネルは、同一のサンプルレートを有する、より小さいグループにグループ化されることができる。

【0159】

DATA FORMATフィールド(B3:B0)は、データの形式を示すために使用され得る。例えば、バイナリ、ビットストリーム、文字列、および浮動小数点等の種々のデータタイプを使用することができる。種々のデータ形式およびDATA FORMATフィールド中のそれらの対応する表現の実施例が、統一バス通信プロトコルの例示的实施形態について図13に示されている。

【0160】

図11aをもう1度参照すると、READ IRQ SOURCE関数(X3:X0=0011)の実行は、この関数が最後に実行されてからの第1のIRQソースを返すために使用され得る。これは、バスエラーが起こったフレーム内のビット位置についての情報、センサからの状態情報等を返すことができる。IRQイベントが起こった場合、エラーコードが、図14に示されるようにDATAレジスタ0x0000および0x0001において報告される。図14は、統一バス通信プロトコルの実施形態に対する関数READ LAST ERRORの実行後にDATAレジスタから読み取られた状態の実施例を示す。IRQイベントが起こらない限り、いかなるエラーコードも報告されないであろう。統一バス通信プロトコルの少なくとも1つの実施形態で使用されることができる、種々のエラーコードの実施例が図15に示されている。他の実施形態では、より多いまたは少ないエラーコードを使用することができる。

【0161】

図11aを再度参照すると、REQUEST RELATIVE CALIBRATION DATA関数(X3:X0=0110)の実行は、構成要素の製造差異を減少させるために使用され得る。校正係数が、構成要素の典型的な性能に関して測定される。校正データは、ビットストリーム用の16ビットデジタルワードを除いて、ポートによって特定されたものと同じデータ形式を使用して、データフィールド中で返される。実施形態では、デジタルワードは、-4.0に等しいMSBを伴う2補数表現で符号付き分数(fracti on)として符号化され得る(例えば、0x8000は-4.0000であり、0x7FFFは3.99988である)。デジタルワードが、現在の選択されているポートに接続される全チャンネルについて返されるであろう。返される第1のワードは、フルスケールと比較した測定の相対精度を示すであろう。実施例として、返された値が2.00である場合には、ポートからの値は、この構成要素についてデータシートで記述されるような典型的な値であるものより2倍高い。値は、係数が16符号付きデジタルワードとして記憶され得る、ビットストリームを除いて、ポートデータと同一のデータ形式で記憶される。いくつかの実施形態では、デジタルワードは、0から25%の間の値を伴う符号なし整数、すなわち、測定精度がフルスケールの約3.125%であることを意味する、「0001000000000000」として記憶され得る。

【0162】

REQUEST ABSOLUTE CALIBRATION DATA関数(X3:X0=0111)の実行は、SI単位での出力量による絶対校正情報を取得するために使用され得る。これは、補間を用いて達成される。例えば、線形、多項式、スプライン、およびクレンショウ等の種々の補間技法を使用することができる。この関数は、測定された物理的パラメータのオフセットおよびスケリングについての校正情報を返すために使用されることができる。返される第1のワードは、REQUEST ABSOLUTE CALIBRATION DATA関数について上記で説明されるように、フルスケールと

10

20

30

40

50

比較した測定の絶対精度を示すであろう。いくつかの実施形態では、絶対および相対較正データ関数は、単一のコマンドに組み込まれ得、アドレス指定されたスレーブデバイスは、これらの値のいずれか（すなわち、相対または絶対較正データ）で応答するように要求されるであろうか、またはどのタイプの較正情報が利用可能であるかをそれ自体がマスターデバイスに伝え得る。

【0163】

READ AND INCREMENT ADDRESS 関数 (X3 : X0 = 1000) の実行は、スレーブデバイス54の内側のレジスタADDRESSによって特定されるような16ビットデータを読み取り、このレジスタのコンテンツをYワードに転送するために使用され得る。その後、スレーブデバイス54の内側のADDRESSレジスタの値は、1だけ増分される。

10

【0164】

WRITE AND INCREMENT ADDRESS 関数 (X3 : X0 = 1001) の実行は、Yワードで16ビットデータを取り込み、このデータをスレーブデバイス54の内側のレジスタADDRESSによって特定されるアドレスに書き込むために使用され得る。その後、スレーブデバイス54の内側のADDRESSレジスタの値は、1だけ増分される。

【0165】

EXECUTE FUNCTION AT ADDRESS 関数 (X3 : X0 = 1010) の実行は、スレーブデバイス54がプロセッサを備える場合に使用され得、その場合、このプロセッサは、次のフレームの開始時に、Yワードによって定義される16ビットアドレスを用いて、サブルーチンへのジャンプ動作を行うであろう。

20

【0166】

EXECUTE FUNCTION AT ADDRESS INDIRECT 関数 (X3 : X0 = 1011) の実行は、スレーブデバイス54がプロセッサを備える場合に使用され得、その場合、プロセッサは、次のフレームの開始時に、Yワードによって定義されるメモリエセルの内側に含まれるアドレスを用いて、サブルーチンへのジャンプを行うであろう。

【0167】

PERFORM DEVICE SELF TEST 関数 (X3 : X0 = 0100) の実行は、例えば、デバッグおよび製造試験のために行われ得る、デバイス自己試験を行うために使用され得る。自己試験の結果は、エラーコードをもたらすか、またはいかなるエラーコードももたらさないであろう。エラーコードについては、READ LAST ERROR 関数の説明を参照されたい。低レベル割り込みが、自己試験の終了時にアクティブにされるであろう。エラーコード「デバイス自己試験」は、デバイス試験が完了したときに報告されるであろう。

30

【0168】

SET GAIN 関数 (X3 : X0 = 1100) の実行は、チャンネルの利得を所望の値に設定するために使用され得る。

【0169】

SET DELAY 関数 (X3 : X0 = 1101) の実行は、ポートまたは個々のチャンネルに対して、サンプルの数の遅延を所望の値に設定するために使用され得る。個々のチャンネルが選択される場合、チャンネル数が特定される必要があるであろう。これは、CHANNEL SELECTION レジスタを使用して行われることができる。いくつかの実施形態では、遅延は、より細かい遅延精度のために、わずかな数のサンプルで特定され得る。

40

【0170】

SET POWER 関数 (X3 : X0 = 1110) の実行は、ポート、またはビットB2 : B0を伴うデバイスの電力消費を設定するために使用され得る。これの実施例が、ビットB2 : B0を使用して、図17に示されている。別の実施形態では、異なる数のビットまたは異なる符号化が使用され得る。

50

【0171】

マスタデバイス52およびスレーブデバイス54は、バス64を経由したデバイスの転送をサポートすることができるように通信チャンネルを設定する。マスタデバイス52への入力は、処理された無線データから、プロセッサから、またはTDMあるいはI²Sベースのチャンネル等の何らかの他のデジタル通信リンクを通してもたらされ得る。マスタデバイス52またはスレーブデバイス54によって使用される各ポートには、バス64上で伝送されるデータ中の正しいデータスロット（すなわち、タイムスロット）にトラフィックをダイレクトするレジスタが関連付けられる。例えば、マスタデバイス52およびスレーブデバイス54は、コマンドによってポートアクティビティを設定する、8つのレジスタを含むことができる（別の数のレジスタを代替実施形態で使用されることができ）。複数のチャンネルが、データポートにおいて一緒に割り付けまたはグループ化され、プログラムされることができ、次いで、選択されたチャンネルがアクティブにされることができ。これは、スレーブデバイス54がバス64にロックした（すなわち、同期化された）後にマスタデバイス52によって為されることができ。これらのデータチャンネルから、またはそこへのデータは、フレーム中で2回以上生じるように、または例えば、本説明において以降でさらに詳細に説明されるであろう、SKIPおよびREPEAT等の種々のパラメータを使用することによって、フレーム中で異なる間隔を有するようにプログラムされることができ。

10

【0172】

実施形態では、4つのタイプの動作が、これらのレジスタに関連付けられ、動作は、データポート（Data Port）、設定（Setup）、チャンネルのアクティブ化（Activate channel）、および能力（Capability）である。代替実施形態では、他の動作を使用することができる。加えて、マスタデバイス52は、レジスタ0x0C-0x0Dに位置するアドレスポートを有する。スレーブデバイス54は、随意に、例えば、アドレスにジャンプする機能等の特殊機能のサポートのために、アドレスレジスタを含み得る。レジスタは、バイトベースであり、アドレスゼロからの16ビットワードREAD動作は、最下位ビット（LSB）におけるレジスタアドレス0のコンテンツ、および最上位ビット（MSB）におけるレジスタアドレス1のコンテンツを返すであろう。レジスタアドレスは、16ビットモードを使用するとき2で乗算される。

20

【0173】

ポート番号は、概して、デバイス上のハードウェアにおいて固定される。特定のポート番号に関連付けられるオーディオまたはデータストリームは、デバイス特有である。ポート番号の実施例は、本説明のポート定義の節で挙げられる。マスタデバイス52のレジスタマップに（例えば、I²Cを通して）直接アクセスすることができる一方で、スレーブデバイス54のレジスタマップは、マスタデバイス52によってプログラムされることができる。チャンネル情報は、統一バス通信プロトコルを使用して、データスロット（すなわち、タイムスロット）を割り当てるために使用される。チャンネル割当は、変更されることができ、システムおよび使用事例に依存する。チャンネル割当は、デバイスからのどのポートが、バス64上の特定のタイムスロットを割り当てられるかを定義する。実施形態では、マスタデバイス52は、少なくとも2つの入力および2つの出力ポートを有することができ、各々が最大32ビットワードを取り扱うことができる。他の実施形態では、異なる数の入力ポートまたは出力ポート、および異なるサイズのデータをマスタデバイス52に使用することができる。一般に、スレーブデバイスは、ポートに対するいかなる要件も有さない。これらの動作へのレジスタマップは、統一バス通信プロトコルの少なくとも1つの実施形態では、マスタデバイス52またはスレーブデバイス54に使用されることができ、レジスタ定義の実施例である図16aに示されている。レジスタマップの別の実施例が、図16bに示されている。これらのフィールドは、1つ以上のレジスタの1つ以上のビットを使用して実装されることができ、代替実施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。アドレスOFFSE

30

40

50

Tが、マスタデバイス52に対する絶対レジスタアドレスに追加され得る。次いで、OFFSET+ADDRESSを使用して、実際のアドレスを計算することができる。スレーブデバイスについては、OFFSETは、0x00である。これは、マスタデバイス52がどのアドレスを読み取るか、どのアドレスに書き込むかを知るように、スレーブデバイスが主要バスレジスタ(0x00-0x07)に対する絶対アドレスを使用するであろうことを意味する。しかしながら、マスタデバイス52は、すでにそれ自体についての情報を有するので、そのレジスタマップでゼロとは異なるオフセットをそれ自体が使用し得る。しかしながら、レジスタのシーケンスは、ソフトウェアの簡単のために同一であり得る。他の実施形態では、マスタデバイス52およびスレーブデバイスに対するレジスタのシーケンスは、異なり得る。

10

【0174】

DATA PORT関数は、1つまたは2つの8ビットレジスタによって表すことができる。これらのレジスタからREAD動作を行うとき、8または16ビットワードがYワードで返されるであろう。これらのレジスタへのWRITE動作(8または16ビット)を行うとき、値が更新されるであろう。これらのレジスタのデフォルト値は、電源投入リセット動作後に0x0000である。DATAフィールド(reg.0x00、B7:0)は、16ビットエンティティの下位8ビットを表す。一実施形態では、16ビットデータ転送中に、上位アドレスビットは、ゼロであると仮定される(すなわち、16ビットデータは、簡単にするために偶数アドレスにのみ書き込むことができる)。DATAフィールド(reg.0x01、B15:8)は、16ビットデータエンティティの上位8ビットを表す。

20

【0175】

設定ポート関数に関連付けられるレジスタは、マスタデバイス52およびスレーブデバイス54が、統一バス通信プロトコルを使用して、バス64上で伝送されるデータ中の所望のデータスロット(すなわち、タイムスロット)に情報を伝送することができるように、値を割り当てられる。割当の詳細の実施例は、図16aに示される種々のフィールドの説明によって挙げられる。

【0176】

ACTIVATEフィールド(reg.0x03、B7)は、データストリームの機能、およびそれがどのようにして制御されるかを定義するために使用され得る。入力ストリームが非アクティブにされる場合、出力はゼロになるであろう。出力ストリームが非アクティブにされる場合、何もバス64に書き込まれないであろう。

30

【0177】

PREPAREビット(reg.0x03、B6)は、オンにされるであろう、現在の選択されているポートに関連付けられる回路を定義するために使用され得る。このビットから読み取るとき、ポートの現在の状態が返されるであろう(論理1は、ポートがデータ転送を始める準備ができていることを意味する)。これは、全ての回路が準備できており、データ転送が始まる時に故障が起こらないことを確実にするために使用されることができる。

【0178】

40

代替実施形態(例えば、図16b)では、POWER LEVELフィールド(reg.0x02、B6:B4)は、システムの電力消費を最適化するために、個々のデバイス、ポート、およびチャネルの電力消費および動作モードを設定するように使用され得る。システム構成要素は、電力消費を最小化するために使用されることができる、いくつかのモードで動作することができる。例えば、CHANNEL SELECTIONフィールドを使用して、個々のチャネルを選択することができる。選択されていないチャネルは、電力を節約するために、低電力モードで設定され得る。例えば、統一バス通信プロトコルの1つの使用法では、PREPAREビットがアクティブにされるまで、全てのチャネルが低電力動作モードにあり得る。PREPAREビットがアクティブにされた後、CHANNEL SELECTIONフィールドで選択されているチャネルは、通常動作モード

50

にされ得る。PREPAREビットの状態を読み取ることによって、選択されているチャンネルがデータを受信または伝送する準備ができており（限定されないが、例えば、いくらかのデバイス依存性初期電源投入遅延後等に）、ACTIVATEビットを使用することによってアクティブにされることができるときを知ることが可能である（すなわち、全ての選択されたチャンネルが準備できているときにPREPAREビットは論理1レベルになるであろう）。

【0179】

少なくとも1つの実施形態では、電力がデバイスに印加されないとき、バス64は、依然として動作を継続することが可能であり得、すなわち、ESD構造は、電力がスレーブデバイス54またはマスタデバイス52に印加されないときに、バス64を停滞させるべきではない。

10

【0180】

ここで図17を参照すると、例示的实施形態での種々の電力消費レベルの実施例が示されている。本実施例では、電力消費スケリングは、関数によって定義され、絶対的ではない。換言すると、図17に示される正確な電力レベルは、異なるスレーブデバイスについて変化し得る。さらに、 $B6 : B4 = 011$ から 111 によって定義される電力消費レベルについては、ポートまたはデバイスは、完全に機能的であり得るか、またはそうではない場合がある。デバイス全体の電力消費が設定される場合、ブロードキャストのためのポート(15)が使用され得る。ポートまたはデバイスは、いくつかの電力レベルのみをサポートし得る。例示的实施形態では、マスタデバイス52がスレーブデバイス54によってサポートされていないPOWER LEVELの値を書き込む場合、それは、サポートされる最も近い値にマップされることができ、POWER LEVELフィールドのデフォルト値は、010（すなわち、SLEEPモード）である。

20

【0181】

図16aを再度参照すると、IRQ23 MASKフィールド(reg. 0x03、B5)は、通常はS0ビットをアクティブにするであろうスレーブ状態レベル「10」および「11」として、バス64を経由して通信される、スレーブデバイスからの割り込みを有効または無効にするために使用される。IRQ23 MASKフィールドが1に等しいとき、スレーブデバイス54は、スレーブ状態レベルがレベル2またはレベル3であることを内部論理が示すときに、高レベル割り込みを発行するであろう（すなわち、スレーブデバイス54が割り込みを要求しており、かつバス64にこの割り込みを承認することを可能にされる）。このビットがゼロに等しいとき、何も起こらず、示されたスレーブ状態レベルは、S0ビットに影響を及ぼさないであろう。IRQ23 MASKフィールドのデフォルト値は、1である（すなわち、通常は割り込みが有効にされる）。

30

【0182】

IRQF MASKフィールド(reg. 0x03、B4)は、スレーブマスタデバイス52がこのスレーブデバイスに行うように要求する所与のFUNCTIONをデバイスがサポートしない場合に生成される、割り込みを有効にするために使用される。IRQF MASKフィールドが1に等しいとき、マスタデバイス52が、それがサポートしないFUNCTION動作を実行するようにスレーブデバイス54に求めるとき、スレーブデバイス54は、低レベル割り込みに設定されるであろう。これは、自動的にS0割り込みビットを設定するであろう。スレーブデバイス54がすでに、割り込みが保留中である状況にある場合、新しいことは何も起こらないであろう。IRQF MASKフィールドがゼロに等しいときは、何も起こらない。IRQF MASKフィールドのデフォルト値は、1である。

40

【0183】

PORTフィールド(reg. 0x03、B3 : B0)は、デバイス内のどのポートがアドレス指定されているかを示す。この例示的实施形態では、ポート番号は、番号15がブロードキャストのために保留される（すなわち、全てのポートがマスタデバイス52からのコマンドをリッスンする）、0 - 15であり得る。ポートは、ゼロから最大8

50

つのチャンネルのいずれかと接続されることができ、このマッピングは、配線接続されている。代替実施形態では、ポートは、8つより多くのチャンネルと接続されることができる。全てのチャンネルは、それらが同一のデータ形式を有するように特定のポートに接続される。スレーブデバイス54またはマスタデバイス52は、複数のポートを有することができる。ブロードキャストコマンド中、全てのポートは、例えば、ACTIVEおよびBANK等のある関数への変更をリッスンするであろう。ポートを使用するために、アクティブにされるであろうチャンネルが、最初に選択されるべきである。次いで、データが準備できる前にデバイスがいくらかの時間を要する場合に、PREPAREビットがアクティブにされるべきであり、適切なフレームおよびデータ形式が選択されるべきであり、最終的に、ACTIVEフィールドを真に設定することによって、ポートがアクティブに

10

【0184】

ここで図18を参照すると、統一バス通信プロトコルを使用して2つのスレーブデバイス54aおよび54bと通信する、マスタデバイス52'の例示的实施形態が示されており、スレーブデバイス54aおよび54bは、種々のチャンネルに割り付けられるポートを有する。種々のオーディオおよびデータチャンネルに対するポート番号が、各論理ブロック内に示されている。これらのポート番号、チャンネル、およびインターフェースは、一例のみとして提供され、ポート番号、チャンネル、およびインターフェースの種々の組み合わせを伴う種々の他の実施形態があり得る。

【0185】

マスタデバイス52'は、多重化および同期エンジン100と、ポート102から110とを含む。多重化および同期エンジン100は、使用されている統一バス通信プロトコルのフレーム形式に従って、データを生成し、バス64上のあるタイムスロット中にデータを配置するために使用される。多重化および同期エンジン100はまた、統一バス通信プロトコルに対する同期化情報を生成するためにも使用される。ワードフレーム形式は、バス64上の通信がワードモード下で行われるときに使用され、ビットストリームフレーム形式は、バス64上の通信がビットストリームモード下で行われるときに使用される。これらの形式の両方を、本明細書において以降でさらに詳細に説明する。本実施例では、伝送されるデータの一部が、ポート104~110から受信される。

20

【0186】

一般に、プロセッサを使用して、多重化および同期エンジンを実装することができる。しかしながら、多重化および同期エンジンのために、およそ2Kゲート総数があるので、状態マシンまたは他の専用回路も実装に使用されることができる。例えば、典型的な0.18 μ mプロセスでは、マスタデバイス用の同期エンジンは、0.04mm²で実装されることができる。完全オーディオサポートを伴う完全なマスタデバイスは、いくぶん大型であり得るが、0.1mm²未満であり得る。スレーブデバイス用の同期エンジンは、両方の場合において同様のサイズであり得る。これは、小さいサイズが、より単純なデバイスドライバ、およびハードウェア実装のより良好な試験を意味するため、有益である。代替実施形態では、携帯電話またはスマートフォン用のベースバンドプロセッサ等のマスタデバイス52'を含む電子デバイス用のメインプロセッサを使用して、マスタデバイス52'を実装することができ、その場合、制御情報は、マスタデバイス52'によって内部で生成され得、ポート104から110は、随意的であり得る。

30

40

【0187】

ポート102は、情報をバス64に接続される全てのデバイスのポートに伝送するために使用される、ブロードキャストポートである。それは、全ての他のポートのレジスタマップに接続され、1度に全てのポートを制御するために使用される。それは通常、オーディオデータ転送には使用されない。ポート104は、1つの方向へI²S形式に従って情報を転送するように、I²Sインターフェースに連結される(例えば、伝送機の役割を果たす)。ポート106は、反対方向へI²S形式に従って情報を転送するように、別のI²Sインターフェースに連結される(例えば、受信機の役割を果たす)。ポート108は

50

、M c B S P T D M形式に従って情報を転送するように、M c B S Pインターフェースに連結される。ポート110は、I²C形式に従って制御情報を受信するように、I²Cインターフェースに連結される。

【0188】

スレーブデバイス54aは、多重化および同期エンジン120と、ポート122から126と、アナログ/デジタル変換器(ADC)128と、デジタル/アナログ変換器(DAC)130とを備える。ADC128およびDAC130は両方とも、本実施例では2つのチャンネルを有する。スレーブデバイス54aの多重化および同期エンジン120は、以下でさらに詳細に説明されるように、マスタデバイス52'に同期化するために使用される。ポート122は、ブロードキャストポートである。本実施例では、ポートアドレス15は、コマンドのブロードキャストに使用され、すなわち、アドレス15に送信される任意のコマンドは、特定のスレーブデバイスに接続される全てのポートに影響を及ぼすであろう。さらに、スレーブアドレス15もまた、ブロードキャストイングのために保留され、すなわち、このアドレスへのWRITE動作は、全てのスレーブデバイスに影響を及ぼすであろう。したがって、スレーブアドレス15およびポート15への書き込みは、システム内の全てのデバイスおよび全てのポートに影響を及ぼすであろう(例えば、同時にデータの動作停止または同時起動のために)。ポート124は、ADC128の2つのチャンネルに接続されるカスタムポート(全データビットに直接ワイヤを使用する並列接続のためのI²CまたはI²S等の非標準インターフェースを備えることを意味する)である。代替実施形態では、2つより多くのチャンネルがあり得、ポート124は、2つより多くのチャンネルに接続され得る。ADC128は、マイクロホン、電流センサ、電圧センサ、圧力センサ、温度センサ等のデータを生成する1つ以上のデバイスに接続される。異なる場所に位置付けられた2つのマイクロホン等の2つの同一のタイプのデバイスがADC128に接続される、実施形態があり得る。ポート126は、DAC128の2つのチャンネルに接続される。代替実施形態では、2つより多くのチャンネルがあり得、ポート126は、2つより多くのチャンネルに接続され得る。DAC126は、限定されないが、例えば、出力スピーカ、デジタルFM受信機、デジタルFM伝送機、Bluetooth(登録商標)インターフェース、振動器、温度センサ、バッテリーの充電状態を監視するセンサ、加速度計、ジャイロスコープ、GPS受信機、またはデジタルコンパス等のデータを受信する1つ以上のデバイスに接続される。

【0189】

スレーブデバイス54bは、多重化および同期エンジン140と、ポート142から152と、ADC154と、DAC156と、I/O158および160とを備える。I/O158および160は、それぞれ、例えば、ステレオADC(2つのチャンネルとして表される)、および4チャンネルDAC(4つのチャンネルとして表される)であり得る。ポート142から148、ADC154およびDAC156は、ポート124およびADC128と比較して、より多くのチャンネルがポート146およびADC15に使用されることを除いて、ポート122から126、ADC128、およびDAC130に類似する。ポート150は、I²Sプロトコルに従って、順にI²Sインターフェースに接続されるI/O158にデータを送信または受信するように構成される。ポート152は、I²Cプロトコルに従って、順にI²Cインターフェースに接続されるI/O160にデータを送信または受信するように構成される。ポート150は、物理的I²Sインターフェースに接続され、左右に2つのオーディオチャンネルを含む。2つのチャンネルは、両方ではなく、入力または出力のいずれかであり得る。ポート150は、マルチフォーマットバス64からデータを捕捉し、データをI²S形式に転送するか、または反対方向に逆も同様に行うであろう。ポート152は、I²Cインターフェースに接続され、I²CマスタデバイスまたはI²Cスレーブデバイスとして機能することができる。適切なタイミングを使用することによって、I²Cコマンドは、マルチフォーマットバス64を経由して通信され、I²Cインターフェースに書き込まれるか、またはI²Cインターフェースから読み取られることができ(スレーブデバイスとして動作されるとき)、I²Cコマンドは、マルチ

10

20

30

40

50

フォーマットバス64からレジスタデータを要求し、それによって、起こっている他のアクティビティの全てと同時に、マルチフォーマットバス64を経由してI²Cアクティビティをトンネリングするために使用することができ。

【0190】

場合によっては、スレーブデバイス54bは、代わりに、フレーミング能力（すなわち、フレーム形式を設定する能力）がない追加のマスタデバイスとして実装され得、それによって、元のマスタデバイス52'からのいかなる介入も伴わずに、他のデバイスのレジスタにアクセスするために、これらのレジスタへの直接アクセスを得る。コマンドの符号化（例えば、PINGはコマンドフィールド内で000として符号化される）により、任意のPINGコマンドは、代替コマンドに変更されることができる（これが起こることを主要マスタデバイス52'が可能にする場合）。

10

【0191】

場合によっては、データ帯域幅が単一のワイヤにわたって十分ではない場合、1つより多くのデータラインがデータを転送するために使用され得るか、または2つの別個のバスがマスタデバイスとスレーブデバイスとの間で使用され得る。これは、いくつかのデバイスが、低帯域幅（例えば、大抵は制御または低速等時性転送に使用される）のみを使用し、本システムの他の部品、例えば、デジタルスピーカまたはマイクロホンがより高いクロック周波数を使用し得る場合に有益である。本システムを低帯域幅部分および高帯域幅部分に分割することによって、電力消費を削減することができ、場合によっては配線をより単純にすることができる。

20

【0192】

もう一度図16aを参照すると、CHANNEL SELECTIONフィールド（reg. 0x02、B7:0）は、所与のポートがアクティブにされるときに、所与のポートからのどのチャンネルが使用に選択されるかを決定するために使用され得る。ポートからのチャンネルの数は、固定され、CHANNEL SELECTIONフィールドは、どのチャンネルが、ワード開始、チャンネル長、およびスキップフィールドを含む種々のパラメータの影響を受けるかを選択するために使用される。アクティブにされるように選択されるチャンネルは、CHANNEL SELECTIONフィールド中で1に設定された対応するビットを有する。チャンネル間のマッピング、およびポートのどのチャンネルが通信に使用されるかを説明するCHANNEL SELECTIONフィールドの実施例が、図19

30

【0193】

例示的实施形態では、CHANNEL SELECTIONフィールドによって定義される全てのチャンネルは、同一のサンプリング周波数を有し、実装を単純化するように同時にサンプリングされるであろう。サンプリングは、これらの値をデジタル信号に変換するときに、アナログドメイン内で行われる。同様に、複数のデジタル・アナログ変換器を含むポートは、フレーム内で同一の遅延を有するべきであり、その結果として、同一の信号が全てのチャンネルに印加される場合、これらのアナログ信号が同一に見える（たとえこの情報がバスを経由して順次に転送されたとしても）。サンプルがフレームにつき1回配置されるとき、サンプリングイベントは、フレームの開始として受け止められる。1つより多くのサンプルが存在する場合、それらは、フレームの開始を第1のサンプルとして等距離で離間され得る。フラクショナルフローを使用するとき等、サンプリングが全フレームにわたって同一ではない場合、複数のフレームにわたってサンプリングを同期化するために、同期化ビットDS0からDS1をサンプリング基準として使用することができる。各ポートは、各自のCHANNEL SELECTIONフィールドを有し、アクティブであるポートは、reg. 0x02でPORTフィールドによって選択される。

40

【0194】

50

もう1度図16aを参照すると、STARTフィールド(reg. 0x04、B7:B0)は、フレーム中の第1のタイムスロットを定義するために使用され得、その時点で、ポートが定義され、時間基準としてコマンドワードを使用して、データを伝送または受信する。実施形態では、この値は、符号なし2進数で半ニブル(半ニブルは2ビットである)の単位で与えられる。例として、STARTフィールドに対する半ニブルでの「00001000」という値は、コマンドワードの終了後にデータが16ビットを開始することを示す。

【0195】

CURRENT BANKフィールド(reg. 0x0F、B4)(図29a参照)は、どのレジスタが次のREAD、WRITE、またはFUNCTION動作に使用されるかを選択するために使用され得る。PING動作を伴う全フレーム中に、アクティブなメモリバンクが出力されるであろう。しかしながら、少なくとも1つの実施形態では、バス64にアタッチされたデバイスは、最初に、単一ビットエラーから守るために、2つのフレームの遅延後にそのレジスタバンクを変更するであろう。これは、出力または入力故障を伴わずに、2つの動作モードの間でシームレスな変更を可能にする。CURRENT BANKビットは、アクティブなレジスタバンクを選択するためにマスタデバイス52によって使用される。

10

【0196】

従来、これらの異なるデータモードの間の切り替えは、量を減少させ、レジスタの設定を変更し、次いで、量を増加させることによって行われることに留意されたい。有利には、統一バス通信プロトコルの側面は、レジスタを切り替えることができ、アクティブなレジスタがPING動作で示されているため、データを即時に切り替える能力であり、例えば、オーディオ出力においてクリックを伴わずにオーディオシナリオを変更することが可能である。

20

【0197】

図16bの代替実施形態では、BANK MODEフィールド(reg. 0x05、B7)は、BANK SELECTビットの解釈を決定するために使用され得る。BANK MODEフィールドがゼロに等しいとき、BANK SELECTビットは、I/O動作中、最上位アドレスビットに取って代わる。さらに、(I/O動作中にマスタデバイス52によって与えられるような)I/Oアドレスの最上位ビットは、現在のバンクを選択するために使用される。BANK MODEフィールドが1に等しいとき、BANK SELECTビットは、どのバンクに次のI/O動作が影響を及ぼすかを選択するであろう。種々の可能性が図20に示され、図20は、例示的实施形態におけるBANK MODEフィールドおよびBANK SELECTフィールドに応じて、READまたはWRITE動作中に選択されるバンクおよびアドレスの例を示す。

30

【0198】

図16bの代替実施形態を再度参照すると、BANK SELECTフィールド(reg. 0x05、B6)は、どのレジスタが次のREADまたはWRITE動作に使用されるかを選択するために使用され得る。PING動作を伴う全フレーム中に、アクティブなメモリバンクが出力されるであろう。しかしながら、実施形態では、バス64にアタッチされたデバイスは、最初に、例えば、単一ビットエラーを防止するために役立つため等、ある数のフレームの遅延後に、BANK MODEフィールドを変更するであろう。この場合、CURRENT BANKビットは、アクティブなレジスタバンクを選択するためにマスタデバイス52によって使用される。BANK MODEフィールドは、図20に示されるようなBANK SELECTフィールドの解釈を決定する。

40

【0199】

図16aをもう1度参照すると、LENGTHフィールド(reg. 0x06、B5:B0)は、半ニブルの数(すなわち、2ビットの単位)で、各チャンネルのデータサイズを定義するために使用され得る。例示的实施形態では、LENGTHフィールド中の値は、2進マイナス1で符号化され、2ビットの単位で求められる(例えば、“01011” =

50

0 x 0 Bは、 $11 + 1 = 12 \times 2$ ビットまたは24ビットデータ幅として解釈される)。本実施例では128ビットである、最大チャンネル幅も定義される。最大チャンネル幅以上が量(例えば、IEEEクアド精度複素数)を表すために使用される場合、これは、2つのチャンネルをポートにマージすることによって達成することができる。SUBGROUPフィールドを使用して、マージされたチャンネルの間に間隔を追加することが依然として可能である。LENGTHフィールドは、チャンネルサイズがビットストリームデータに対して1ビットであるため、ビットストリームデータを転送するポートについては無視される。ビットストリームモードでは、チャンネルの長さは1である。

【0200】

SKIPフィールド(reg. 0x06、B7:B6)は、REPEATフィールドがアクティブであるときにスキップされる、タイムスロットまたはデータスロットの数を定義するために使用され得る。本実施例では、SKIPフィールド中の値は、2進数で符号化され得る(例えば、「00000」は、ポートが繰り返される場合にスキップがないことを意味し、「00001」は、同一のポートからのデータの次のブロックが、前のブロックの16ビット後に来ることを意味する)。これは、本説明において以下でさらに説明されるように、サブフレームにおいて複数のポートからのデータを混合するために使用することができる。SKIPフィールドを使用することの実施例は、図21に示さ、図21では、使用されている1のスキップがあり、ワードモードが使用されているため、単一のポートからのデータがサブフレーム中で2回以上繰り返される(すなわち、2つのコマンドワードの間で)。各ポートは、各自のSKIPフィールドを有し、アクティブであるポートは、レジスタ0x02でPORTフィールドによって選択される。

【0201】

図16aをもう1度参照すると、SYNCHRONIZEフィールド(reg. 0x05、B7:B6)は、等時性転送、非同期転送、マルチフレーム転送の間で選択するために使用される。例示的实施形態による、所与のフレームにおける等時性、非同期、マルチフレーム転送の使用に応じて設定される、種々のフィールド値が、図22aに示されている(図22cは、場合によっては使用され得る代替実施形態を示す)。

【0202】

図16aをもう1度参照すると、フィールドPCLKD(reg. 0x07、B7:B4)は、スレーブデバイス中のポートのバスクロックとサンプリングクロックとの間の比を設定する。26.00MHzに対するサポートが提供されない限り、分数オプションによる除算(例えば、両方のエッジ上のクロック、2.5、6.5、および12.5による除算)をコードする必要はない。スペースを節約するために、フィールドは、図22bに示されるように、圧縮様式で符号化されることができる。少なくともいくつかの実施形態では、各ポートは、各自のPCLKDフィールドを有することができる、アクティブであるポートは、レジスタ0x02でPORTフィールドによって選択される。

【0203】

ここで図22aを参照すると、構成1は、デフォルト構成であり、等時性データ転送をもたらすであろう(すなわち、データは、全サブフレーム中で有効である)。構成2は、ポートの第1のチャンネルからのデータの前に2つの状態ビットを追加することによって、非同期転送を取り扱う。第1の状態ビットは、後に続くポートデータが有効であるかどうかを示し(非同期データ転送では、準備ができていないデータは無効と見なされて破棄される)、第2の状態ビットは、受信デバイスがそれらを受け入れることができるかどうかを示す。代替実施形態では、データに先行する1ビットは、伝送デバイスが新しいデータを有し、受信デバイスがそれらを受け入れなければならないことを示すために使用されることができる。代替として、データに先行する1ビットは、スレーブデバイスがデータを受け入れる準備ができていないことを示すために使用されることができ、このデータは、すでにマスタデバイスまたは別の伝送デバイスから入手可能なはずである。構成2は、帯域幅がこれらのビットに利用可能である場合、複数非同期チャンネルに有用である。

【0204】

10

20

30

40

50

構成 3 および 4 は、分数同期化に使用され、例えば、48 kHz チャンネル中で 44.1 kHz においてサンプリングされるデータに対して使用される。この場合、伝送デバイスは、単一のポートからのデータの伝送をいくつかのフレーム中へ拡張し得る。この場合、DS0 および DS1 ビットは、2 つおきのフレームで有効であり、これらのビットがアクティブではないときに、サンプルが有効であるか否かを決定するために使用される、内部分数相加算器（分数カウンタ）をリセットするために使用される。この伝送の開始は、DS0 または DS1 ビットを論理 1 に設定することによって、信号伝達される。マスタデバイス 52 またはスレーブデバイス 54 の中に位置する伝送機は、DS0 および DS1 ビットをアクティブにすることによって、データの流れを制御することができる。

【0205】

図 16 a をもう 1 度参照すると、SUBGROUP フィールド (reg. 0x05、B5 : B3) は、所与のポートからのチャンネルをいくつかのサブグループにグループ化することにより、サブグループ中のチャンネルに対するデータを提供するソースの待ち時間を低減させるために使用され得る。伝送されるチャンネルの数が SUBGROUP フィールドの値に等しいとき、これらのチャンネルからのデータの伝送は、SKIP フィールド中で特定されるニブル数に等しい、いくつかのタイムスロットに対して停止し、次いで、繰り返されるであろう。SUBGROUP フィールドが「111」に等しい場合には、SUBGROUP フィールドは無視され、ポート中のチャンネルからのデータは、1 つのグループ中で伝送されるであろう（例えば、いかなるサブグループもないであろう）。デフォルト値は、「111」である。

【0206】

図 23 は、ワードモードで低遅延を達成するために、同一のポートからのチャンネル間の分割を伴う複数のデータチャンネルを使用するときのフレーム設定（すなわち、フレーム形式）の実施例を示し、SUBGROUP フィールドは、ポートのチャンネルからのデータをいくつかのサブグループに分割するために使用される。SUBGROUP フィールドは、それぞれ、B3 および B4 : B5 によって特定される 2 つの乗数から成る。SKIP が起こる前に伝送されるチャンネルの数は、フィールド B3 と B4 : B5 とによって定義される積に等しい。図 23 の実施例から分かるように、SUBGROUP フィールドは、いくつかのチャンネルがサブグループにおいて一緒にグループ化されるかを特定し、これは、ポートに対する全てのアクティブなチャンネルについて行われ得る。したがって、一例では、図 23 のポートが、オーディオ 0、オーディオ 1、オーディオ 2、およびオーディオ 3 を提供する、4 つのチャンネルを有する場合には、オーディオ 0 およびオーディオ 1 に対するチャンネルは、第 1 のサブグループを形成し、オーディオ 2 およびオーディオ 3 に対するチャンネルは、第 2 のサブグループを形成する。図 23 は、単一のサブフレームの構造のみを示すことに留意されたい。この構造は、フレームを完成させるように 2 回以上繰り返されるであろう。したがって、サブフレームは、次の制御ワードに先立って起こるデータが後に続く制御ワードを備えるフレームの一部であるようにワードモードで定義される。

【0207】

図 16 a を再度参照すると、REPEAT フィールド (reg. 0x05、B2 : B0) は、ポートからのデータがサブフレーム内で何回繰り返されるかを決定するために使用され得る。REPEAT フィールドのデフォルト値は、「000」である。REPEAT フィールドを使用して、複数のデータチャンネルからのデータがワードモードで数回繰り返される時のフレーム設定の実施例が、図 24 に示されている。実施形態では、REPEAT フィールドは、2 つの乗数、すなわち、B0 および B1 : B2 から成る。CHANNEL SELECTION フィールドによって定義されるチャンネルがフレーム中に伝送される回数は、REPEAT フィールドのフィールド B0 と B1 : B2 とによって定義されるような積に等しい。したがって、図 24 の実施例では、ポートの 2 つのデータチャンネル（オーディオ 0 およびオーディオ 1）からのデータが使用される。START フィールドは、ニブル単位で 8 に設定され、LENGTH フィールドは、ニブル単位で 4 に設定され、REPEAT フィールドは、ニブル単位で「001」であり、CHANNEL SEL

10

20

30

40

50

ECTIIONフィールドは、「00000011」であり、SUBGROUPフィールドは、(ニブル単位で)「000」であり、COMMAND SEPARATIONフィールドは、ニブル単位で $112/4 = 28$ である(ニブル単位の全ての値は、10をベースとする形式で値を得るために2で乗算されることを思い出されたい)。図24の実施例では、チャンネルオーディオ0およびオーディオ1からのデータは、サブグループ中で一緒にグループ化され、3回繰り返され、SKIPフィールドが0の値を有するため、サブグループの間に間隙がないことが分かる。ビット112 - 127におけるデータチャンネルオーディオ1からのデータの後に、ビット0からビット127の構造が、フレームを完成させるように2回以上繰り返されるであろうことに留意されたい。したがって、図24は、単一のサブフレームを示し、3つのサブフレームが、フレームを完成させるために使用される。

10

【0208】

全てのデータワードが転送される前にサブフレームが完成した場合、伝送を終了させることができ、その場合、たとえプログラミングエラーが生じたとしても、マスタデバイス52またはスレーブデバイス54は、S、X、またはYワードに書き込むことを可能にされない。したがって、本実施例では、3つのデータペアの終了後に続く他のデータがないので、「010」より大きい任意のREPEAT値によって、図24に示されるものと同じ構成が取得されることができる。各ポートは、各自のREPEATフィールドを有し、アクティブであるポートは、`reg. 0x02`でPORTフィールドによって選択される。

20

【0209】

ビットストリームモードでは、データは、1度に1ビット転送され、データは、1度に1つのデータチャンネルで各データチャンネルから全てのデータを伝送する代わりに、異なるデータチャンネルから多重化される。ビットストリームモードを実装するために、いくつかのフィールドの定義が、この例示的实施形態で変化する。例えば、ポートのビットストリームデータチャンネルからデータを転送するとき、以下のフィールド、すなわち、LENGTH、SKIP、およびSYNCHRONIZEは、現在使用されていない。START、REPEAT、およびSUBGROUPフィールドの解釈は、多重化ビットストリームの性質に適合するように修正される。全ての他のフィールドは不変である。

【0210】

図16aを再度参照すると、ビットストリームモードで、STARTフィールドは、2つのフィールド(HSTARTおよびVSTART)に分割され、ビットストリームデータを転送するポート、または(本説明において以降で説明されるように)ビットストリームデータに変換されるデジタルワードデータを転送するポートに使用される、START IN BITSTREAM MODEフィールド(`reg. 0x04`、B7:B0)であるように修正され得る。HSTARTフィールド(`reg. 0x04`、B7:B4)の値は、単一のサブフレーム内のポートからのチャンネルデータの開始を与え、2進数でコードされることができる。VSTARTフィールド(`reg. 0x04`、B3:B0)中の値は、フレームにおける垂直方向への、ポートからのチャンネルデータの開始を与える。

30

【0211】

ビットストリームモードでは、SUBGROUP AND REPEATフィールドは、2つのフィールドに分割される、ビットストリームモードでのVSPACINGおよびHSPACINGフィールド(`reg. 0x05`、B5:B0)であるように修正され得る。HSPACINGフィールド(B2:B0)中の値は、単一のサブフレーム内のビットストリームの反復間の距離を与える。ビットストリームモードでは、サブフレームの内側の制御ワードが個々のビットに分割されるため、サブフレームは、3つの制御ワードS、X、およびYのうちの1つのビット(例えば、S15、X15、Y15)から定義される間隔、ならびに制御ワードS、X、およびYのうちの1つからの次のビット(例えば、S14、X14、Y14)の開始である。VSPACINGフィールド(B5:B3)中の値は、垂直方向へのビットストリームの反復間の距離を与える。

40

50

【 0 2 1 2 】

ビットストリームモードで使用される構造は、図 2 6 a から 2 8 d の実施例で理解することができる。

【 0 2 1 3 】

ここで図 2 6 a および 2 6 b を参照すると、ビットストリームモードで H S T A R T および V S T A R T フィールドの異なる値を使用することによって達成することができる、異なるフレーム形式の実施例が示されている。両方の場合において、C H A N N E L S E L E C T I O N フィールドは、0 0 0 0 0 0 1 1 に設定される（すなわち、ポートから最初の 2 つのビットストリームデータチャンネルを選択する）。ビットは、左上の隅から始まって、行に沿って行の終了まで移動し、次いで、次の行の最左ビットまで移動し、行の終了まで等、フレームに対するデータの全てが伝送されるまで継続して、伝送される。ビットストリームフレーム形式は、フレーム中の種々のビットに対するタイムスロットを定義することに留意されたい。ビットは、実際には、そのビットに対するデータを有する対応するデバイスによって送信される。例えば、図 2 6 a では、S 1 5 ビットは、マスタデバイス 5 2 によって送信され、ビットストリームデータチャンネル M 0 からの第 1 のビットは、そのチャンネルを有するスレーブデバイスによって送信される。

10

【 0 2 1 4 】

図 2 6 a では、0 0 0 1 の値が、H S T A R T フィールド中にあり、0 0 0 0 の値が、V S T A R T フィールド中にあり、0 0 0 の値が、V S P A C I N G フィールド中にあり（反復率が 1 となり、すなわち、間隙がないことを意味する）、1 1 1 の値が、H S P A C I N G フィールド中にある（1 つだけのチャンネルを有するであろうことを意味する）（これらの値は、図 2 7 に従って符号化される）。示されるように、フレームは、S ワードの S 1 5 ビット、次いで、ポートの第 1 のデータチャンネル M 0 からの第 1 のビット、およびポートの第 2 のデータチャンネル M 1 からの第 1 のビットで始まる。S ワードからの次のビット S 1 4 が送信され、その後、ポートの第 1 のデータチャンネル M 0 からの第 2 のビット、およびポートの第 2 のデータチャンネル M 2 からの第 2 のビットが続く。このパターンは、フレームの S ワードからのビットの全てが送信されるまで継続し、その時に、フレームの X ワードからのビットが、データチャンネル M 0 および M 1 のビットが続く S ワードのビットと同様の方法で送信される。このパターンは、フレームの X ワードからのビットの全てが送信されるまで継続し、その時に、フレームの Y ワードからのビットが、データチャンネル M 0 および M 1 のビットが続く同様の方法で送信される。この時点で、完全フレームが、バス 6 4 上で送信されている。

20

30

【 0 2 1 5 】

図 2 6 b では、0 0 0 1 の値が、H S T A R T フィールド中にあり、0 0 0 0 の値が、V S T A R T フィールド中にあり、0 1 0 = 2 x 1 の値が、H S P A C I N G フィールド中にあり、0 0 0 の値が、V S P A C I N G フィールド中にある（値は図 2 7 に示される表から導出される）。示されるように、フレームは、S ワードの S 1 5 ビット、次いで、ポートの第 1 のデータチャンネル M 0 からの第 1 のビット、およびポートの第 2 のデータチャンネル M 1 からの第 1 のビットで始まる。しかしながら、この時点で、ポートのデータチャンネルからのデータは、繰り返される。したがって、H S P A C I N G フィールドが 2 の値を有するので、データチャンネル M 0 からの第 2 のビットが送信され、その後、データチャンネル M 1 からの第 2 のビットが続く。これは、最大数のビットストリームに達するまで H S P A C I N G 列の後にポートのデータチャンネルからデータを繰り返し、次いで、コマンドワードから次のビットを送信すると考えることができる。次いで、S ワードからの次のビット S 1 4 が送信され、その後、それぞれ、データチャンネル M 0 および M 1 からの第 3 のビットが続く。その後、再度、ポートのデータチャンネル M 0 および M 1 が続く。このパターンは、フレームの S ワードからのビットの全てが送信されるまで継続し、その時に、フレームの X ワードからのビット、それに続いて、データチャンネル M 0 および M 1 のビット 2 回が、S ワードのビットと同様に送信される。このパターンは、フレームの X ワードからのビットの全てが送信されるまで継続し、その時に、フレームの Y ワードか

40

50

らのビット、それに続いて、データチャンネルM0およびM1のビット2回が同様に送信される。いくつかの実施形態では、VSPACINGおよびHSPACINGの符号化は、異なって、例えば、各フィールドに4ビットを使用し、情報の圧縮が使用されない、直接2進符号化を用いて、行われ得る。

【0216】

概して、コマンドチャンネルとして使用されるビットストリームフレーム中のビットストリームチャンネルに書き込むことは違法である。なぜなら、これがバス64をクラッシュするからである。したがって、少なくとも1つの実施形態では、0000の値がHSTARTフィールドに書き込まれる場合には、いかなるデータ出力もバス64上で許可されるべきではない。

10

【0217】

少なくとも1つの実施形態では、ビットストリームモードでは、HSPACINGおよびVSPACINGフィールド中のLSB値が、図27の第1の表に示されるように定義される一方で、HSPACINGおよびVSPACINGフィールド中のMSBおよびLSB+1値は、図27の第2の表に示されるように定義される。全てのビットがHSPACINGおよびVSPACINGフィールドのうちの1つ中で1に設定される場合には、そのフィールドは無視されるであろう。そうでなければ、ビットストリームデータチャンネルからのビットストリームは、LSBによって定義される乗数と、MSBおよびLSB+1によって定義される乗数との積である距離で繰り返されるであろう。HSPACINGフィールドの値が111である場合には、これは、ポートのビットストリームデータチャンネルからのビットストリームが水平方向に繰り返されないであろうことを意味する。VSPACINGフィールドの値が111である場合には、ポートのビットストリームデータチャンネルからのビットストリームが垂直方向へ中断を伴わずに送信されるであろうことを意味する。これの実施例が、図28dに示されている。別の実施形態では、ビットストリームモードでTDMワードを転送するとき、VSPACINGフィールドは、ポート情報(値が符号なし2進プラス1で符号化される、1から8の範囲内の値)を搬送するためにいくつかのビットストリームが使用されるかを示すために使用することができ、HSPACINGフィールドは、これらのビットストリームのうちの第1のものを示すために使用することができる。

20

【0218】

HSTART、VSTART、HSPACING、およびVSPACINGフィールドは、以下の実施例から分かるように、デバイスの各ポートについて含まれるデータチャンネルに対して特定されることに留意されたい。

30

【0219】

ここで図28aを参照すると、統一バス通信プロトコルを使用する、電流(I)および電圧(V)感知を伴うステレオシステム170の例示的实施形態が示されている。ステレオシステム170は、ベースバンドプロセッサ172と、第1のスピーカ176を伴う第1の増幅ユニット174と、第2のスピーカ180を伴う第2の増幅ユニット178と、バス182とを備える。第1の増幅ユニット174およびスピーカ176は、ステレオシステム170用の左オーディオチャンネルであると見なすことができる一方で、第2の増幅ユニット178およびスピーカ180は、ステレオシステム170用の右オーディオチャンネルであると見なすことができる。バス182は、ベースバンドプロセッサ172を第1および第2の増幅器ユニット174および178と連結する。バス182上の伝送は、統一バス通信プロトコルに従っている。

40

【0220】

ベースバンドプロセッサ172は、処理ユニット184と、多重化および同期エンジン186と、メモリ188とを備える。処理ユニット184は、DSP等のプロセッサ、または専用ハードウェア回路であり得る。多重化および同期エンジン186は、統一バス通信プロトコルに従ってバス182を経由して第1および第2の増幅ユニット174および178と通信し、その動作を制御することができるように、ベースバンドプロセッサ17

50

2がマスタデバイスとしての役割を果たすように構成される。

【0221】

第1の増幅ユニット174は、制御ユニット190と、多重化および同期エンジン192と、デルタシグマ変換器194と、増幅器196と、電流センサ198と、電圧センサ200とを備える。多重化および同期エンジン192は、第1の増幅ユニット174がスレーブデバイスとしての役割を果たすように構成され、統一バス通信プロトコルに従ってバス182を經由してベースバンドプロセッサ172と通信することができる。制御ユニット190は、第1の増幅ユニット174の動作を制御する、プロセッサまたは専用ハードウェアおよびファームウェアによって実装されることができる。動作時に、オーディオデータは、増幅のためにベースバンドプロセッサ172から第1の増幅ユニット174へ
10 伝送される。オーディオデータは、変換器194を介してアナログ形式に変換され、増幅器196によって増幅され、スピーカ176によって出力される。増幅ユニット174の動作に係る電流および電圧情報は、それぞれ、電流センサ198および電圧センサ200によって測定され、増幅ユニット174の動作条件を監視するためにベースバンドプロセッサ172に伝送される。増幅ユニット174が、過熱、過電流、または電圧不足等の危険な状態にあることを監視が示す場合、ベースバンドプロセッサ172は、当業者によって理解されるように、危険な状態に対処するように制御命令を増幅ユニット174に送信することができる。

【0222】

第2の増幅ユニット178は、制御ユニット202と、多重化および同期エンジン204と、デルタシグマ変換器206と、増幅器208と、電流センサ210と、電圧センサ212とを備える。多重化および同期エンジン204は、第2の増幅ユニット178がスレーブデバイスとしての役割を果たすように構成され、統一バス通信プロトコルに従ってバス182を經由してベースバンドプロセッサ172と通信することができる。制御ユニット202は、制御ユニット190と同様に実装されることができる。動作時に、第2の増幅ユニット178は、第1のオーディオ増幅ユニット174と同様に動作する。
20

【0223】

多重化および同期エンジンブロック192および204は、ハードウェアブロックとして実装されることができることに留意されたい。さらに、代替実施形態では、制御ユニット190および202は、それぞれ、多重化および同期エンジン192および204と統
30 合されることができる。

【0224】

ここで図28bを参照すると、図28aのステレオシステムに使用することができるビットストリームフレーム形式の例が示されている。このビットストリームフレーム形式では、水平次元のみが、ビットストリームデータチャンネルを多重化するために使用される。この場合、使用される8つのフレームチャンネルがある。第1のフレームチャンネルは、制御ワードに対する制御ビットを伝送するために制御チャンネルとして使用される。第2のフレームチャンネルは、左ステレオチャンネルに対するオーディオデータを伝送するために使用される。第3のフレームチャンネルは、左ステレオチャンネルに対する電圧情報を伝送するために使用される。第4のフレームチャンネルは、左ステレオチャンネルに対する電流情報を伝送
40 するために使用される。第5のフレームチャンネルは、データを伝送するために使用されない。この理由は、現在のシステムと適合性がある動作周波数をサポートするために、7つの代わりに8つのビットストリームを使用することができ、それによって、1つのチャンネルを空のままにすることである。第6のフレームチャンネルは、右ステレオチャンネルに対するオーディオデータを伝送するために使用される。第7のフレームチャンネルは、右ステレオチャンネルに対する電圧情報を伝送するために使用される。第8のフレームチャンネルは、右ステレオチャンネルに対する電流情報を伝送するために使用される。各フレームチャンネルに対するデータは、1度に1ビット送信され、他のフレームチャンネルと時間多重化される。このビットストリームフレーム形式は、合計384ビットに対して、48ビットの長さおよび8ビットの幅を有する。実施例では、クロックレートは、48kHz×64×7
50

= $7 \times 3.072 \text{ Mbit}$ であり、したがって、 $f_{\text{clock}} = 12.288 \text{ MHz}$ である（両方のクロックエッジを使用し、最も近い共通クロック周波数に切り上げることによって求められる）。どれだけ多くの帯域幅を使用するかを決定するために、出力サンプルレートをオーバーサンプリングレートおよびビットストリームフレームチャンネルの数で乗算することができる。例えば、 48 kHz 出力サンプルレート、64に等しいオーバーサンプリングレート、および7つのチャンネルを仮定すると、これは、 21.504 MHz に等しいであろう。しかしながら、これを 12.288 MHz システムと適合性があるようにするために、1つの空のチャンネルが使用され得、 24.576 MHz クロックシステムが使用される。データがクロックの両方のエッジ上で転送される場合、 12.288 MHz クロックが使用される。

10

【0225】

しかしながら、オーディオビットストリームチャンネルLおよびRに使用される全オーディオ帯域幅は、電流および電圧感知には使用されない。これは、スピーカ176および180の機械的共鳴が全オーディオ帯域幅よりはるかに低いためである。加えて、熱時定数は、オーディオデータの変化よりはるかに長い。したがって、インピーダンスモデリングがオーディオデータほど多くの帯域幅を使用しないため、より小さい帯域幅を電流および電圧感知/制御データに使用することができる。

【0226】

図28aの例示的システムでは、所与のフレームチャンネル内で多重化することによって、帯域幅の使用を向上させることができる。この概念は、本実施例と同様に、種々のデータの帯域幅の差がある、オーディオ用途以外の他のタイプの用途に使用されることができる。共通フレームチャンネル内の多重化は、帯域幅の使用の低減をもたらす、複数のデータチャンネルの間でビットストリームフレームチャンネルを共有することによって、達成される。換言すると、1つより多くのデータチャンネルを単一のフレームチャンネルに割り当てることは、オーバーサンプリング、および帯域幅の使用の変化を効果的にもたらす。これは、いくつかのカウンタおよびバッファを使用することによって、容易に実装されることができる。

20

【0227】

ここで図28cを参照すると、いくつかのデータチャンネルが共通フレームチャンネル（例えば、第3のフレームチャンネル）に多重化される、図28aのステレオシステムに使用することができる、ビットストリームフレーム形式の別の実施例が示されている。これは、本実施例では、図28bで使用されるフレーム形式に対するクロック周波数の半分である、より低いクロック周波数を使用して、同一のデータがバス182を經由して伝送されることを可能にする。

30

【0228】

図28cの実施例では、使用される4つのフレームチャンネルがある。第1のフレームチャンネルは、制御ワードに対する制御ビットを伝送するために制御チャンネルとして使用される。第2のフレームチャンネルは、左ステレオチャンネルに対するオーディオデータを伝送するために使用される。第3のフレームチャンネルは、左右のステレオチャンネルに対する電圧および電流情報を伝送するために使用される。第4のフレームチャンネルは、右ステレオチャンネルに対するオーディオデータを伝送するために使用される。もう1度、各フレームチャンネルに対するデータは、1度に1ビット送信され、他のフレームチャンネルと時間多重化される。このビットストリームフレーム形式は、合計192ビットに対して、48ビットの長さ（または行数）および4ビットの幅を有する。行数は、フレームチャンネル中のデータがいくつかのフレームにわたって区分化される（これは、DS0 - DS1ビットを使用することによって行われることができる）場合に、48より長くあり得る。本実施例では、クロックレートは、 $48 \text{ kHz} \times 64 \times (3 + 1) = 4 \times 3.072 \text{ Mbaud} = 6.144 \text{ MHz}$ である。出力サンプルレートは、 48 kHz であり、オーバーサンプルレートは、64倍高く（すなわち、ビットストリームが出力サンプルにつき64ビットを占有する）、1つの制御フレームチャンネルを加えた3つのビットストリームフレームチャンネル

40

50

がある。したがって、これが占有する総帯域幅は、両方のデータエッジ上のサンプリングが使用される場合、 $48\text{ k/s} \times 64\text{ ビット} \times 4 = 12.288\text{ Mbaud}$ ($\text{Mbit/s} = \text{Mbaud}$) または 6.144 MHz である。

【0229】

第3のフレームチャネルは、左オーディオチャネルに対する電圧情報のためのデータチャネルからのデータ、左オーディオチャネルに対する電流情報のためのデータチャネルからのデータ、右オーディオチャネルに対する電圧情報のためのデータチャネルからのデータ、および右オーディオチャネルに対する電流情報のためのデータチャネルからのデータが、垂直方向へ共通フレームチャネル中で一緒に多重化される、多重化フレームチャネルの実施例である。

10

【0230】

この帯域幅制御の概念は、データチャネルからのデータを繰り返すように、追加のフレームチャネルをビットストリームフレーム形式に追加することによって、制御チャネルの帯域幅を制御するように拡張されることもできる。換言すると、ビットストリームフレーム形式の水平方向へビットストリームデータチャネルからのデータを繰り返すことによって、制御フレームチャネルに割り付けられた帯域幅を低減させることができ、したがって、制御ワードの伝送に使用される帯域幅は、より小さい。この実施例は、制御フレームチャネルが、 $1/4 = 25\%$ の帯域幅については4つのビットストリームフレームチャネルのうちの一つである、図28cの実施例と比較して、制御フレームチャネルが、 $1/6 = 16.7\%$ の帯域幅については6つのビットストリームフレームチャネルのうちの一つ

20

である、図28dに示されている。この場合、クロックレートは、 $48\text{ kHz} \times (200/3) \times (2+1) = 9.6\text{ Mbit} = 4.8\text{ MHz}$ である ($200/3$ は、オーバーサンプリング率であり、 $2+1$ は、オーディオデータに対する2つのチャネル、および他のデータに対する1つのチャネルを表す)。

【0231】

ここで図28dを参照すると、例示的实施形態においてビットストリームモードでHSTART、VSTART、HSPACING、およびVSPACINGフィールドの値を設定することによって達成することができる、ビットストリームフレーム形式の実施例が示されている。増幅ユニット174は、スレーブD1として指定され、増幅ユニット178は、スレーブD2として指定される。スレーブD1は、左オーディオチャネル用の第1のポートを提供し、ビットストリームフィールドに対する以下の設定、すなわち、HSTART = 0001 (すなわち、コマンドチャネル後の第1のビットストリームチャネル)、VSTART = 0000 (すなわち、フレームの先頭から開始する)、HSPACING = 001 = 1×3 (すなわち、水平方向へ3ビット後に繰り返す)、およびVSPACING = 000 (すなわち、垂直方向へ次のビットの直後に繰り返す)を有する。スレーブD1は、IV感知用の第2のポートを提供し、ビットストリームフィールドに対する以下の設定、すなわち、HSTART = 0011 (すなわち、コマンドチャネル後の第3のビットストリームチャネル)、VSTART = 0000 (すなわち、フレームの先頭から開始する)、HSPACING = 111 (すなわち、水平方向に繰り返さない)、およびVSPACING = 100 (すなわち、垂直方向へ4ビット後にポートからデータを繰り返す)を有する。スレーブD2は、右オーディオチャネル用の第1のポートを提供し、ビットストリームフィールドに対する以下の設定、すなわち、HSTART = 0010 (すなわち、コマンドチャネル後の第2のビットストリーム)、VSTART = 0000 (すなわち、フレームの先頭から開始する)、HSPACING = 001 = 1×3 (すなわち、水平方向へ3ビット後に繰り返す)、およびVSPACING = 000 (垂直方向に間隙がない)を有する。スレーブD2もまた、IV感知用の第2のポートを提供し、ビットストリームフィールドに対する以下の設定、すなわち、HSTART = 0011 (すなわち、コマンドチャネル後の第3のビットストリームチャネル)、VSTART = 0010 (すなわち、第3のビットストリームチャネル中のフレームの先頭の下方で2ビットを開始する)、HSPACING = 111 (すなわち、水平方向に繰り返さない)、およびV

30

40

50

SPACING = 100 = 4 × 1 (すなわち、垂直方向へ4つビットをスキップした後に繰り返す)を有する。

【0232】

HSPACINGおよびVSPACINGのパラメータを変化させることによって、制御チャンネルに割り付けられる帯域幅の割合と比較して、等時性ビットストリームフレームチャンネルに割り付けられる帯域幅の割合を変化させることができることが分かる。チャンネルが、多くの場合、水平方向へ繰り返される(すなわち、HSPACINGの低い値)場合には、より多くの帯域幅が、制御チャンネルと比較して等時性チャンネルに割り付けられる。VSPACINGが大きい場合、より少ない帯域幅が、制御チャンネルと比較して等時性チャンネルに割り付けられる。本実施例は、4.8MHzで動作する両エッジサンプリングデータラインを使用することによって、200/3(64に近い)倍でオーバーサンプリングされる、2つのオーディオビットストリームチャンネルを依然として転送し、その上、フィードバック用途に適した4つのデータチャンネルも持つことができることを示す。SLIMbusを用いて同じことをしようとすることは、24.576MHzのシステムクロックを必要とするであろう。本実施例では、バスクロック周波数は、完全ステレオオーディオ、デバイス制御、およびIV感知を伴って、わずか4.8MHzである。したがって、統一バス通信プロトコルの少なくとも1つの実施形態とともに、ビットストリームフレームチャンネルの帯域幅制御を使用するという可能性によって、相当量の節電が得られる。

10

【0233】

したがって、異なるビットストリームデータチャンネルからのデータを共通フレームチャンネルに組み込むこと、および異なるフレームチャンネルへのビットストリームデータチャンネルの割付を繰り返すことのうちの少なくとも1つを行うことによって、より効率的な帯域幅の使用、したがって、より効率的なデータ伝送のために、特定の用途に応じて、種々のデータチャンネルおよび制御フレームチャンネルの帯域幅を変化させることができる。

20

【0234】

少なくともいくつかの実施形態では、ビットストリームモードでTDMワードを転送することが可能である。この場合、SUBGROUPフィールドは、異なる解釈を有する。TDMワードは、他の列(すなわち、フレームチャンネル)が存在しなかったかのように、単一の列(すなわち、単一のフレームチャンネル)中で転送されるであろう。図16aを再度参照すると、次いで、ビットストリームモードで、SUBGROUPフィールドは、ビットストリームモードフィールド(reg. 0x05、B5:B3)中でTDM WORDS TRANSFERREDになるように修正される。次いで、SUBGROUPパラメータの以前の定義が可能ではなく、全てのポートは、これらのデータチャンネルの間でいかなる中断も伴わずに、全ての選択されたチャンネルから全てのデータを転送するのである。ビットストリームモードフィールド中のTDM WORDS TRANSFERREDは、ビットストリームフレーム形式で、どのビットストリームフレームチャンネルがワードデータを転送するために使用されるかを示す。実施形態では、時分割多重化(TDM)データを転送するために、ビットストリームフレームチャンネル0-7を使用することができる。

30

【0235】

TDMデータは、説明されるように、ビットストリームモードでサポートされることができるデジタルワードである。ビットストリームフレームチャンネル0は、制御チャンネルとして使用され、この例示的实施形態ではSワード、Xワード、またはYワードからビットを送信するためにのみ使用されるため、いかなるデータ出力にもマップされない。ビットストリームモードでは、DS0ビットは、単一のビットストリームフレームチャンネル内の多数のデータチャンネルをサポートするため、または制御チャンネル中に間隙を伴わずに、各フレーム中でSワード、Xワード、またはYワードを送信することにより、この例示的实施形態では48ビットであり、これらのワードが各々16ビットである、ビットストリームフレーム長と適合性がないデータ長をサポートするために、データ伝送の開始を示すように使用されることができる。例えば、合計で64ビットに等しい、16ビットをそれぞれ

40

50

れ有する、4つのデータチャンネルが、ビットストリームモードで1つのフレームチャンネル中で送信される必要があるとき、これは、いくつかのフレームにわたってデータ同期化を使用して行われることができる。他の実施形態では、ビットストリームチャンネル中に配置されるTDMワードのより良好な同期化を可能にするように、ビットストリームモードさえも、制御ワードの間に間隔があり得る。

【0236】

ここで図29aを参照すると、マスタデバイス52に使用されることができるレジスタの定義の実施例が示されている。代替実施形態が図29bに示されているが、以下の説明は、主に図29aに関係する実施形態に焦点を合わせる。レジスタは、概して、以下の機能、すなわち、アドレスポート、コマンドポート、および設定ポートを提供する、3つのグループに分離される。レジスタは、ここで説明される複数のフィールドの値を提供する。代替実施形態では、他のレジスタを定義することができ、レジスタの1つ以上のビットを使用して、これらのフィールドを実装することができることに留意されたい。代替実施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。

【0237】

COMMAND SEPARATIONフィールド(reg. 0x08、B7:B0)は、実施形態では、長さ8ビットであり、符号なし整数を表す。コマンド分離レジスタの機能は、図25aに示されるようなフレーム形式タイプの値に依存するであろう。図25cは、統一バス通信プロトコルの実施形態で使用されるフレームタイプに応じた、COMMAND SEPARATIONフィールドの定義の代替実施形態の実施例を示す。図25cは、図29bと併せて使用される。

【0238】

フレームタイプがワードモードに設定される場合、ワードフレーム形式が使用され、コマンド分離距離は、ニブル(すなわち、4ビット)の単位で特定され、コマンドワード自体を含まない、2つのコマンドワードの間の距離を与える。分離値は、ゼロであり得、その場合、コマンド情報のみがバス64に転送される(これは図40に示されている)。デフォルト値は、レジスタをプログラミングするために帯域幅の増加を可能にする、48ビットのデフォルトフレーム長と同等あるように設定されることができ、それによって、バス64が迅速に設定され、アタッチされたスレーブデバイスがマスタデバイス54との同期化を迅速に取得することを可能にする。

【0239】

フレームタイプがビットストリームモードに設定される場合、ビットストリームフレーム形式が使用され、制御ビットは、データビットストリームの間でインターレースされるであろう。このモードでは、コマンド分離レジスタは、コマンドビットの間のデータビット数を与えるであろう。例えば、COMMAND SEPARATIONフィールド中の3の値は、4ビットごとにコマンドビットが遭遇されるであろう(すなわち、コマンドビットは、データビットまたは空のタイムスロットであり得る、3ビットによって分離されるであろう)ことを意味する。ビットストリームモードでは、S、X、およびYワードは、ビットストリームデータの間で離間され、ビットストリームに対する低遅延をもたらす。ビットストリームモードが使用される場合、少なくとも1つの実施形態では、スレーブデバイスの全てに対するロック時間を短縮するために、全てのスレーブデバイスがマスタデバイス52との同期化を取得するまで、マスタデバイス52がデータ通信を開始し、全てのデータスロット中でゼロを送信する前に、1つのアドレスが、最初に全てのスレーブデバイスに割り当てられ得る。スレーブデバイス54がマスタデバイス52とのロックから外れる場合、スレーブデバイス54は、より長い全同期化検索を回避するために、COMMAND SEPARATIONフィールドの以前の値および同一のフレームタイプとの同期化を再獲得しようとし得る。フレームタイプを制御する内部レジスタは、それが書き込まれる次のフレームの開始時に更新されるであろう。実施形態では、COMMAND

10

20

30

40

50

SEPARATIONフィールドのデフォルト値は、電源投入時に、384ビットのデフォルトフレーム長と同等である、0x1Cであり得る。ビットストリームモードでコマンドワードの間に間隔を追加することが可能である。検索時間を制限するため、および単純化の目的で、いくつかの実施形態では、図25bに示されるように、限定数のサブフレーム長が使用される。

【0240】

少なくともいくつかの実施形態では、スレーブデバイスが、データ転送においていかなる中断も伴わずに、制御チャンネルに割り付けられた帯域幅の割合およびフレーム形式を同時に変更することができない限り、このレジスタは、スレーブデバイスの内側で使用されない。他の場合において、スレーブデバイスの内部同期化エンジンは、必要な情報を提供され得る。いくつかの実施形態では、COMMAND SEPARATIONフィールドは、全てのポートに対して共有されることができ、各バンクレジスタに対するCOMMAND SEPARATIONフィールドがあり得、BANKビットによってアクティブなフィールドを選択することができる。いくつかの実施形態では、フレームタイプビットを含む、COMMAND SEPARATIONフィールドは、FRAME STRUCTUREフィールドによって置き換えられることができる。これらの構造の両方は、コマンドワードの間の分離、およびワードモードまたはビットストリーム数がデータを搬送するために使用されるかどうかについての情報を含む。

【0241】

SAMPLE RATE RATIOフィールド(reg. 0x09)は、TDM出力を伴うシグマデルタ変換器に対するオーバーサンプリングレートを示すために使用され得る。このレジスタは、単純な方法でデシメーションおよび補間フィルタを設定するために、およびマルチフレーム同期化のサポートのために使用され得る。この情報は、多くの場合、COMMAND SEPARATION値(FRAME STRUCTURE値とも称することができる、図69a参照)およびREPEATパラメータから、またはFRAC TIONALレジスタから直接推測することができる。しかしながら、データチャンネルについて定義されるようなバスクロック周波数と出力サンプルの周波数との間の比を直接プログラムすることが便利である、場合がある(例えば、現在のBANKの変更中、およびREPEATを使用してパラメータが6回以上繰り返される場合)。これは、SAMPLE RATE RATIOレジスタを使用することによって、達成される。加えて、このレジスタは、いくつかの実施形態では、マルチフレーム同期化のために保留されるフィールドを備え得る。この例示的実施形態では、SAMPLE RATE RATIOフィールドは、MULTI FRAME、FIRST STAGE、OPTIONAL STAGE、SECOND STAGE、およびTHIRD STAGEというサブフィールドを備える。

【0242】

MULTI FRAMEフィールド(reg. 0x09、B7)は、サブフレーム同期化の代わりに、DS0およびDS1ビットによる同期化を選択するために使用され得る。ポート情報は、通常、S、X、およびYワードの最後のビットに関して定義される。しかしながら、場合によっては、効率的なデータ転送のために、サブフレームの開始よりもむしろ、異なるタイミング基準、例えば、1つおきのサブフレームの開始が使用される。これは、MULTI FRAMEビットを真に設定し、同期化のソースとしてDS0ビットまたはDS1ビットのいずれかを選択すること(これはSYNCHRONIZEを「10」または「11」に等しく設定することによって行われる)によって、達成することができる。MULTI FRAMEビットが偽であるか、またはこのレジスタが欠落している場合、DS0およびDS1ビットが同期化に使用されるときに分数転送のみがサポートされる。

【0243】

FIRST STAGEフィールド(reg. 0x09、B6:B5)は、多段同期サンプルレート変換器の第1の段階に対するサンプルレート変換率を定義するために使用さ

10

20

30

40

50

れ得る。FIRST STAGEフィールドがどのようにして符号化され得るかという実施例が、図29cに示されている。サンプルレート変換率は、図29c - 29eに示されるように、フィールドの結果によって見出される。

【0244】

OPTIONAL STAGEフィールド(reg. 0x09、B4 : B3)は、多段同期サンプルレート変換器の随意的な段階に対するサンプルレート変換率を定義するために使用され得る。この段階は、必要であれば、26.00MHz周波数の直接サポートに使用することができる。スペースを節約するために、OPTIONAL STAGEフィールドがどのようにして符号化され得るかという実施例が、図29dに示されている。

【0245】

SECOND STAGEフィールド(reg. 0x09、B2)は、多段同期サンプルレート変換器の第2または第3の段階に対するサンプルレート変換率を定義するために使用され得る。このセクションは、セクション1および3の反対側で実装することができる、すなわち、全体的なアクションがデシメーションである場合、第2の段階は、補間を行うことができ、全体的なアクションが補間である場合、第2の段階は、所与の係数によるデシメーションを行うことができる。これは、いくつかのシステム構成で、より良好な性能を可能にする。スペースを節約するために、SECOND STAGEフィールドがどのようにして符号化され得るかという実施例が、図29eに示されている。

【0246】

THIRD STAGEフィールド(reg. 0x09、B1 : B0)は、多段同期サンプルレート変換器の最後の段階に対するサンプルレート変換率を定義するために使用され得る。スペースを節約するために、THIRD STAGEフィールドがどのようにして符号化され得るかという実施例が、図29fに示されている。図29gおよび29hは、完全なマルチフォーマットデシメータおよび補間器システムを示す。具体的には、図29gは、複数のサンプル率比および種々のクロック周波数との適合性に好適である、マルチフォーマットデシメータシステムの実施例を示す。各サンプルレート変換セクションは、選択されたサンプル率比にかかわらず、全体的な利得が一定であるように、倍率が先行するか、または後に続き得る。図29hは、複数のサンプル率比および種々のクロック周波数との適合性に好適である、マルチフォーマット補間器システムの実施例を示す。各サンプルレート変換セクションは、選択されたサンプル率比にかかわらず、全体的な利得が一定であるように、倍率が先行するか、または後に続き得る。これが有益であれば、セクションの順番を変更すること、またはサンプルレート変換器の総利得が一定であることを確実にするようにセクションの間にスケールリングを追加することが可能である。

【0247】

FRACTIONALフィールドは、2つのサブフィールド、すなわち、FRACTIONAL LOW(reg. 0x0A、B7 : B0)およびFRACTIONAL HIGH(reg. 0x0B、B7 : B0)を有し、フラクショナルフローのデータ伝送に使用され得る。これは、システムクロックと、(REPEATフィールドによって特定される)フレームあたりのサンプル数および(LENGTHフィールドによって特定される)フレームあたりのクロックサイクル数との間の比がすでに定義されているためである。FRACTIONALフィールドを設定することによって、分数オーバーサンプリング率 Y/X が、FRACTIONAL HIGH(すなわち、 Y)フィールドとFRACTIONAL LOW(すなわち、 X)フィールドとの間の比として定義される。マスタデバイス52については、この機能は、分数データフローを制御するために使用され、随意的である。デフォルト値は全てゼロである。FRACTIONAL HIGH(X)フィールドおよびFRACTIONAL LOW(Y)フィールドはまた、デシメータおよび補間器によって使用されるオーバーサンプリング率を定義するために使用され得る。他の実施形態では、このフィールドの符号化は、異なり得るが、依然として2つの2進符号化ワードの間の比を示すために使用される。

【0248】

10

20

30

40

50

ADDRESSフィールド(レジスタ0x0Cおよび0x0D)は、どのレジスタ上で、マスタデバイス52、および随意にスレーブデバイス54が、READ、WRITE、またはFUNCTION動作を行うかを選択するために使用され得る。それはまた、内部関数を実行するときスレーブデバイスによって使用されることもできる。

【0249】

FUNCTIONフィールド(reg.0x0E)は、FRAME CONTROLレジスタを使用してFUNCTION動作を開始するとき、どのような動作が開始されるかを定義するために使用され得る。

【0250】

以下のフィールド、すなわち、FRAME CONTROL、CURRENT BANK、およびSLAVE DEVICE ADDRESSを有する、フレーム制御レジスタフィールド(reg.0x0F)が使用され得る。これらのフィールドは、レジスタの1つ以上のビットを使用して、実装されることができ、代替実施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。例えば、代替実施形態では、フレーム制御レジスタフィールドは、DATA SIZEおよびPOWERフィールドを備え得る。

【0251】

FRAME CONTROLフィールド(reg.0x0F、B7:B5)は、READ/WRITE/FUNCTION動作を開始するため、およびWRITE動作の進行を監視するために使用され得る。このフィールドへの書き込みの解釈の実施例が、図29iに示されている。このフィールドから読み取るとき、返される値は、現在のフレーム動作に等しいであろう。このフィールドに書き込んだ後、所望の動作が1回実行されることができ、その後、動作は、この例示的实施形態ではPING動作であり得る。FRAME CONTROLレジスタは、次のフレームに対する動作モードを選択する。したがって、READ動作を開始するために、READ動作に対する正しい3ビットコード定数(8または16ビット読み取りデータのいずれか)が、マスタデバイス52の内側のFRAME CONTROLレジスタに書き込まれ、その後、デバイスレジスタREAD動作が次のフレーム中で始まるであろう。

【0252】

CURRENT BANKフィールド(reg.0x0F、B4)は、データ転送のパラメータを特定するためにスレーブデバイス54が使用するべきである、現在のレジスタバンクを選択するために使用され得る。このフィールドは、PINGおよび他の動作中に異なる意味を有する。デフォルト値はゼロである。PING動作中、このフィールドは、スレーブデバイス54を制御するためにどのメモリバンクが使用されるかを選択するために使用されることができる。したがって、フレーム形式またはオーディオモードを変更するために、バス64にアタッチされたデバイスが異なるメモリバンクを使用し得るように、CURRENT BANKビットを変更するPINGコマンドが最初に発行される。例えば、READ、WRITE、およびFUNCTION動作等の他の動作中、このフィールドは、どのメモリバンクに現在の動作(すなわち、READ、WRITE、またはFUNCTION)が適用されるかを選択するために使用されることができ、これは、オーディオ出力においていかなる故障も伴わずに2つの異なるオーディオモードの間で切り替えること等の、出力においていかなる故障も伴わずに2つの異なるデータモードの間でシームレスに切り替えるように、2つの交互レジスタセットの間で選択するために使用されることができる。スレーブデバイス54は、新しい設定が次のフレームの開始時に使用されるように、各自のレジスタを内部で更新するべきである。これを行う1つの方法は、BANKフィールド(X12)上にCURRENT BANKビットをビットコピーすることである。スレーブレジスタ0x02から0x07中の全ての値は、X12タイムスロットによって(すなわち、Xワードで)選択される。例示的实施形態では、全てのスレーブデバイスは、ポートへのデータおよびポートからのデータのストリーミングのために2つの

10

20

30

40

50

動作モードを有することができる。データモード間の実際の変化は、Xコマンドワードの伝送後の2つのフレーム境界後に起こるのである(すなわち、スレーブデバイスは、例えば、X12ビットの2つの等しい読み取り値が取得されたときに、最初にその内部バンクを変更することができる)。これは、BANKビットが、1つの値、例えば、ゼロから、別の値、例えば、1に変化しているときに、スレーブデバイスが、以前に使用された値とは異なる値とともに補助レジスタバンク(例えば、レジスタ0x02 - 0x07の代替セット)を使用するであろうことを意味する。この変化は、BANKビットがPINGフレーム中で変化した第2のフレームの後に、全てのスレーブデバイスで起こる。2つのレジスタバンクは、同一のBANKビットによって、READ、WRITE、およびFUNCTION動作中に選択される。CURRENT BANKビットは、次のフレームに対するX12(BANK)ビットの値を決定するために、マスタデバイス52によって使用される。

10

【0253】

SLAVE DEVICE ADDRESSフィールド(reg. 0x0F、B3:B0)は、最上位ビットとしてB3を有し、選択されるスレーブデバイスアドレスを特定するために使用され得る。リセット後のデフォルト値は、「0000」である。スレーブデバイスアドレス「1111」が使用されるとき、コマンドが全てのデバイスにブロードキャストされるであろう。しかしながら、同時に複数のデバイスから読み取る場合に曖昧性をもたらし得るため、ブロードキャストコマンドの使用については注意すべきである。また、以下でさらに詳細に説明されるように、いくつかのデバイスを同時に選択することができるように、マスタデバイス52によって選択される特定のデバイス群を定義することができる、実施形態があり得る。

20

【0254】

INTERFACE CONTROLレジスタ(reg. 0x010)は、動作周波数、クロックソース、およびマルチマスタデバイス動作の選択を設定すること等のバス64の基本動作を制御するために使用され得る。例示的实施形態では、INTERFACE CONTROLレジスタは、以下のサブフィールド、すなわち、図32aにも示される、MCLKD、CLOCK SOURCE、ENABLE MULTI MASTER、ENABLE MASTER IRQ、およびREQUEST MULTI MASTERを備える(図29bに対応する代替实施形態が図32bに示されている)。これらのフィールドは、レジスタの1つ以上のビットを使用して、実装されることができる。代替实施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。リセットイベント後のINTERFACE CONTROLフィールドのデフォルト値は、「00010000」に設定される。

30

【0255】

MCLKDクロック分割フィールド(reg. 0x10、B7:B4)は、バス64に使用されるクロック信号(すなわち、クロックバス信号)の周波数を設定するために使用され得る。実施形態では、バスクロック信号の周波数は、周波数分割を使用することによって、外部クロックソース信号(EXTCLK)から導出され得る。周波数分割は、INTERFACE CONTROLフィールドのMCLKDフィールドによって決定される。例示的实施形態では、外部バスクロック信号は、単一の立ち上がりエッジのみを有するが、マスタデバイス52の内部回路は、少なくとも2つの立ち上がりエッジを使用する。したがって、全ての分割率は、2という因数に基づく。図33aは、少なくとも1つの例示的实施形態では、スペースを節約するために、どのようにしてMCLKDフィールドが符号化され得るかという実施例を示す。図33bは、図29bに対応する実施形態で使用され得る、周波数分割を示し、その場合、周波数分割に使用される因数は、単一ワイヤバス実施形態については最後から2番目の列、2ワイヤバス実施形態については最後の列で特定される。クロック分割率が「0000」であるとき、バスクロック信号が停止し、クロック出力が低く駆動されるであろう。PLLを採用するシステムは、動作モード、例え

40

50

ば、低または高周波数動作、あるいは分数PLLクロックシステムに対する約数を設定するために、カスタムレジスタを使用することができる。MCLKDフィールドのデフォルト値は、「0001」に設定される。

【0256】

少なくともいくつかの実施形態では、クロック信号ラインもまた、電力を節約するように、静的レベル（例えば、低論理または高論理値のいずれか）で保たれ得る。この場合、クロック分割器は、分割率が整数ではない場合を除いて、立ち上がりクロックエッジ上の出力を変化させる。したがって、電力低下モードに整数除算値を使用することが利点である。加えて、クロック変化は、フレームの開始時に起こる。クロック分割因数が0である場合、いかなるクロック出力もない。この条件で、マルチマスタモードが有効にされない場合には、変化がクロックライン上で起こる前にデータラインが変化すれば、クロックが再び再起動する。これは、非常に低電力の動作に使用されることができる。いくつかの実施形態では、本システムは、完全バッファ状態を信号伝達し得るUARTコントローラからの要求に迅速に応答することができるように、クロックを再び開始する明確に定義された時間、例えば、1マイクロ秒を有するであろう。データラインが変化した後には供給される初期クロック信号は、約1マイクロ秒であり得る、非常に迅速な起動時間という利点を有する、RC駆動型緩和発振器等の通常使用される（典型的には、水晶発振器）ほど正確ではないクロック信号であり得る。緩和発振器は、高品質水晶発振器よりかなり多くのジッタを有し、サンプリングイベントで精度の増加があり得る、高品質オーディオ目的に適していない。後に、主要水晶発振器が完全に安定化したとき、マスタデバイス52は、クロックソースをこの他のソースに変更し得る。このスキームを使用することによって、低電力モードから回復するときに、UARTコントローラからデータを迅速に捕捉し、それによって、いかなるデータも損失しないように、高速起動時間を有するとともに、精密オーディオ目的で高品質の低ジッタクロック信号を依然として供給することが可能である。

【0257】

CLOCK SOURCEフィールド（reg. 0x10、B3）は、2つの異なるクロックソースの間で選択するために使用され得る。マスタデバイス52がこの特徴をサポートすることは随意的である。高周波数クロック、例えば、オーディオ用途のための19.20MHz、および低周波数クロック、例えば、低電力動作に使用される32.768kHzがあり得る。これらの場合において、マスタデバイス52は、データにおいていかなる故障も起こらないように、これらのクロックソースの間でゲート制御するか、または切り替える。クロックソース間の変更は、クロック信号が低であるときにフレームの終了時に行われることができる。別の実施形態では、マスタデバイス52は、クロックを無効にし、別のデバイスがクロックを駆動することができるという可能性のために、バスホルダを使用してクロックラインを低いままにさせることができる。いくつかの実施形態では、クロック発振器のうちの1つは、高品質クロックソースであり得、他方は、より不良なジッタ性能を伴う高速安定化クロックソースであり得る。

【0258】

ENABLE MULTI MASTERフィールド（reg. 0x10、B2）は、複数のマスタデバイスの使用を有効にするために使用され得、1に設定されるときはアクティブである。マスタデバイスがこのビットを1に設定する場合、別のマスタデバイスは、コマンドフィールドセクション中にコマンドを書き込み始めることができる。それは、PING動作中にX8ビットの値を決定するために使用される。このビットから読み取るとき、それは、最後に読み取られたX9ビットの状態を返すであろう。このビットは、バスクロック信号を生成するマスタデバイスによってアクティブにされることができる。デフォルト設定は1である（すなわち、有効にされる）。

【0259】

ENABLE MASTER IRQ MASKフィールド（reg. 0x10、B1）は、PING動作中のX9ビットのアクティブ化に基づいて、割り込みを有効にするために使用され得る。1に設定されるとき、X9ビットは、マスタデバイス52へのIRQ

10

20

30

40

50

を生成するであろう。さらに、X9ビットは、スレーブデバイスからのIRQのように扱われることができる(すなわち、S0ビットがこの例示の実施形態で設定されるべきである)。デフォルト設定はゼロである(すなわち、無効にされる)。

【0260】

REQUEST MULTI MASTERフィールド(reg. 0x10、B0)は、X9ビット上にコピーされ得、マルチマスタ動作に対する要求を信号伝達するために使用され得る。許可された場合、それは、このビットを再度解放するべきである(すなわち、それをゼロに設定する)。このビットを読み取るとき、それは、バス64から読み取られた最後のX9ビットの状態を与えるであろう。

【0261】

IRQ MASKレジスタ(reg. 0x11)は、どのイベントがマスタデバイス52への割り込み要求を生成することを可能にされるかを決定するために使用され得る。図34aは、IRQ MASKレジスタで使用されることができサブフィールドの実施例を示す(代替実施形態が図29bに対応する図34bに示されている)。図34bの実施形態では、サブフィールドは、BUS ERROR MASKフィールド、IO ERROR MASKフィールド、ATTENTION MASKフィールド、S0 DELAY ENABLEフィールド、ATTACHMENT MASKフィールド、WRC MASKフィールド、RDC MASKフィールド、およびFRAME DONE MASKフィールドを含む。これらのフィールドは、レジスタの1つ以上のビットを使用して、実装されることができ、代替実施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。IRQ MASKフィールド内のビットの各々は、マスタデバイス52によって読み取られるか、または書き込まれる。スレーブデバイスは、例えば、「10」以上のスレーブ状態レベルを示して、またはデバイス状態レベルを変更することによって、これらのビットのうちのいくつかを間接的に設定または消去し得る。代替実施形態(すなわち、図34a)では、WRCおよびRDCビットが、IOC(IO動作完了)ビットおよび(マルチマスタ動作に使用される、XおよびYワードの変化を監視する)COMMAND ERRORビット、ならびに対応するマスクビットによって置換されるであろう。

【0262】

BUS ERROR MASKフィールド(reg. 0x11、B7)は、マスタ割り込みを有効または無効にするために使用され得る。このフィールドが高に設定されるとき、状態レジスタ中のアクティブなバスエラービットに基づいて割り込みを有効にするであろう。この割り込みは、アクティブローレベル出力によって信号伝達され、状態レジスタ0x12を読み取るときに消去される。

【0263】

IO ERROR MASKフィールド(reg. 0x11、B6)は、I²C割り込みを有効または無効にするために使用され得る。このフィールドが高に設定されるとき、状態レジスタ中のアクティブなIOエラービットに基づいて割り込みを有効にするであろう(すなわち、READ/WRITE/FUNCTIONコマンドに続く確認ビットS14が設定されない)。この割り込みは、アクティブローレベル出力によって信号伝達され、状態レジスタを読み取るときに消去される。

【0264】

ATTENTION MASKフィールド(reg. 0x11、B5)は、S0ビットをアクティブにすることによって、スレーブデバイス注意要求に基づいてマスタ割り込みを有効または無効にするために使用され得る。このフィールドが高に設定されるとき、「10」または「11」のスレーブ状態レベルに基づいて割り込みを有効にするであろう。換言すると、ATTENTION MASKフィールドが有効にされるとき、マスタデバイス52は、スレーブデバイス54が注意を要求するときはいつでも割り込みを生成するであろう。割り込みは、アクティブローレベル出力として、マスタデバイス52によって

10

20

30

40

50

コントローラに信号伝達され、状態レジスタを読み取るときに消去される。スレーブデバイス54が依然として注意を要求しており、ATTENTION MASKフィールドが依然として次のフレーム中で有効にされる場合、新しい割り込みが生成されるであろう。マスタデバイス52は、複数の割り込みが生成されることを回避するように、IRQが起こるときにATTENTION MASKフィールドを消去することができる。

【0265】

S0 DELAY ENABLEフィールド(reg. 0x11、B4)は、スレーブデバイス54が注意を要求するときに、READ、WRITE、またはFUNCTION動作の遅延を有効または無効にするために使用され得る。S0 DELAY ENABLEフィールドが高に設定されるとき、S0タイムスロット中にバス64から読み取られる値に基づいて、強制PING動作を有効にするであろう。スレーブデバイス54は、「10」または「11」のスレーブデバイス状態を有するときに、注意の必要性を信号伝達するようにS0ビットをアクティブにすることができる(すなわち、スレーブデバイス54は、S0タイムスロット中にその状態レジスタのMSBをコピーすることができる)。現在の動作がREAD、WRITE、またはFUNCTION動作である場合、次のフレームまで遅延させられ、S0 DELAY ENABLEフィールドおよびS0ビットが1に等しければPING動作によって置換されるであろう。次のフレームの開始時に、READ、WRITE、またはFUNCTION動作が再試行されるであろう。PING動作は、S0ビットの値にかかわらず、不変に続行するであろう。S0 DELAY ENABLEフィールドを有効にすることによって、任意のスレーブデバイスが注意を必要とする場合に、多くても1つのフレームの待ち時間が生じるであろう。コマンドワード以外のデータトラフィックが、デバイス状態を読み取りながら、依然として継続することができる。S0 DELAY ENABLEフィールドが「0」である場合、S0ビットの値にかかわらず、任意のREAD、WRITE、またはFUNCTIONトランザクションが続行する。マスタデバイス52は、スレーブレジスタの読み取りまたは書き込みを阻止することを回避するように、IRQが信号伝達されるときにS0 DELAY ENABLEフィールドを消去することができる。

【0266】

ATTACHMENT MASKビット(reg. 0x11、B3)は、これらの割り込みを有効または無効にすることによって、デバイスアタッチ監視を制御するために使用され得る。このビットが低であるとき、アクティブなATTACHMENTビットによる任意の割り込みが無効にされる。このフィールドが高に設定されるとき、アタッチされている、またはバス64から除去されているデバイスに基づいて、割り込みを有効にし、それによって、ATTACHMENTビットを高に設定するであろう。スレーブデバイスがその状態値を変化させる場合、またはスレーブデバイスがバス64から除去される(それによって、その状態値を効果的に変化させるPING動作中に信号伝達しない)場合、またはスレーブデバイスがもはやマスタデバイスと同期しなくなり、それによって、信号伝達を停止する場合、または(物理的アタッチ後または同期化が取得された後に)スレーブデバイスがマスタデバイスと同期し、この状態値がPING動作によって読み取られた場合に、これが起こり得る。ATTACHMENT割り込みは、状態レジスタ0x12(すなわち、ATTACHMENTビットを含むレジスタ)を読み取るときに消去される。読み取られる状態値がマスタデバイス52からの介入の必要性を示す(すなわち、状態値が「10」または「11」である)場合、マスタデバイス52のコントローラに対する通常動作は、割り込みのソースを見出すように、スレーブ状態レジスタからデバイス状態値をリードバックすることであろう。これが完了した後、注意を必要とするスレーブデバイスへのレジスタ制御コマンドを書き込むことによって、任意の必要なアクションを実行することができる。

【0267】

WRC MASKフィールド(reg. 0x11、B2)は、割り込みを有効または無効にするために使用され得る。IRQ MASKレジスタ0x11は、外部または内部の

10

20

30

40

50

専用（すなわち、物理的）割り込みラインを通過してマスタデバイス52に直接向かう、割り込みを有効または無効にする。これらの割り込みは、典型的には、WRITE動作が完了した後に、バス64を経由し、マスタデバイス52からの専用ラインを通らずに通信される、スレーブデバイスからの割り込みと同一ではない。割り込みは、専用ラインを通過してマスタバスコントローラから直接通信される。このフィールドが高に設定されるとき、アクティブなWRITE動作の完了に基づいて、割り込みを有効にするであろう。これは、WRCビット上の高・低遷移から見出されることができる。割り込みは、アクティブローレベル出力によって信号伝達されることができる。割り込みは、状態レジスタ0x12を読み取るときに消去されることができる。代替実施形態（すなわち、図34a）では、このビットは、IO動作の完了をマスクするため、およびIO動作（READ/WRITE/FUNCTION）が完了したときに割り込みを有効にするために使用され得る。割り込みのマスクングは、WRC MASKフィールドに類似するであろう。

10

【0268】

RDC MASKフィールド（reg. 0x11、B1）は、WRC MASKフィールドについて上記で説明されたものと同様に、READ動作が完了した後に、マスタデバイスからの割り込みを有効または無効にするために使用され得る。このフィールドが高に設定されるとき、アクティブなREAD動作の完了に基づいて、割り込みを有効にするであろう。これは、RDCビット上の高・低遷移から見出すことができる。割り込みは、アクティブローレベル出力によって信号伝達されることができる。割り込みは、状態レジスタ0x12を読み取るときに消去されることができる。代替実施形態（すなわち、図34a）では、このビットは、COMMAND ERROR割り込み、すなわち、主要マスタデバイス以外のデバイスによって行われる、XおよびYワードに位置するビットの変化をマスクするために使用され得る。このビットは、アクティブハイである。これは、マルチマスタモードが許可されないときに、安全対策として使用され得る。このIRQマスクビットは、マルチマスタ動作モードでは無効にされるべきである。

20

【0269】

FRAME DONE MASKフィールド（reg. 0x11、B0）は、立ち上がりクロックエッジ上のフレームの完了時にアクティブにされる、マスタ割り込み（代替実施形態では専用ラインを通して、またはI²Cを通して信号伝達される）を有効または無効にするために使用され得る。このフィールドが高に設定されるとき、内部フレームカウンタの完了に基づいて、割り込みを有効にするであろう。マスタデバイス52中の内部フレームカウンタは、FRAME DONE MASKフィールドを変更するための例示的タイミング図を示す、図35に示されるように、変更モード中に依然として作動しているべきである。変更モードでは、インターフェース70中の電力トランジスタ76（図3b参照）はオンにされる。

30

【0270】

図36aを参照すると、例示的实施形態では、INTERFACE STATUSレジスタ（reg. 0x12）は、いくつかのサブフィールドを備える（代替実施形態が図29bに対応する図36bに示されている）。図36bの例示的实施形態では、INTERFACE STATUSレジスタは、BUS ERRORフィールド、IO ERRORフィールド、STATUS1およびSTATUS0フィールド、ATTACHMENTフィールド、WRCフィールド、RDCフィールド、FRAME DONEフィールド、およびSLAVE STATUSフィールドを備える。これらのフィールドは、レジスタの1つ以上のビットを使用して、実装されることができる。代替実施形態では、異なる実装をこれらのフィールドに使用することができ、フィールドが異なって順序付けられ得、または使用される他のフィールド、あるいは使用される他のレジスタがあり得る。リセットイベント後のINTERFACE STATUSレジスタのデフォルト値は、「00000000」であり得る。一般に、INTERFACE STATUSレジスタは、バス64からの状態情報を有し、IRQを制御するために使用される。代替実施形態（すなわち、図36a）では、WRCおよびRDCビットは、IOC（IO動作完了）ビットおよび

40

50

(マルチマスタ動作に使用される、XおよびYワードの変化を監視する) COMMAND ERRORビット、ならびに対応するマスクビットによって置換されるであろう。マルチマスタ動作は、PING動作中に、これらのビットの両方、ならびにX9およびX8制御ビットを使用し得る。これは、複数のI²C接続を伴う状況で、または試験目的で使用することができ、外部ユニットが、システムハードウェアまたはソフトウェアへの変更を必要とすることなく、バス64に接続されたデバイスの内部状態をリードバックし得る。

【0271】

動作中に、INTERFACE STATUSフィールドは、バス64からの状態情報を提供し、マスタデバイス52へのIRQをアクティブにするために使用され得る。それは、マスタデバイス52またはスレーブデバイス54による、READ、WRITE、PING、およびFUNCTION動作によって制御される。INTERFACE STATUSレジスタ内のビットの各々は、マスタデバイス52によって読み取られ、使用されることができる。このフィールドは、マスタデバイス52とスレーブデバイス54との間のデータ通信を制御することを支援し、また、バスにアタッチされたスレーブデバイスの状態を監視するために使用されることもできる。

【0272】

BUS ERRORフィールド(reg. 0x12、B7)は、アクティブハイであるときに違法バス動作が起こったことを示すために使用され得る。この条件は、バス64上の値が、マスタデバイス52がバス64を駆動している(例えば、スレーブデバイスが同期化パターンと対立している)ときにあるべき値とは異なる場合に、検出されることができる。違法バス動作が検出されると、それは、BUS ERRORフィールドが読み取られるまで設定されたままとなるであろう。違法バス動作が観察され、対応するIRQ MASKビットが有効にされる(すなわち、高に設定される)場合に、割り込みが生成されるであろう。マルチマスタ動作状態が許可された(PING動作におけるRELEASE BUSビット(X8)がゼロである)とき、コマンドワードのXおよびYフィールド中の任意のバス対立は、別のマスタデバイスによって引き起こされると仮定されるため、これらは無視されるべきである。この場合、他のマスタデバイス52がパリティビットを書き込むであろう。

【0273】

IO ERRORフィールド(reg. 0x12、B6)は、READ/WRITE/FUNCTION動作中のエラー、またはPING動作中のパリティエラー(すなわち、データ中のエラー)を示すために使用され得る。IO ERRORフィールドは、スレーブデバイス54がREAD、WRITE、またはFUNCTIONコマンドを確認しない場合に設定されるであろう。例えば、READまたはWRITE、あるいはいくつかのFUNCTIONコマンドに続く確認ビットS14が、アドレス指定されたスレーブデバイスによってアクティブにされない場合には、IO ERRORフィールドが設定され、確認ビットS14がPINGまたはいくつかのFUNCTIONコマンドの完了に続いてアクティブにされる場合にも、IO ERRORフィールドが設定されるであろう。IO ERRORフィールドは、状態レジスタ0x12を読み取るとリセットされる。このスキームは、単一のレジスタを読み取ることによって、バス64上の全てのバストラフィックがエラーについてチェックされることを可能にする。

【0274】

STATUS1およびSTATUS0フィールド(reg. 0x12、B5:B4)は、バス64にアタッチされる任意のスレーブデバイスから読み取られる最高状態レベルを記憶するために使用され得る。任意のスレーブデバイスが、STATUS1およびSTATUS0フィールドによって示されるよりも高い状態レベルを有する場合、これらのフィールドは、次のPING動作中に、この新しい値に更新されるであろう。INTERFACE STATUSレジスタの読み取りは、STATUS1およびSTATUS0フィールドを消去しないであろう。これらのフィールドは、全PING動作および電源投入リセットイベント中に更新され、したがって、有効である。実施例として、スレーブ状態値は

10

20

30

40

50

、リセット後に「00」であり、スレーブデバイスがバス64にアタッチされた後は「01」である。次いで、スレーブデバイスが緊急の注意を必要とし、PING動作中にスレーブ状態として「11」を信号伝達すると仮定すると、次いで、STATUS1およびSTATUS0フィールドは、この値に更新される。スレーブ状態レジスタを読み取った後、値は依然として「11」である。次のフレーム中に、最高スレーブ状態レベルが「10」であると仮定されたい。このPING動作の終了時に、STATUS1およびSTATUS0フィールドは、この新しい値（「10」）に更新されるであろう。これは、スレーブデバイスがこのような場合を示すときに、エラーが最初に消去されるように行われる。対応するINTERRUPT MASKビットが有効にされ、注意を要求するスレーブデバイスの結果として割り込みが生成された場合、INTERFACE STATUSレジスタを読み取ると、マスタデバイス52からコントローラへのIRQラインが消去されるであろう。スレーブデバイスが依然として注意を要求し、割り込みマスクビットが設定される場合、新しい割り込みが、次のPING動作中に生成されるであろう。通常動作中に、このフィールドからの読み取りは、「01」を返すであろう、すなわち、1つ以上のスレーブデバイスがバス64にアタッチされ、特殊サービスの要求はない。したがって、STATUS1およびSTATUS0フィールドは、任意のスレーブデバイスがバス64にアタッチされているかどうかを決定するため、または低い優先度（例えば、「10」の状態レベル）あるいは高い優先度（例えば、「11」の状態レベル）を伴う注意を必要とするスレーブデバイスを区別するための両方に使用されることができ、いかなるスレーブデバイスもバス64にアタッチされていない場合、STATUS1およびSTATUS0フィールドは、「00」に設定されるであろう。

【0275】

ATTACHMENTフィールド（reg. 0x12、B3）は、最後のPING動作以降に、スレーブデバイスが接続解除、またはバス64に接続した場合を示すように、アクティブハイに設定され得る。ATTACHMENTフィールドの値は、PING動作中にXおよびYワードで示されるスレーブデバイスの状態レベルを、スレーブデバイスの以前の状態レベルと比較することによって、見出されることができ、任意のスレーブデバイスの状態レベルが{“01”, “10”, “11”}から“00”に変化した（すなわち、スレーブデバイスがバス64から接続解除した）場合、ATTACHMENTフィールドは、低に設定されるであろう。1つより多くのマスタデバイスがある場合、任意の二次マスタデバイスが、任意のスレーブデバイスのようにバス64にアタッチされ、スレーブデバイスと同様にデバイスアドレスを用いて列挙されるであろう。状態が“00”から{“01”, “10”, “11”}に変化した（すなわち、スレーブデバイスがバス64にアタッチされた）場合、ATTACHMENTフィールドはまた、高に設定されるであろう。ATTACHMENTフィールドは、その瞬間に消去される、INTERFACE STATUSレジスタが読み取られるまで、設定された後に高（「1」）のままとなるであろう。任意の他の場合において、ATTACHMENTフィールドは、低のままとなるであろう。リセットイベント後のATTACHMENTフィールドのデフォルト値は、「0」である。

【0276】

WRCフィールド（reg. 0x12、B2）は、WRITE動作の完了を示すために使用され得る。このビットは、WRITE動作のアクティブ化を表し、アクティブハイである。それは、FRAME CONTROLレジスタへの書き込み後にアクティブにされる。それは、S14（確認）ビットが書き込まれた直後のWRITE動作の終了まで高のままとなるであろう。IOエラーが起こった場合（例えば、否定応答）、状態レジスタが読み取られた後まで、このビットはリセットされない。WRITE動作を行う試行は、次のフレーム中で開始するであろう。デバイス割り込みによって遅延させられない限り（すなわち、S0ビットおよびS0 DELAYの両方がアクティブにされる）、WRITE動作が続行し、そうでなければ、次のフレームまで遅延させられ、再度試行されるであろう。WRITE動作が完了した後、このビットは、S14における有効ACKの直後に設

定されるであろう。スレーブデバイスがWRITE動作を確認しない場合には、IO ERRORフィールドが設定され、WRCフィールドがリセットされないであろう。この場合、それは最初に、このレジスタを読み取った後にリセットされるであろう。代替実施形態（すなわち、図36a）では、このビットは、IO動作の完了を示すため、およびIO動作（READ/WRITE/FUNCTION）が完了したときに割り込みを有効にするために使用され得る。ビットは、アクティブハイであり得る。

【0277】

RDCフィールド（reg. 0x12、B1）は、READ動作の完了を示すために使用され得る。このビットは、READ動作のアクティブ化を表し、アクティブハイである。それは、FRAME CONTROLレジスタへの書き込み後に高に設定される。それは、S14（確認）ビットが書き込まれた直後のREAD動作の終了まで高のままとなるであろう。IOエラーが起こった場合（例えば、否定応答）、状態レジスタが読み取られた後まで、このビットはリセットされない。デバイス割り込みによって遅延させられない限り（すなわち、S0ビットおよびS0 DELAYの両方がアクティブにされる）、READ動作が続行し、そうでなければ、次のフレームまで遅延させられ、再度試行されるであろう。READ動作が完了した後、このビットは、S14における有効ACKの直後に再度リセットされるであろう。スレーブデバイスがREAD動作を確認しない場合、ERROR IOフィールドが設定され、WRCフィールドは、低にならないであろう。この場合、それは最初に、このレジスタを読み取った後にリセットされるであろう。代替実施形態では、このビットは、主要マスタデバイス以外のデバイスによって行われる、XおよびYワード内の任意のビット（例えば、COMMAND ERROR）の変化を示すために使用され得る。このビットは、アクティブハイである。いくつかの実施形態では、コマンドワード（S、X、Y）内の任意のビットが、マスタデバイス52によって許可されるものと適合性がない値に変更された場合に、フィールドCOMMAND ERRORはアクティブになるであろう。実施例として、Sワード内のビットまたはXワード中のコマンドタイプ（B15：B13）が変更される場合、これは、電気雑音、またはバス64を引き継ごうとする別のデバイスの指示である。しかしながら、別のデバイスが（REQUEST BUSコマンドを介して）バス64を制御できるように要求を行い、マスタデバイスが（RELEASE BUSコマンドを介して）これを許可した場合には、Xワードの変更が許可されるべきであり、COMMAND ERRORのアクティブ化をもたらすべきではない。

【0278】

FRAME DONEフィールド（reg. 0x12、B0）は、フレームが完了したことを示すために使用され得る。FRAME DONEフィールドは、フレームの最後のビットにおいてアクティブハイに設定される。したがって、FRAME DONEフィールドは、動作をバス64の基本タイミングと同期化するために使用され得る。実施形態では、FRAME DONEフィールドは、デバイスが変更され、いかなるデータ通信もアクティブではなく、フレームカウンタが通常は内部で作動しているときに、依然として有効である。FRAME DONEフィールドは、状態レジスタが読み取られるまで設定され続けるであろう。FRAME DONEフィールドは、PING動作が完了したかどうかを伝えるために、または、全てのデバイスが充電されていることを確実にするように、通信を開始する前に、例えば、ある数のフレームを数えるための基本タイミングのために使用されることができる。対応するIRQ MASKビットが有効にされるとき、割り込みが全フレームの終了時に生成されるであろう。

【0279】

SLAVE STATUSレジスタ（regs. 0x13から0x15）は、マスタデバイス52の内側に存在するレジスタを使用して実装され得、スレーブアドレス0-11を伴ってバス64にアタッチされるスレーブデバイスの状態を示すであろう。これらのレジスタは、各々2ビットであり、合計3バイトに対するスレーブデバイス0-11の状態に対応する12個の値を含む。いくつかの実施形態では、スレーブデバイス12-14（

10

20

30

40

50

デバイス15はブロードキャストのために保留される)を直接監視することは可能ではないが、それらは、S0ビットをアクティブにすることを通して、依然として割り込みを発行することができる。いくつかの代替実施形態では、アドレス12-14は、以下でさらに詳細に説明されるように、デバイス群をアドレス指定するために使用される。いずれにしても、状態値がPING動作中に読み取られる。任意のスレーブデバイスから読み取られる最高状態値は、対応する2ビット符号なし状態ビットと比較される。このスレーブ状態値が、これらのビット中の値より大きい場合、これらのビットは、この新しい値に更新される。SLAVE STATUSレジスタが読み取られるとき、その値は消去されないが、次のPING動作中に上書きされるであろう、すなわち、「11」の最大値が、1つのスレーブデバイスからのSLAVE STATUSレジスタから読み取られ、次のPING動作中の最大値が「10」である場合、このデバイスに対するSLAVE STATUSレジスタ値は、次のPING動作の完了時に「10」に更新されるであろう。3つ全てのSLAVE STATUSレジスタの初期値は、リセット動作後に「00000000」である。いかなるスレーブデバイスもバス64上に存在しない場合、何もアクティブにされず、負の論理信号伝達スキームが、バス64のこの例示的实施形態に使用されるため、論理ゼロに等しい、一定の高レベルが、バス64上で読み取られるであろう。

【0280】

図29bの代替的な例示的实施形態に関して、図29aについて示され、説明されるものと比較して、異なるレジスタ定義がある。具体的には、図29bの定義は、FRAME RATEフィールド、OVERSAMPLE(X)フィールド、OVERSAMPLE(Y)フィールド、MULTI MASTERフィールド、POWER CONTROLフィールド、およびCLOCK DIVフィールドを含む。他のフィールドは、図29aに示されるものに類似するが、異なるレジスタに位置し、および/またはフィールドに割り当てられた異なるビット数を有し得、概して、図29bに関して議論されない。OVERSAMPLE(X)OVERSAMPLE(Y)フィールドは、図29aのFRACTIONAL(LOW)およびFRACTIONAL(HIGH)フィールドに対応し、議論されない。図29bのMULTI MASTERフィールドは、後に議論されるように、図29aのENABLE MULTI MASTERフィールドと比較して異なる意味を有する。CLOCK DIVフィールドは、図29aのMCLKDフィールドに類似し、議論されない。

【0281】

FRAME RATEフィールド(reg.0x009、B7:B0)は、バス64のシステム周波数を定義するために使用され得る。値は、実施形態では符号なし2進数としてコードされる、200kHzの単位で定義することができる。12.2MHzおよび11.2MHzに対して符号化される値は、12.288および11.2896MHzとして扱われ得る(その倍数も同様)。

【0282】

図29bのFRAME CONTROLフィールドの例示的实施形態に基づく、WRITEおよびREAD動作の組み合わせが、図30に示されている。少なくとも1つの実施形態では、図29bに関して、FRAME CONTROLフィールド中の第1のビット(B7)は、保留中WRITE動作の状態を提供し、アクティブハイである。レジスタWRITE動作は、コントローラが、マスタデバイス52のデータレジスタ(WD15:WD0またはWD7:WD0)に、次いで、マスタデバイス52のアドレスレジスタ(ADDR6-0、および随意に8ビットWRITE動作についてはREG7-0)に書き込むことによって、開始し得る。次いで、コントローラは、IO CONTROL REGISTER(IOC)に書き込み、B7:B6="10"を設定することによって、実際のWRITE動作を開始し得る。次いで、WRITEレジスタ動作を行うことの試行が、次のフレーム中で開始し得る。デバイス割り込みによって遅延させられない限り、WRITE動作は続行し、そうでなければ、次のフレームまで遅延させられ、再度試行されるであろう。WRITE動作が完了した後、FRAME CONTROLフィールドは、エラー

状態中を除いて、Yワードの最後のビットが書き込まれた後に、即時に「00」に再度リセットされるであろう。IO ERRORがある場合、FRAME CONTROLフィールドの第1のビットは、IO ERRORビットが読み取られて消去されるまで、高のままとなり続けるであろう。これは、マスタデバイス52がIO ERRORのソースを見出すことを可能にする。WRITE CONTROLビット(reg. 0x00F、B7、すなわち、FRAME CONTROLフィールドの第1のビット)が高のとき、新しいREADまたはWRITE動作は、WRITE CONTROLビットが低に戻るまで、開始されるべきではない。

【0283】

実施形態では、同一の動作においてWRITE CONTROLおよびREAD CONTROLビットの両方をアクティブにすることは違法である。この条件は、マスタデバイス52によって無視され、この場合、ビットB7 : B6の以前の状態が維持されることができる。この場合、マスタデバイス52は、PING動作(すなわち、変化なし)であるようにこの条件を符号化することができる。

【0284】

少なくとも1つの実施形態では、図29bのFRAME CONTROLフィールドのREAD CONTROLビット(reg. 0x00F、B6)は、動作を開始するため、およびこれらの動作の進行を監視するために使用され得、その実施例は、図31に示されている。READ CONTROLビットからの読み取りは、任意の保留中動作の状態を与えるであろう。DATAおよびADDRESSレジスタが、READおよびWRITE動作に使用されるであろう。動作が完了した後、このREAD CONTROLビットが、再度リセットされるであろう。IO ERRORがある場合、READ CONTROLビットは、IO ERRORビットが読み取られて消去されるまで、高のままであり続けるであろう。これは、マスタデバイス52がIO ERRORのソースを見出すことを可能にする。他の実施形態では、動作のタイプ(すなわち、PING/FUNCTION/WRITE 8ビット/READ 8ビット/WRITE 16ビット/READ 16ビット)が、レジスタ0x00FのビットB7 - B5によって決定され得る。

【0285】

動作が進行中であるとき、新しい動作は開始されない。FRAME CONTROLフィールド = '11' であるとき、FUNCTION動作が開始される。FUNCTION DATAフィールドを利用する関数は、(FRAME CONTROLフィールドからの読み取りに関して)READ/WRITE動作のように扱われるであろう。行われるFUNCTION動作のタイプは、FUNCTIONレジスタ(0x15)によって決定されるであろう。

【0286】

DATA SIZE(reg. 0x00F、B5)フィールドは、READおよびWRITE動作に対するデータ幅を選択するために使用され得る。DATA SIZEフィールド中の論理0は、8ビットデータをもたらす、DATA SIZEフィールド中の論理1は、16ビットデータ動作をもたらす。電源投入リセットイベント後のデフォルト値は、「0」、すなわち、8ビット動作である。

【0287】

図29bのENABLE MULTIフィールド(reg. 0x15、B7)は、マルチフレーム同期化(すなわち、複数のフレームにわたって伝送されるデータを同期化するためのDS0ビットの使用を有効にするために使用され得る。ENABLE MULTIフィールドが1に設定されるとき、マルチフレーム同期化が有効にされ、CLOCK DIVIDERフィールドは、DS0ビットが1(通常は0)に設定され得る、フレーム数中で距離を選択するために使用され得る。これは、スレーブデバイス54によって、いくつかのデータ動作モードで、より効率的なデータ転送に使用されることができる。ENABLE MULTIフィールドが1に設定されるとき、CLOCK DIVIDERフィールドは更新されないであろう。ENABLE MULTIフィールドを0から1に

10

20

30

40

50

変更した後、後に続く第1のフレームは、1に設定されたDS0ビットを有するであろう。DS0ビットのアクティブ化は、SET OVERSAMPLE RATIO関数を使用して、オーバーサンプル率をデータレジスタに書き込むことによって行われる。実施例として、レジスタ0x0000が0x02に設定され、レジスタ0x0001が0x01に設定される（これらのレジスタについては図16bを参照）場合には、DS0ビットは、1つおきのフレーム中で1に等しく設定されるであろう。

【0288】

図29bのPOWER CONTROLフィールド(reg.0x15、B5)は、単一ワイヤ実施形態に対するバス64の動作を制御するために使用され得る。POWER CONTROLフィールドが0に設定される場合には、データがバス64上で伝送される 10
ときに、スレーブデバイスの充電がない。POWER CONTROLフィールドが1に設定される場合には、バス上にデータ転送がないときに、スレーブデバイスの充電がある。POWER CONTROLフィールドは、この情報がI²Cコマンドから受信される時間にかかわらず、フレームの最後のビットにおいて内部で更新されることができる。

【0289】

図9a、16a、22a、22b、25a、25b、29a、29c-29i、32a、33a、34a、および36aに示される例示的实施形態は、第1の代替的な例示的实施形態と一緒に使用されることができる一方で、図9b、16b、17、22c、25c、29b、30、31、32b、33b、34b、および36bに示される例示的实施形態は、第2の代替的な例示的实施形態と一緒に使用されることができることに留意されたい。この段落で記述されない、他の図に示される例示的实施形態は、第1および第2の代替的な例示的实施形態の両方とともに使用されることができる。 20

【0290】

少なくとも1つの実施形態では、最大で12個のスレーブデバイス(0-11)が、状態データで応答し得る。マスタデバイス12によるSLAVE STATUSレジスタへの任意の書き込みは、無視される(アクションなし)。SLAVE STATUSレジスタの例示的定義が、図37に示されているが、他の定義および別の数のスレーブデバイスを他の実施形態で使用されることができる。

【0291】

少なくとも1つの実施形態では、部品ID中の製造業者符号化シーケンスは、バス64 30
に最初にアタッチするスレーブデバイスへのアドレスの動的アドレス割付で使用される。これは、ここで説明されるように、圧縮符号化形式、または一般符号化形式を使用することによって為されることができる。

【0292】

例示的实施形態では、圧縮符号化形式は、部品IDを、スレーブデバイス54のデバイスIDとして使用されることができる32ビットに符号化する。形式は、図38aに示されるように、位置符号化、製造業者名、および部品番号を含む。この例示的实施形態では、位置符号化は、最大で15の値(1-15)を許容し、製造業者名は、2文字として符号化され、部品番号は、2進数で符号化され、10進5桁を許容する(位置符号化は本説明において以降で説明される)。製造業者名で使用される文字は、各文字にASCII文字コードの5つの最下位ビット、すなわち、A="000001"、B="000010"等を使用する。製造業者名に対する2文字は、固有の2文字の頭字語をもたらすように選択される。マスタデバイス52は、圧縮符号化形式を使用するとき、4バイトの情報を読み取ることができる。16ビットREAD動作がマスタデバイス52によって使用される場合、2バイトが全READ動作中に返されるであろう。最上位バイトが、第1のバイトを表す一方で、最下位バイトは、第2のバイトを表す。 40

【0293】

圧縮符号化形式の実施例が以下に続く。製造業者XMCOがXM007000と名付けられた部品をコードしたいと仮定する。最初の2つの文字は、製造業者コード「XM」であり、部品番号は、「07000」であり、位置情報はない。圧縮符号化は、B31=' 50

0'を使用する。位置情報がないため、これは、B30:B27="0000"をもたらす。文字「X」は、88のASCIIコードを有し、文字「M」は、77のASCIIコードを有する。ASCIIコードは、元の値から64を差し引くことによって、8ビットから5ビットまで圧縮することができる。したがって、ビットB26:B17は、{'X'-64, 'M'-64}={24, 13}={"11000", "01101"}に等しい。代替として、値は、5ビットを得るように0x1Fで論理的にAND演算される、ASCIIコードから見出されることができる。次いで、部品番号は、2進数に変換され、すなわち、"07000"="0x1B58"=0.0001.1011.0101.1000である。したがって、圧縮符号化形式での32ビットIDは、{B31, B30:B27, B26:B17, B16:B0}="0000.0110.0001.1010.0001.1011.0101.1000"}=0x061A.1B58である。

10

【0294】

例示の実施形態では、一般符号化形式は、スレーブデバイス54のデバイスIDを符号化するように、一連のASCII文字として部品番号を定義し、その実施例は、図38bに示されている。部品番号は、NULL文字(すなわち、ゼロに等しいバイト)によって終端されるASCII文字として表すことができる、任意の名称を有し得る。第1のバイトは、形式(すなわち、短い符号化または一般符号化)、部品等級、および随意的な位置情報を符号化するために使用される。第2のバイトは、バスのバージョンおよびチップのバージョンを示すために使用される。第2のバイトに続く文字は、ASCII値を使用した部品名として解釈される。部品等級は、異なる量のタイプの特定の部品を区別するために使用されることができる。バスバージョンは、バージョン1.00と同等のバージョン「0001」で始まり得る。チップバージョンは、バージョン「0000」で始まり、後続のバージョンに対して増分し得る。チップバージョン番号の増分は、単一ステップである必要はない。例えば、第1のバージョンは、「0000」であり得、「0000」、「00001」、「0010」等を数える代わりに、その後、「0001」、「0011」、「0111」という以降のバージョンが続き、最終的に、最後のバージョンに「1111」が使用される。このスキームは、実用的な理由で(メタルマスク変更にとってより容易である)使用されることができるが、より大きい数は、概して、より新しいバージョンに対応する。16ビットREAD動作がマスタデバイス52によって使用される場合には、2バイトが全READ動作中に返されるであろう。8つの最上位ビットは、第1のバイトを表し、8つの最下位ビットは、第2のバイトを表す。文字列が奇数のバイトを有するとき、それは、16ビットデータ読み取りに適合しないであろう。したがって、単一のバイトが文字列を終端すると予期され、ワードが読み取られる場合、最上位バイトが使用され、例えば、シーケンスがワード境界の中央で終端されるとき、最上位バイトはゼロである。シーケンスが早期に終端されないことを確実にするために、このモードでは、ID中にゼロに等しいバイトがない。また、部品を符号化するとき、実際の製造業者以外の製造業者名は使用されない。

20

30

【0295】

一般符号化形式の実施例が以下に続く。製造業者ADIが、部品名「ADI-SSM2377ACBZ-RL」を伴うチップを符号化したい。チップは、位置情報を有さず、「A」という部品等級を有する。バスバージョンは、第1の公式バージョン1.00="0001"である。チップのバージョンは、第1のバージョン="0000"である。一般符号化は、ASCIIで符号化される部品番号が後に続く、16ビット圧縮情報フィールドによって行われる。符号化は、以下のように行われ、すなわち、B7='1'(一般符号化)、B6:B3="0000"(位置情報なし)、B2:B0="000"(部品等級A)であり、すなわち、読み取られる第1のバイトは0x80である。第2のバイトは、チップバージョンが後に続くバスバージョンであり、すなわち、"0001"および"0000"=0x10である。アドレスゼロからの複数の読み取りを使用して読み取られる、完全部品名は(実際の文字が示され、返された値はこれらのASCII符号化である)、{'A', 'D', 'I', '-', 'S', 'S', 'M', '2', '3', '7', '7', 'A', 'C', 'B', 'Z', '-', 'R', 'L'}

40

50

7', '7', 'A', 'C', 'B', 'Z', '-', 'R', 'L'}である。読み取られる最後のバイトは、NULL: B7: B0 = "00000000"である。このデバイスが最高アービトレーション優先度を有すると仮定して、アドレス0から読み取り、デバイス0は、以下の値のシーケンス、すなわち、{0x80, 0x10, 0x41, 0x44, 0x49, 0x2D, 0x53, 0x53, 0x4D, 0x32, 0x33, 0x37, 0x37, 0x41, 0x43, 0x42, 0x5A, 0x2D, 0x52, 0x4C, 0x00}をもたらすであろう。

【0296】

時として、これらのデバイスが同一の部品情報を含み得るため、それらの位置的配置に基づいて、いくつかの同一の部品を区別する必要がある。この場合、固有のスレーブアドレスをこれらのデバイスに割り当てることができるように、それらを区別するために、追加のフィールドが使用される。位置フィールドが、この目的で使用される。実施形態では、位置フィールドは、16の異なる値を有することができる。値「0000」は、位置情報が利用可能でない場合、または必要ではない場合のために保留される。残りの15の値は、位置を示すために使用される。

10

【0297】

デバイスの番号付けは、種々の方法で特定されることができる。例えば、ハンドヘルドユニット中に6つのマイクロホン（以下のように定義される）がある場合、主要マイクロホンには、位置1を割り当てることができ、左マイクロホンには、位置2を割り当てることができ、右マイクロホンには、位置3を割り当てることができ、後方の雑音マイクロホンには、位置4を割り当てることができ、第2の雑音マイクロホンには、位置5を割り当てることができ、ラウドスピーカの隣のマイクロホンには、位置6を割り当てることができる。

20

【0298】

別の実施例として、ヘッドセットの内側で複数のマイクロホンを使用するとき、以下の番号付けスキームが使用され得る。左主要マイクロホンは、位置1を割り当てられ得、右マイクロホンは、位置2を割り当てられ得、左耳外部マイクロホンは、位置3を割り当てられ得、右耳外部マイクロホンは、位置4を割り当てられ得、左耳内部マイクロホンは、位置5を割り当てられ得、右耳内部マイクロホンは、位置6を割り当てられ得、追加の左マイクロホンは、位置7を割り当てられ得る。

30

【0299】

別の実施例として、DOLBYTMサラウンド音響マルチスピーカシステムについては、以下の番号付けスキームが使用され得る。左前スピーカは、位置1を割り当てられ得、右前スピーカは、位置2を割り当てられ得、中央スピーカは、位置3を割り当てられ得、左後スピーカは、位置4を割り当てられ得、右後スピーカは、位置5を割り当てられ得、サブウーファスピーカは、位置6を割り当てられ得、左脇スピーカは、位置7を割り当てられ得、右脇スピーカは、位置8を割り当てられ得る。

【0300】

少なくとも1つの実施形態では、一般的な位置割当は、要素の場所に基づいて、複数の要素に行われることができる。例えば、最初に、要素を左から右に（左が最初である）ソートすることができ、次いで、要素を前から後に（前が最初である）ソートすることができ、次いで、要素を上から下に（上が最初である）ソートすることができ、次いで、要素を最大から最小（最大が最初である）にソートすることができる。

40

【0301】

前述のように、統一バス通信プロトコルは、動作モードに基づいて、異なるフレーム形式を有する。動作のワードモードでは、ワードフレーム形式が使用される。動作のビットストリームモードでは、ビットストリームフレーム形式または統一フレーム形式が使用される。ワード、ビットストリーム、および統一フレーム形式の実施例が続く。

【0302】

ここで図39を参照すると、単一の時間フレームに対して、ワードモードで使用される

50

一般的なワードフレームが示されている。マスタデバイス52とスレーブデバイス54との間の通信は、Sワードで始まり、その後にXワード、次いで、Yワードが続く、フレーム中に起こる。Sワードとコマンドワードとの間には、使用シナリオに応じて、データ伝送に割り付けられた空の空間があり得る。空の空間の長さは、COMMAND SEPARATIONレジスタによって定義される。

【0303】

マスタデバイス52によるSワードの伝送は、バス64に物理的に接続されるスレーブデバイスが、マスタデバイス52と通信するためにロックされる(すなわち、同期化される)ことを可能にする(同期化する方法は、本説明において以降でさらに詳細に説明される)。Sワードの伝送はまた、注意を必要とする任意の割り込みが設定されているかどうかをマスタデバイス52が決定することも可能にする。

10

【0304】

Sワードの伝送とXワードの伝送との間に、スレーブデバイス54またはマスタデバイス52のうちの少なくとも1つによってピックアップされることができる、ランダムデータまたは情報が、バス64を経由して伝送され得る。この情報は、ビットストリーム、オーディオデータ、または例えば、センサデータ等の他の数値データを含むが、それらに限定されない。Xワードの伝送は、マスタデバイス52およびスレーブデバイス54が、任意の特定の機能が実行されるべき場合を決定すること、またはスレーブデバイスのうちのいくつかの状態を決定すること、または動作のためのアドレスの一部分を特定することを可能にする。(いかなる割り込みも設定されなかったと仮定して)Xワードの伝送後に、前述のように、さらなるデータが、バス64を経由して伝送され得る。次いで、動作に応じて、スレーブデバイスのうちの少なくともいくつかに対する動作または状態情報のために、さらなるアドレス情報またはデータ情報を提供する、Yワードが伝送される。

20

【0305】

ここで図40を参照すると、単一の時間フレームに対して、ワードフレーム形式の例示的实施形態が示されている。本実施例では、制御ワードが次々に続き、フレーム長が $3 * (4 * \text{コマンド分離値} + 16) = 3 * (4 * 0 + 16) = 48$ ビットであるため、コマンド分離は、最小分離値である、0ビットである。図40の表の最上行および第1の列によって定義されるビット番号は、どのタイムスロット中で、ビットがバス64上で伝送されるかを示し、例えば、タイムスロット0x02Eは、ビットY1を含む。このフレーム形式は、制御データのみが転送されるか、または制御データに対する非常に高い帯域幅が短い時間量にわたって使用される、状況に有用である。CURRENT BANKレジスタビットは、必要であれば、このデータモードへ、およびそこから変化するために使用されることができる。

30

【0306】

ここで図41を参照すると、ワードモードで単一の時間フレームに使用されるワードフレーム形式の別の例示的实施形態が示されている。本実施例では、コマンド分離は、 $4 * 28 = 112$ ビットであり、フレーム長は、 $3 * (4 * 28 + 16) = 3 * (128) = 384$ ビットである。本実施例は、5つのオーディオデータチャンネル(A、B、C、D、およびE)、および16ビットを有する各チャンネルを伴う2つの未使用データチャンネルを示す。本実施例は、6.144MHzバスクロックおよび48kHzサンプル周波数と良好に適合する。値が「Z」(すなわち、3状態)である場合、マスタデバイス52は、「0-1-Z-Z」パターンを駆動するであろう。値が「0」である場合、マスタデバイス52は、「0-1-0-0」パターンを駆動するであろう。値が「1」である場合、マスタデバイス52は、「0-1-Z-Z」パターンを駆動するであろう。値がスレーブデバイス54によって決定される場合、マスタデバイス52が、「0-1-Z-Z」パターンを駆動するであろう一方で、スレーブデバイス54は、スレーブデバイス54が伝送しているビットの値に応じて、単一ワイヤ事例に対して「Z-Z-U-0」パターンまたは「Z-Z-Z-Z」パターンのいずれかを駆動するであろう。示されたデータ形式は、12.288MHz導出バスクロックと良好に適合し、5つのアクティブなオーディオソース

40

50

が示されている。しかしながら、このデータ形式が最大で7つのアクティブなソースを同時に可能にするため、フレーム中に2つの未使用チャンネルまたは空のチャンネルがある。

【0307】

ここで図42を参照すると、ビットストリームモードでの単一のフレームのためのビットストリームフレーム形式の例示的实施形態が示されている。本実施例では、ビットストリームフレーム形式は、4つのビットストリームフレームチャンネル、すなわち、制御チャンネル(フレームチャンネル0)および3つのビットストリームデータチャンネル(フレームチャンネル1から3)を備える。ビットストリームフレームチャンネルのセットは、各行のビット0、4、8、およびCから開始し、各ビットストリームフレームチャンネルに対して1度に1ビット送信される。制御チャンネルは、Sワードからのビット、その後Xワードからのビット、その後Yワードからのビットを送信するために使用され。ビットストリームデータフレームチャンネルは、異なるビットストリームデータチャンネルからビットを送信するために使用される。本実施例では、1ビットが制御フレームチャンネルから送信され、その後ビットストリームデータフレームチャンネル1からの1ビットが続き、その後ビットストリームデータフレームチャンネル2からの1ビットが続き、その後ビットストリームデータフレームチャンネル3からの1ビットが続く。したがって、コマンド分離値は、3ビットに等しい(すなわち、第4ビットごとに、伝送されるSワード、Xワード、またはYワードのいずれかからのコマンドビットがある)。フレーム長は、 $4 * 48 = 192$ ビットである。

【0308】

図42に示されるフレームのフレーム構造は、実際には、48行および3列である。しかしながら、フレームは、ビットがバス64上で伝送されるタイムスロットに対応する、ビット番号に従って示される。このフレームに対するタイムスロットは、最上行の左隅から開始し、最上行の端まで、次いで、次の行の左側からその行の端まで等、継続することを理解されたい。

【0309】

データが図42に示されるフレーム形式で構造化される、典型的なシナリオは、デジタル入力およびデジタルマイクロホンに伴う2つのクラスD増幅器があり、増幅器およびマイクロホンが各々、ビットストリームデータを生成するときである。別のシナリオは、ビットストリームデータを生成する、3つのデジタルマイクロホンである。この場合、3つのデジタルマイクロホンは、 3.072 MHzでサンプリングされることができ、バスクロック周波数は、例えば、 6.144 MHzであり得る。したがって、このフレーム形式は、伝送機および受信機の両方に対する複数のビットストリームデータチャンネルにインターレースするために好適である。バス64を経由して伝送されるビットストリームデータチャンネルの各々の間に1ビットの待ち時間のみがあるため、このビットストリームフレーム形式は、処理においてわずかな遅延を可能にする。また、任意の値によるデシメーションが、システムによって以降で使用されることができ、ビットストリームを生成するために使用されるシグマデルタ変換器のオーバーサンプリング率は、このスキームを使用する特定の値に制限されない。実践では、 12.288 、 19.20 、および 26.00 MHzクロックシステムをサポートするために、有限数のオーバーサンプリング率が使用される。

【0310】

ここで図43を参照すると、ビットストリームモードで使用することができる、単一の時間フレームに対する統一フレーム形式の例示的实施形態が示されている。本実施例では、統一フレーム形式は、制御チャンネル(フレームチャンネル0)、5つのビットストリームデータチャンネル(フレームチャンネル1から5)、および2つの仮想フレームチャンネル(フレームチャンネル6および7)を備える。統一フレーム形式は、少なくとも1つの仮想フレームチャンネルを使用する、フレーム形式である。仮想フレームチャンネルは、デジタルワードデータから変換されている(すなわち、入力データの場合)、またはデジタルワードデータに変換されるであろう(すなわち、出力データの場合)、データを伝送するために使

用される。デジタルワードデータは、対応する仮想フレームチャネルによって指定されるタイムスロットにおいてデジタルワードを1度に1ビット伝送することによって、ビットストリームに変換される。

【0311】

本実施例では、フレームチャネルは、各行のビット0および8から開始し、各フレームチャネルから1度に1ビット送信される。したがって、コマンド分離値は、7ビットに等しい(すなわち、8ビットごとにSワード、Xワード、またはYワードからのコマンドビットがある)。フレーム長は、 $8 * 48 = 384$ ビットである。このフレームのフレーム構造は、実際には、24行および8列である。しかしながら、フレームは、以前に説明されたように、ビット数に従って示される。

10

【0312】

データが図43に示される形式で構造化される、典型的なシナリオは、ビットストリームデータを生成する、5つのデジタルマイクロホンと、ビットストリームデータを受信するか、または仮想フレームチャネルによって提供されるビットストリームデータからデジタルワードデータを再構築することができるかのいずれかである、2つのスピーカとがあるときである。本実施例では、5つのデジタルマイクロホンチャネル(フレームチャネルM0からM4)は、768kHz(48のデシメーション率を伴う16kHzサンプリング周波数と同等である)であり得、2つの仮想フレームチャネル(フレームチャネルAおよびB)は各々、16ビットバイナリワードを伴って48kHzであり得る、2つのオーディオチャネルである。バスクロック周波数は、6.144MHzであり得る。

20

【0313】

ここで図44を参照すると、ビットストリームモードで使用するための単一の時間フレームに対する統一フレーム形式の別の例示的实施形態が示されている。本実施例では、統一フレーム形式は、制御チャネル(フレームチャネル0)と、5つのビットストリームデータチャネル(フレームチャネル1から5)と、2つの仮想フレームチャネル(フレームチャネル6および7)とを備える。しかしながら、本実施例では、各仮想フレームチャネルは、2つのワードデータチャネルに対するデータを有する。したがって、本実施例では、2つの仮想フレームチャネルは、4つのワードデータチャネルからのデータを搬送するために使用される。例えば、フレームチャネル6は、ワードデータチャネルCおよびワードデータチャネルAに対応する、ビットストリームデータを有する。フレームチャネル6は、ワードデータチャネルCを使用するデバイスへのデータの受信または伝送、およびワードデータチャネルAを使用するデバイスへのデータの受信または送信を行うために使用される。タイムスロットは、最初にワードデータチャネルCのために、次いで、ワードデータチャネルAのために使用される。加えて、フレームチャネル7は、タイムスロットが、最初にワードデータチャネルD、次いで、ワードデータチャネルBに使用される、ワードデータチャネルDおよびワードデータチャネルBのために対応するビットストリームデータを有する。この場合、仮想フレームチャネル中のデータは、バス64上で送信される際に、デジタルワードデータのビットストリームバージョンである。データ同期化ビット(DS0)は、データチャネルに対するデータが開始および停止する(例えば、AまたはC)ときについての曖昧性を回避するように、いくつかのフレームにわたってワードデータに対応するビットストリームデータを同期化するために使用され得る。空間を節約するために、DS0およびDS1ビットは、PINGフレームの中央(すなわち、フレームが開始する直前ではなく)で転送される。これらのビットは、次いで、その後の全サブフレーム中の所与の段階によって増分される、既知の値(例えば、ゼロ)に分数カウンタをリセットするために使用される。次いで、オーバーフロービットは、それがソース(すなわち、データチャネル)AまたはソースCからのデータであるかどうかを選択するために使用される。フレームは、Sワードの後の第1のビットから始まり、その時、対応する1つ以上のスレーブデバイスが、レジスタバンク選択およびデータ同期化について更新されている。ワードデータチャネルを使用する種々のデバイスにおいて、デジタルワードデータが、バス64上での伝送のためにビットストリームデータと多重化されるか、または統一

30

40

50

ビットストリームバージョンから、それがデバイスによって使用されるワードデータに逆多重化される。実施例として、システムは、ビットストリームデータを受信する2つのデジタル増幅器、およびBluetooth（登録商標）またはデジタルFM受信機あるいは伝送機への接続を含むことができる。第1のスレーブデバイス（すなわち、増幅器）が、ビットストリームデータまたは多重化TDMデータを使用するであろう一方で、他のデバイスは、多重化TDMデータを使用するであろう。

【0314】

本実施例では、フレームチャネルは、各行のビット0および8から始まり、各フレームチャネルから1度に1ビット送信される。したがって、コマンド分離値は、7ビットに等しい（すなわち、8ビットごとにSワード、Xワード、またはYワードのいずれかからのコマンドビットがある）。フレーム長は、 $8 * 48 = 384$ ビットである。このフレームのフレーム構造は、実際には、24行および8列である。しかしながら、フレームは、前述のようにビット番号に従って示される。データが図44に示される形式で構造化される、典型的なシナリオは、デジタルワードデータを生成または使用するデバイスのために、ビットストリームデータを生成する5つのデジタルマイクロホンと、4つの補助オーディオチャネルとがあるときである。例えば、5つのデジタルマイクロホンチャネル（フレームチャネルM0からM4）は、1.536MHzのデータレート（64のデシメーション率を伴う24kHzサンプリング周波数と同等である）を有することができ、4つのデジタルワードチャネル（A、B、C、およびD）は各々、16ビットのワード長を伴って48kHzでサンプリングされる、単一のオーディオチャネルを表す。バスクロック周波数は、12.288MHzであり得る。

【0315】

ここで図45aから48cを参照すると、統一バス通信プロトコルに対するバス周波数、チャネルの数およびタイプの有用な組み合わせの実施例が示されている。これらの図中の表は、バス64に対する参照クロック信号として使用される、外部クロック周波数に従ってソートされている。奇数分割比を伴う構成を、2ワイヤバス実施形態で、または記載される周波数の2倍である外部クロック信号とともに、使用することができる。利用可能な制御帯域幅は、高速モードI²Cバスとほぼ同一である。16ビットデータを使用する400kHz I²Cバスは、毎秒12.5kHz動作に対応する。

【0316】

ここで図45aおよび45bを参照すると、ワードモードで動作するときの統一バス通信プロトコルの実施形態に対するバス周波数、チャネルの数およびタイプの例示的組み合わせの表が示されている。これらの表は、図45aの表の第6の行中の実施例を見ることによって理解することができる。この場合、サブフレーム長が $5 * 24 * 2 + 16 = 256$ ビットであり、16ビットの追加がサブフレーム中の制御ワードのためのものであるように、各々24ビット長であり、2回繰り返される、5つのオーディオチャネルがある。反復は、2つのデジタルワードが、各サブフレーム中のチャネルの各々から得られることを意味する。反復は、データチャネルA、B、C、D、およびEからのデータに対するA - A - B - B - C - C - D - D - E - Eに従って行われることができるが、低遅延のためには、反復A - B - C - D - E - A - B - C - D - Eが使用され得る。サブフレーム中の制御データに使用される帯域幅は、サブフレームにおいて16制御ビットがあるため、 $16 / 256 = 6.3\%$ である。図45aの行3に示される実施例は、S/PDIF適合性に非常に適している。

【0317】

図45aに関して、行1中の実施例は、ワードモードでサブフレーム中で3回繰り返される、1つのオーディオチャネルを有する。行2中の実施例は、ワードモードでサブフレーム中で1回繰り返される1つのオーディオチャネルを有する。行3中の実施例は、ワードモードでサブフレーム中で2回繰り返される、2つのオーディオチャネルを有する。行4中の実施例は、ワードモードでサブフレーム中で1回繰り返される、3つのオーディオチャネルを有する。行5中の実施例は、ワードモードでサブフレーム中で4回繰り返され

10

20

30

40

50

る、3つのオーディオチャンネルを有する。行6中の実施例は、ワードモードでサブフレーム中で2回繰り返される、5つのオーディオチャンネルを有する。行7中の実施例は、ワードモードでサブフレーム中で2回繰り返される、6つのオーディオチャンネルを有する。行8中の実施例は、ワードモードでサブフレーム中で1回繰り返される、12個のオーディオチャンネルを有する。これらの実施形態では、制御チャンネルの帯域幅は、データに対するフレーム中に割り付けられるフレームチャンネルの数、およびフレームチャンネルの幅に応じて、減少させられることができる。

【0318】

図45bに関して、行1中の実施例は、ワードモードでサブフレーム中で4回繰り返される、4つのオーディオチャンネルを有する。行2中の実施例は、ワードモードでサブフレーム中で4回繰り返される、6つのオーディオチャンネルを有する。行3中の実施例は、ワードモードでサブフレーム中で2回繰り返される、8つのオーディオチャンネルを有する。行4中の実施例は、ワードモードでサブフレーム中で2回繰り返される、12個のオーディオチャンネルを有する。これらの実施例では、制御チャンネルの帯域幅は、12個のオーディオチャンネルの各々の帯域幅の半分である。

【0319】

ここで図46aおよび46bを参照すると、ビットストリームモードで動作し、ビットストリームフレーム形式を使用するときの統一バス通信プロトコルの実施形態に対するバス周波数、オーバーサンプリングレート、チャンネルの数およびタイプの例示的組み合わせの表が示されている。これらの組み合わせは、例えば、いくつかのデジタルマイクロホンの組み合わせに使用されることができる。また、インターリーブされたビットストリームチャンネルを使用して、ビットストリームチャンネルの帯域幅を拡張することも可能であり、例えば、48kHzサンプル周波数は、以下に示されるように、同一のクロッキング要件に対して、各々が16kHzで3つのビットストリームチャンネルを使用する。したがって、インターリーブされたビットストリームチャンネルを用いて、(図中の列として示されるような)1つのビットストリームチャンネルは、1つより多くのデバイスに関連付けられるビットストリームデータを有することができる。この文脈では、「に関連付けられる」という語句は、データがポートによって生成されるか、またはデバイスのポートによって使用され、これを、1つのデバイスのための複数のポート、または1つより多くのデバイスのための複数のポートへ拡張できることを意味する。図28cおよび28dは、複数のソース/ポートの間でビットストリームフレームチャンネル(すなわち、列)を共有する実施例を示す。

【0320】

図46aおよび46bの表は、図46aの表の第3の行中の実施例を見ることによって理解することができる。この場合、これらのフレームチャンネルのうちの3つがオーディオデータに使用され、フレームチャンネルのうちの1つが制御チャンネルとして使用される、ビットストリームフレーム形式で合計4つのビットストリームフレームチャンネルがある。したがって、制御データに使用される帯域幅は、 $1/4 = 25\%$ である。実際のサンプルレートは、図46aおよび46bに示される実施例に対して変化する。実際のサンプルレートは、出力サンプル周波数をオーバーサンプリングレートで乗算することから求めることができ、すなわち、図46aの第1の行に対する実際のサンプルレートは、 $16 * 48 = 768 \text{ kHz}$ であろう。シグマデルタ変換器からの出力ビットストリームが、このサンプルレート(768kHz)である一方で、出力サンプルレートは、どのようなデシメーションフィルタが使用されているかに応じて、16kHzとは異なり得る。例えば、デシメーションフィルタが48倍デシメートする場合には、出力サンプル周波数は、第1の行に示されるように16kHzであろう。フレーム制御チャンネルに対する制御データも16kHzでサンプリングされるが、これらのサンプリングレートが異なる実施例もある。図46aおよび46bの実施例の全ては、ビットストリームフレーム形式を利用し、チャンネルは、サブフレーム中で1回繰り返されるのみである。ビットストリームモードでは、サブフレームは、制御ビットから始まり、次の制御ビットの前まで続くビットを含むビットを

10

20

30

40

50

有する、フレームの部分として定義される。200/3のオーバーサンプリング率を伴う図46bの行2に示される実施例は、(オーバーサンプリング率が64に近い)19.20MHzシステム周波数を伴うデジタルマイクロホンに非常に適している。

【0321】

図46aに関して、行1および2中の実施例は、フレームチャンネルのうち的一方がビットストリームデータチャンネルに使用され、他方のチャンネルが制御チャンネルに使用される、2つのビットストリームフレームチャンネルを有する。行3および4中の実施例は、フレームチャンネルのうち3つがビットストリームデータチャンネルに使用され、他のチャンネルが制御チャンネルに使用される、4つのビットストリームフレームチャンネルを有する。行5中の実施例は、フレームチャンネルのうち5つがビットストリームデータチャンネルに使用され、他のフレームチャンネルが制御チャンネルに使用される、6つのビットストリームフレームチャンネルを有する。行6中の実施例は、フレームチャンネルのうち7つがビットストリームデータチャンネルに使用され、他のフレームチャンネルが制御チャンネルに使用される、8つのビットストリームフレームチャンネルを有する。行7中の実施例は、フレームチャンネルのうち11個がビットストリームデータチャンネルに使用され、他のフレームチャンネルが制御チャンネルに使用される、12個のビットストリームフレームチャンネルを有する。行8中の実施例は、フレームチャンネルのうち15個がビットストリームデータチャンネルに使用され、他のフレームチャンネルが制御チャンネルに使用される、16個のビットストリームフレームチャンネルを有する。制御チャンネルに対する帯域幅は、図46aの帯域幅列に示されるようなビットストリームフレームチャンネルの数に応じて変化する。

【0322】

図46bに関して、行1中の実施例は、フレームチャンネルのうち一方がビットストリームデータチャンネルに使用され、他方のフレームチャンネルが制御チャンネルに使用される、2つのビットストリームフレームチャンネルを有する。行2中の実施例は、フレームチャンネルのうち2つがビットストリームデータチャンネルに使用され、他のフレームチャンネルが制御チャンネルに使用される、3つのビットストリームフレームチャンネルを有する。行3中の実施例は、フレームチャンネルのうち3つがビットストリームデータチャンネルに使用され、他のフレームチャンネルが制御チャンネルに使用される、4つのビットストリームフレームチャンネルを有する。行4および5中の実施例は、フレームチャンネルのうち5つがビットストリームデータチャンネルに使用され、他のフレームチャンネルが制御チャンネルに使用される、6つのビットストリームフレームチャンネルを有する。行6中の実施例は、フレームチャンネルのうち11個がビットストリームデータチャンネルに使用され、他のフレームチャンネルが制御チャンネルに使用される、12個のビットストリームフレームチャンネルを有する。制御チャンネルに対する帯域幅は、図46bの帯域幅列に示されるようなビットストリームフレームチャンネルの数に応じて変化する。

【0323】

ここで図47を参照すると、ハイブリッドワードモードで動作するときの統一バス通信プロトコルの実施形態に対するバス周波数、ならびにチャンネルの数およびタイプの例示的組み合わせの表が示されている。この表は、異なるサンプルレートおよび異なる帯域幅の組み合わせを示す(すなわち、異なるソースからのデータが異なる帯域幅を用いて伝送される)。ハイブリッドワードモードは、異なるサンプル周波数が使用される状況(例えば、Dolby 5.1形式、または音声通話および音楽がミキシングに使用されるシナリオ、例えば、同時に8および48kHz)、または異なるワード長が同時に利用可能である場合に有用である。REPEATパラメータが1未満であるとき、データ同期化ビット(DS0およびDS1)が使用される。図47の行1中の実施例は、320ビットのサブフレーム長を伴うDolby 5.1オーディオ転送に好適である。それは、24kHz帯域幅と同等である48kHzのサンプル周波数を伴う5つのオーディオチャンネル(典型的には、左、右、中央、左後、および右後)と、1.6kHz帯域幅と同等である3.2kHzでサンプリングされる1つのチャンネル(典型的にはサブウーファ)とを示す。5の反復値は、7としてコードされるであろう(すなわち、DS0同期化を使用したフレーム

10

20

30

40

50

の終了までの連続反復)。これは、各々24ビット分解能を用いて、48kHzにおいてTDMワードを伴う5つのチャンネルがあり、各チャンネルは、640ビットの単一の長いサブフレーム中で5回繰り返されることを意味する。単一のチャンネルは、24ビット分解能、および9.6kHzサンプル周波数を有するであろう。フレーム周波数は、9.6kHz/3、すなわち、3.2kHzである。行2中の実施例は、全てのチャンネルが16ビット分解能を有する、48kHzでサンプリングされる2つのチャンネル、および8kHzでサンプリングされる2つのチャンネルがあることを意味する。行3中の実施例は、全てのチャンネルが16ビット分解能を有する、48kHzでサンプリングされる2つのチャンネル、および8kHzでサンプリングされる3つのチャンネルがあることを意味する。制御チャンネルに割り付けられる帯域幅を変化させることによって、同一のクロック周波数を使用するときに、より多くのオーディオ/データチャンネルを取得できることが分かる。

10

【0324】

ここで図48a、48b、および48cを参照すると、ビットストリームモードで動作し、統一ビットストリームフレーム形式を使用するときの統一バス通信プロトコルの実施形態に対するバス周波数、チャンネルの数およびタイプの例示的組み合わせの表が示されている。

【0325】

図48aに関して、この組み合わせは、ビットストリームデータチャンネルを使用するポートからの入力データ、およびワードデータチャンネルを使用するポートに提供される出力データがあるときに有用である。例えば、図48aの形式は、いくつかのマイクロホン(すなわち、ビットストリームデータチャンネル)からの入力、および2つ以上のスピーカへのデジタルワードデータ(すなわち、ワードデータチャンネル)での出力があるときに使用されることができる。この場合、ビットストリームチャンネルのうちのいくつかは、統一フレーム形式に従ってデジタルワードデータをビットストリームデータとして伝送するために使用される。本実施例は、10個のビットストリームチャンネル(すなわち、10列)へのデータのグループ化を示す。第1の列は、コマンドワードによって占められる制御フレームチャンネルである。次の5列は、各々1.2MHzでサンプリングされ、サンプリング係数が50である場合に24kHzの期待出力サンプル周波数を伴う、5つのオーディオビットストリームによって占められるビットストリームフレームチャンネルである。最後の4つの列は、各々48kHzで動作し、20ビット分解能を伴う、5つのTDMデジタルワードストリームを搬送することが予期される、仮想ビットストリームフレームチャンネルである。5つのTDMチャンネルは、インターレースされる4つの仮想フレームチャンネルを経由して転送される(すなわち、最初に5つ全てのMSBが送信され、次いで、5つ全てのMSB-1が送信される等)。DS0またはDS1ビットは、同期化に使用されることができる。これは、ビットストリームモードで異なる解釈を有する、SUBGROUPおよびREPEATフィールドをプログラムすることによって行われることができる。REPEATフィールドは、TDM転送に割り付けられたビットストリームの数(すなわち、011として符号化される、本実施例では4)を示し、SUBGROUPフィールドは、(0-7の番号付けスキームが使用されるときに)使用される第1のビットストリームを示すであろう。通常、TDMデータチャンネルは、1つのビットストリームチャンネルの内側で転送される。しかしながら、少なくとも1つの実施形態では、TDMワードを伝送するために使用されるフレームチャンネルの数がTDMデータチャンネルの数より少ない(例えば、VSPACING定義をTDMワードのために変更することができる)、ビットストリームモードでのTDMワードの使用を取り扱うように、特殊な場合が追加されることができる。

20

30

40

【0326】

図48bに関して、行1中の実施例は、フレームチャンネルのうちの1つがビットストリームデータチャンネルに使用され、フレームチャンネルのうちの2つがビットストリームデータチャンネルに変換されるワードデータチャンネルに対する仮想フレームチャンネルとして使用され、他のフレームチャンネルが制御チャンネルに使用される、4つのビットストリームフレ

50

ームチャンネルを有する。行2中の実施例は、フレームチャンネルのうちの3つがビットストリームチャンネルデータに使用され、フレームチャンネルのうちの4つがビットストリームデータに変換されるワードデータチャンネルに使用され、他のフレームチャンネルが制御チャンネルに使用される、8つのビットストリームフレームチャンネルを有する。行3中の実施例は、チャンネルのうちの3つがビットストリームチャンネルデータに使用され、チャンネルのうちの4つがビットストリームデータに変換されるワードデータに対する仮想フレームチャンネルとして使用され、別のフレームチャンネルが制御チャンネルとして使用される、8つのビットストリームフレームチャンネルを有する。この場合、2つのビットストリームフレームチャンネルが、各TDMデータチャンネルを搬送するために使用される。行4中の実施例は、フレームチャンネルのうちの4つがビットストリームチャンネルデータに使用され、フレームチャンネルのうちの3つがビットストリームデータに変換されるワードデータに対する仮想フレームチャンネルとして使用され、他のフレームチャンネルが制御チャンネルとして使用される、8つのビットストリームフレームチャンネルを有する。行5中の実施例は、フレームチャンネルのうちの4つがビットストリームチャンネルデータに使用され、フレームチャンネルのうちの3つがビットストリームデータに変換されるワードデータに対する仮想フレームチャンネルとして使用され、他のフレームチャンネルが制御チャンネルとして使用される、8つのビットストリームフレームチャンネルを有する。本実施例では、3つのTDMデータチャンネルは、3つのビットストリームフレームチャンネルに分散される。行6中の実施例は、フレームチャンネルのうちの5つがビットストリームチャンネルデータに使用され、フレームチャンネルのうちの2つがビットストリームデータに変換されるワードデータに対する仮想フレームチャンネルとして使用され、他のフレームチャンネルが制御チャンネルとして使用される、8つのビットストリームフレームチャンネルを有する。行8中の実施例は、フレームチャンネルのうちの6つがビットストリームチャンネルデータに使用され、フレームチャンネルのうちの1つがビットストリームデータに変換されるワードデータに対する仮想フレームチャンネルとして使用され、別のフレームチャンネルが制御チャンネルとして使用される、8つのビットストリームフレームチャンネルを有する。第3および第5の実施例については、データ同期化ビットが使用される。データ同期化ビットは、マルチフレームサイクルの開始時にアクティブにされることができ、伝送機および受信機の両方がフラクショナルフローの間の同期化を取得するために同一の分数カウンタを使用する。実施例として、分数カウンタの比は、 $1/2$ であり得、すなわち、データは、1つおきのサブフレーム中で開始し、データは、チャンネルAから開始する。次いで、次のサブフレーム中で、データは、フレーム境界がなかったかのように出力され続ける。次いで、第3のサブフレーム中で、このプロセスが再び新たに開始する。これの全ては、1つおきのフレームで分数カウンタをリセットする、DS0ビットによって開始される。このシナリオでは、分数カウンタは、 0 、 $1/2$ 、 $1 \rightarrow 0$ 、 $1/2$ 、 $1 \rightarrow 0$ 等を数える。伝送機を伴うデバイスは、DS0ビットを制御するであろう。

【0327】

図48cに関して、行1中の実施例では、フレーム形式は、8つのビットストリームに分割される。1つのビットストリームフレームチャンネルが、制御情報を搬送するために使用され、ビットストリームフレームチャンネルのうちの3つが、オーバーサンプリングされた情報(例えば、3つのデータビットストリーム)を搬送するために使用され、4つの残りのビットストリームフレームチャンネルが、これらのチャンネルの間で多重化されるTDM情報を伴う5つのワードデータチャンネルを搬送するために使用される。行2中の実施例では、1つのビットストリームフレームチャンネルが、制御情報に使用され、3つのビットストリームフレームチャンネルが、オーバーサンプリングされたデータ(例えば、3つのデータビットストリーム)に使用され、残りの4つのビットストリームフレームチャンネルが、各々20ビットの精度を伴う10個の多重化TDM(ワードデータ)オーディオチャンネルを搬送するために使用される。行3の実施例では、1つのビットストリームフレームが、制御情報を搬送するために使用され、5つのビットストリームフレームチャンネルが、オーバーサンプリングされたデータ(例えば、5つのデータビットストリーム)を搬送するた

10

20

30

40

50

めに使用され、残りの2つのビットストリームフレームチャネルが、5つの多重化TDMワードデータチャネルを搬送するために使用される。

【0328】

図45aから47cの例示的組み合わせは、オーディオワードデータ、例えば、場合によっては、ワード形式、ビットストリーム形式、または統一ビットストリーム形式である計装またはセンサデータ等の異なるタイプのワードデータとともに使用できることに留意されたい。

【0329】

統一バス通信プロトコルはまた、種々のデータ形式をサポートすることもできる。少なくとも1つの実施形態では、統一バス通信プロトコルは、これらの異なる形式のデータに対するトランスポート媒体を提供することによって、4つの異なるデータ形式、すなわち、バイナリワード、ビットストリーム、文字列、および浮動小数点をサポートすることができる。他の実施形態では、これらのデータ形式の異なる組み合わせをサポートすることができる。

10

【0330】

バイナリワード形式は、ワードモード中にバス64上で伝送されるワードデータを表すために使用されることができる。MSBが最初である、2の補数形式を使用して、2進符号化を行うことができる。この場合、数は、1という最大数値を有するように拡大縮小されているものとして解釈されることができる。統一バス通信プロトコルの実施形態に対するバイナリワード形式でNビットを符号化するための一般的形式の実施例が、図49aに示されている。このデータ形式は、I²SおよびTDM符号化と直接適合性がある。

20

【0331】

ビットストリーム符号化形式は、ビットストリームモード中にバス64上で伝送される、オーバーサンプリングされたデータを表すために使用されることができる。オーバーサンプリングされたデータは、典型的には、限定されないが、例えば、デジタルマイクロホンからの出力等のシグマデルタ変換器からの出力として導出される。いくつかのクラスD増幅器はまた、オーバーサンプリングされたデータを入力データとしてサポートする。オーバーサンプリングされたデータは、未加工形態で処理されることができるか、またはベースバンドデータを取得するように以降でデシメートされることができる。使用されるオーバーサンプリング係数は、概して、システム要件に依存する。電話システム用のビットストリーム符号化形式携帯電話とうまく連動する、いくつかの共通オーバーサンプリング係数の実施例が、図49bに示されている。

30

【0332】

ビットストリーム符号化形式は、帯域幅効率があまり良くなく、したがって、低遅延のために、またはPCBライン上で使用されることができ、その場合、帯域幅がより高くあり得る。例えば、ビットストリーム形式がヘッドセットインターフェースに使用される場合、16kHz等の48kHzより低いサンプルレートを使用することができ、16kHzサンプルレートは、例えば、64のオーバーサンプリング率(OSR)で1.024MHz帯域幅要求をもたらす。整数サンプルレート変換と同時に44.1および48kHzサンプルレートをサポートすることができるため、7.056MHzが特別な周波数であることに留意されたい。したがって、この周波数は、(非整数アップサンプリングおよびミキシングが過渡歪みをもたらし得るが、)可能な限り最高のオーディオ品質をサポートすることができる。デシメーションプロセスの一部として3倍の補間を行うことによって、66.67倍のデシメーションを達成することができる。この構成は、19.200MHzクロックシステムに使用されることができる。

40

【0333】

文字列符号化形式は、ASCII形式を使用し、長さ1バイトの全ての文字が許容される。例示的実施形態では、第1の文字が1に等しい場合、文字列がその後に続き、ゼロで終端されるであろう。第1のバイトが254に等しい場合、次の8バイトは文字列の長さを示すであろう。文字列が無制限長を有する場合、第1の文字は255に等しいであろう

50

。文字配列が終了した場合、ゼロを使用することができる。統一バス通信プロトコルの実施形態に対する文字列符号化形式のための可能な組み合わせの実施例が、図49cに示されている。

【0334】

文字列符号化形式は、マスタデバイス52がスレーブデバイス54から読み取り、およびスレーブデバイス54に書き込むときの両方に使用されることができる。文字列の開始の定義は、チャンネルのアクティブ化からであり、スレーブデバイス54がこのコマンドを受信した後に、Xワードの後の次のフレームの開始時に始まる。マスタデバイス52は、文字列を解釈しないが、これらの文字列をトランスポートするにすぎない。変数がBCD形式（すなわち、2進コード化デジタル形式、例えば、「34」は「00110100」
10
としてコード化される）で符号化される場合、文字列形式が使用され得る。BCD形式でデータを受信するデバイスは、プリアンブルを除去することができる。

【0335】

例示の実施形態では、浮動小数点形式は、浮動小数点数の形式を表すために、IEEE
数値規格754-2008を使用する。圧縮符号化形式が使用されるとき、変数が単精度
である一方で、一般符号化形式での精度は、符号化において特定される精度と同一であ
ろう。ポートから読み取られる第1のバイト、またはポートに書き込まれる第1のバイト
は、最上位ワードであり得る。少なくとも1つの実施形態では、マスタデバイス52をサ
ポートするドライバは、以下のIEEE浮動小数点形式、すなわち、binary32、
binary64、decimal32、およびdecimal64を符号化および復号
20
するように構成されることができる。

【0336】

場合によっては、バス64に使用されるクロックレートに正確に合致するデータフロー
を転送することが可能ではない場合があることに留意されたい。実施例として、48kHz
接続を経由して44.1kHzコンテンツを転送することが必要であり得る。両方のサン
プルレートが同一のクロックソースに同期化される場合、2つのサンプルレートの間の
比（すなわち、44.1/48）は整数ではない。しかしながら、それは、正確な分数（
147/160）であり、バス64を経由したトランスポートは、非同期プッシュ/プル
を用いることなく制御されることができる。このタイプのトランスポートは、フラクショ
ナルフローと呼ばれ、少なくとも1つの実施形態ではマスタデバイス52によって制御さ
30
れることができる。フラクショナルフローは、上記のシナリオの単純かつ効率的なサポー
トを可能にする。DS0およびDS1ビットは、フラクショナルフローを制御するために
使用されることができ、全ての動作中に定義される。

【0337】

DS0ビットを設定する1つの方法は、位相累算器を使用し、DS0ビットを制御する
ためにオーバーフロー搬送を使用することであり得る。実施例として、DS0またはDS
1ビットへの書き込みは、8ビット分数位相累算器（ACC）を使用して制御されること
ができる。分数位相累算器は、原理上、有限量の位相 $= 2 \cdot (X/Y)$ を総位相 に
加算し、 X が 2 より大きいときはいつでも、 X から 2 を差し引き、SYNCフラグを
40
高に設定することによって機能する。図50aに示されるアルゴリズムは、少なくとも1
つの実施形態では、統一バス通信プロトコルのこのプロセスを実装するために使用され
ることができる。しかしながら、量 $(2^N - Y)$ が最初に計算される場合、単一の加算器が
使用されるように、アルゴリズムを訂正することが可能である。

【0338】

実施例として、 $X = 147$ 、 $Y = 160$ 、 $N = 8$ （これは48kHzチャンネル上の44
．1kHzデータフローに合致する）、および8ビット累算器があると仮定する。サイク
ル毎に、値Xが累算器ACCに加算され、ACC値がYより大きいとき、次いで、Yが累
算器ACC値から差し引かれるであろう。この例示的シナリオの下での計算が、図50b
に示されている。加算は、（8ビット累算器については）256を法とする整数値を使用
50
して示されていることに留意されたい。いくつかの実施形態では、一般的に、16ビット

累算器を使用することができる。他の実施形態では、一方の変数に対する10ビットおよび他方に対する6ビット等の他の分解能を使用することもできる。場合によっては、変数は、この目的で割り付けられるビットの使用と交換され得る。

【0339】

種々の再生シナリオに対するXおよびYのいくつかの典型的な値の実施例が、図50cに示されている。特別なシステム周波数(例えば、26.000または27.000MHz)が使用され得る場合には、次いで、時間基準として使用することができる、1.00または2.00MHzまで落とすために、クロック分割器が使用され得る。このようにして、たとえば8ビットだけの分解能がXおよびYに対して提供されたとしても、分数分割器は、依然として種々の周波数をサポートすることができる。

10

【0340】

本明細書で説明される統一バス通信プロトコルの種々の実施形態は、種々の異なる用途で使用されることができる。例えば、統一バス通信プロトコルは、チップ間通信を促進するために使用されることができ、その実施例は、図51aおよび51bに示されている。統一バス通信プロトコルはまた、種々のデバイスの間の通信を促進するために使用することもでき、その実施例は、図52aから52iに示されている。

【0341】

ここで図51aを参照すると、統一バス通信プロトコルを使用するディスプレイを伴う制御システム220の例示的实施形態が示されている。制御システム220は、処理ユニット222と、ADC224と、ADC226と、DAC228と、ディスプレイ230と、これらの要素の全てを一緒に連結するバス232とを備える。処理ユニット222は、プロセッサ、フィールドプログラマブルゲートアレイ(FPGA)、特定用途向け集積回路(ASIC)、または当業者によって公知であるような何らかの他の処理回路であり得る。本実施例では、バス232は、1本のワイヤが、クロック信号を伝送するために使用され、別のワイヤが、データ、同期化、およびコマンドデータを伝送するために使用される、2ワイヤ実装を有する。本実施例では、処理ユニット222は、マスタデバイスとしての役割を果たし、スレーブデバイスは、ADC224、ADC226、DAC228、ディスプレイ230である。

20

【0342】

ここで図51bを参照すると、統一バス通信プロトコルを使用する携帯電話システム250の例示的实施形態が示されている。携帯電話システム250は、例えば、携帯電話またはスマートフォンであり得る。携帯電話システム250は、ベースバンドプロセッサ252と、CODEC254と、クラスD増幅器256と、容量センサ258と、IRセンサ260と、クラスD増幅器262と、FMラジオ264と、Bluetooth(登録商標)モジュール266と、4つのマイクロホン268から274と、2つのバス276および278とを備える。バス276は、要素252から266を連結するために使用され、バス278は、要素252、254、および268から274を連結するために使用される。本実施例では、バス276および278は、以前に説明されたように、2ワイヤバス実装形態を使用して実装される。さらに、本実施例では、ベースバンドプロセッサ252は、マスタデバイスとしての役割を果たし、他の要素254から274は、スレーブデバイスとしての役割を果たす。多くのデバイスが同一のバスに連結されるとき、容量損失の増加、およびより低いエネルギー効率が生じるであろうため、システムをよりエネルギー効率的にするために、1つより多くのバスが使用される、用途があり得る。加えて、異なるバスは、より低いクロック速度がよりエネルギー効率的である、異なるクロック速度を有することができる。統一バス通信プロトコルの別の側面は、それが、低費用デジタル付属品を携帯電話システム250に接続するバス上で使用され得ることである。

30

40

【0343】

ここで図52aを参照すると、統一バス通信プロトコルを使用する住宅用安全システム300の例示的实施形態が示されている。住宅用安全システム300は、コントローラ302と、無線モデム304と、電話線306と、ガスセンサ308と、二酸化炭素(CO

50

2) センサ310と、スイッチ312から318と、バス320とを備える。スイッチ312から318は各々、スイッチの状態(例えば、窓またはドアが開いている、または閉じている)に応じて、論理0出力または論理1出力を与えるように、スレーブデバイスに接続される。たとえセンサの各々を伴うスレーブデバイスを含むことのわずかな追加の費用が生じたとしても、これは、より安価な設置費用によって相殺され、それが、スレーブデバイスの追加費用を多大に超え得る。コントローラ302は、本明細書で説明される任意の処理ユニットによって実装されることができ、バス320は、クロック、データ、コマンド、および同期化情報が同一のバスライン上で送信される、単一ワイヤ実施形態として実装されることができ、バス320は、コントローラ302を要素308から318に連結する。コントローラ302は、マスタデバイスとしての役割を果たし、要素308から318は、スレーブデバイスとしての役割を果たす。無線モデムおよび電話線306は、線の状態を直接チェックし、警告が着信するときに適切な権限者に報告する(例えば、ある条件が発生するときに、所有者、消防署、警察等に電話する)ことができる、監視中央ステーションまたは家の所有者と通信するために使用され得る。無線モデム304は、電話線306が故障または破損している場合に、バックアップとしての機能を果たすことができる。

10

【0344】

ここで図52bを参照すると、統一バス通信プロトコルを使用する家庭用娯楽システム350の例示的实施形態が示されている。家庭用娯楽システム350は、DVDプレーヤ352と、増幅器354と、スピーカ356から362と、バス364とを備える。バス364は、DVDプレーヤ352を増幅器354に連結する。バス364は、例えば、標準デジタルケーブルにおいて、単一のワイヤとして実装されることができ、本実施例では、DVDプレーヤ352は、マスタデバイスとしての役割を果たし、増幅器354は、4つのスピーカ356から362に対する4つのオーディオチャンネルとともに、スレーブデバイスとしての役割を果たす。

20

【0345】

ここで図52cを参照すると、統一バス通信プロトコルを使用する家庭用娯楽システム400の別の例示的实施形態が示されている。家庭用娯楽システム400は、DVDプレーヤ402と、HD TVセット404と、スピーカ406から412と、バス414とを備える。バス414は、DVDプレーヤ402をHD TV404に連結する。バス414は、例えば、標準デジタルケーブルにおいて、単一のワイヤとして実装されることができ、本実施例では、DVDプレーヤ402は、マスタデバイスとしての役割を果たし、HD TV404は、4つのスピーカ406から412に対する4つのオーディオチャンネルとともに、スレーブデバイスとしての役割を果たす。

30

【0346】

ここで図52dを参照すると、統一バス通信プロトコルを使用する計装システム450の例示的实施形態が示されている。計装システム450は、マルチメータ452と、センサ454と、これら2つの要素を連結するバス456とを備える。バス456は、例えば、標準デジタルケーブルにおいて、単一のワイヤとして実装されることができ、本実施例では、マルチメータ452は、マスタデバイスとしての役割を果たし、センサ454は、スレーブデバイスとしての役割を果たす。

40

【0347】

ここで図52eを参照すると、統一バス通信プロトコルを使用して通信することができる、電子キー500の例示的实施形態が示されている。この場合、電子キー500は、IDを電子キー500から読み取ることができるよう、統一バス通信プロトコルを使用するバス(図示せず)に接続されることができ、2つの端子502および504を有する。より高い機械的ロバスト性のために、端子は、代替実施形態では、電子キー500の反対側に実装されることができ、電子キー500は、スレーブデバイスとしての役割を果たす。

【0348】

50

ここで図5 2 fを参照すると、統一バス通信プロトコルを使用して通信することができる、メモリスティック5 5 0の例示的实施形態が示されている。メモリスティック5 5 0は、3つの端子、すなわち、正の端子5 5 2、負の端子5 5 4、およびデータ端子5 5 6を有する。データ端子5 5 6は、統一バス通信プロトコルを使用するバス(図示せず)に接続されることができる。メモリスティック5 5 0は、スレーブデバイスとしての役割を果たす。

【0 3 4 9】

ここで図5 2 gを参照すると、統一バス通信プロトコルを使用して通信することができる、加入者識別モジュール(S I M)カード6 0 0の例示的实施形態が示されている。S I Mカード6 0 0は、チップ6 0 2、および端子6 0 4から6 0 8を有する。S I Mカード6 0 0は、電流消費を数mA以下に保つことができる場合、2つの端子とともに実装されることができる。代替として、S I Mカード6 0 0は、図5 2 gに示されるように、より高い所要電力のために3つの端子を使用して実装されることができ、その場合、端子6 0 4から6 0 8のうちの1つは、電力に使用され、端子6 0 4から6 0 8のうちの他の2つは、情報を通信するために使用される。

【0 3 5 0】

ここで図5 2 hを参照すると、統一バス通信プロトコルを使用して通信することができる、暗号化クレジットカード6 5 0の例示的实施形態が示されている。暗号化クレジットカード6 5 0は、チップ6 5 2、および3つの端子6 5 4から6 5 8を有する。統一バス通信プロトコルの使用は、暗号化クレジットカード3 5 0が、より少ない端子とともに実装されることを可能にする。端子6 5 4から6 5 8のうちの1つは、電力に使用され得、端子6 5 4から6 5 8のうちの他の2つは、情報を通信するために使用され得る。クレジットカードチップ6 5 2の電力消費が十分に低い(例えば、3 mAを下回る)場合には、電力を移送するために使用される端子が使用されなくてもよく、単一ワイヤ物理層が代わりに使用され得る。

【0 3 5 1】

他の実施形態では、本明細書で説明されるような統一バス通信プロトコルは、無線通信に使用され得る。一実施形態では、クレジットカードは、例えば、13.8 MHzに位置する搬送波を使用して、情報の無線転送のための誘導に基づく回路を含み得、次いで、転送は、RF場をロードまたはアンロードすることによって取り扱われる。次いで、マスタデバイスが、磁場を変化させることによってスレーブデバイスと通信することができる一方で、スレーブデバイスは、磁場の異なるローディングによって信号伝達し得る。本明細書で説明される統一バス通信プロトコルの種々の実施形態の無線バージョンもまた、他の用途に使用されることができる。

【0 3 5 2】

カード読み取り機が、商取引を取り扱うために使用され、したがって、データセキュリティが、不正コピーおよび盗用から守るために使用される。統一バス通信プロトコルの一側面は、デジタル転送データの暗号化の使用による、これらの状況での安全な解決策の実施可能性である。アナログ信号が直接利用可能ではなく、適切な暗号化キーなしでデジタル符号化信号を復号することが可能ではないため、これは、カード読み取り機からアナログ出力信号をコピーする可能性を無効にするであろう。

【0 3 5 3】

ここで図5 2 iを参照すると、統一バス通信プロトコルを使用する心拍数モニタシステム7 0 0の例示的实施形態が示されている。心拍数モニタシステム7 0 0は、監視デバイス7 0 2と、例えば、指クリップセンサ等の心拍数センサ7 0 4と、それらを一緒に連結するバス7 0 6とを備える。バス7 0 6は、単一のワイヤを使用して実装されることができる。本実施例では、監視デバイス7 0 2は、マスタデバイスとしての役割を果たし、センサ7 0 4は、スレーブデバイスとしての役割を果たす。一実施形態では、監視デバイス7 0 2は、指クリップセンサ7 0 4によって測定される情報に基づいて、心拍数情報を計算する、特殊ソフトウェアアプリケーションを実行する、スマートフォンであり得る。代

10

20

30

40

50

替実施形態では、特殊ハードウェアおよびファームウェアは、監視デバイス702を実装するために使用されることができる。代替実施形態では、心拍数情報を決定するために使用される、生理学的情報を測定するために、他のタイプのセンサを使用することができる。

【0354】

統一バス通信プロトコルを使用することができるデバイスの他の実施例は、バッテリーモニタ、ディスプレイを伴うヘッドセット、雑音低減のための複数のマイクロホンを伴うヘッドセット、ステレオ録音能力を伴うヘッドセット、感度および/または同調を報告するヘッドセット、完全デジタルインターフェースを伴うヘッドセット、低費用センサインターフェース等を備える。バッテリーモニタの場合、スマートバッテリーは、典型的には、3つの端子、すなわち、正電圧端子、負電圧端子、ならびにバッテリーIDおよび他の情報がスマートバッテリーから読み取られることを可能にするデータラインを有する。したがって、統一バス通信プロトコルは、スマートバッテリーと通信し、例えば、温度等のバッテリーパラメータを監視するために使用されることができる。

10

【0355】

ここで図53を参照すると、本明細書で説明される統一バス通信プロトコルの1つ以上の実施形態に従ってバス64を動作させる方法800の例示的实施形態が示されている。

【0356】

802では、統一バス通信プロトコルが、マスタデバイス52によってアクティブにされる。これは、電源投入イベント、リセットイベント後に起こるか、または、何らかの他のイベントによって開始されることができ、その後、例えば、コントローラまたは状態マシンが、マスタデバイス52をプログラムし始めるであろう。802では、マスタデバイス52によって使用される種々のレジスタが、初期化され、統一バス通信プロトコルを使用してバス64を経由して通信することができるように適正な値に設定される。これは、コントローラ等の別のデバイスによって行われ得、またはマスタデバイス52が、例えば、メモリからこれらの設定にアクセスすることができる。

20

【0357】

804では、動作モードが構成される。この動作モードは、バスを経由して通信するために使用されるフレーム形式に対する種々のパラメータを選択することを含む。例えば、この時に、マスタデバイス52は、ワードモードまたはビットストリームモードを使用することの間で選択することができる。

30

【0358】

806では、次いで、マスタデバイス52が、ワードフレーム形式、ビットストリームフレーム形式、または統一ビットストリームフレーム形式のうちの1つから、選択されたモードに使用されるフレーム形式を構成することができる。この時に、少なくともいくつかの実施形態では、マスタデバイス52はまた、データ伝送において故障を引き起こすことなく、マスタデバイス52が動作中に通信の異なるデータモードの間でシームレスに切り替え得るように、種々のバンクレジスタに通信の異なるデータモードを保存することもできる。異なるデータモードは、限定されないが、異なるサンプリングレート、異なるチャンネル選択、異なるポート選択、所与のポートの種々の選択されたチャンネルに対するフレームチャンネルの異なる割付、フレームチャンネルの割付におけるポートチャンネルの異なるサブグループ、フレームチャンネルの割付におけるポートチャンネルの異なる反復、および共通フレームチャンネルへの異なるチャンネルからのデータの多重化のうちの1つ以上の組み合わせ等の選択されたフレーム形式に対する種々のパラメータを特定することができる。いくつかの実施形態では、マスタデバイス52は、高速安定化発振器がUARTデバイスをより良好にサポートすることを可能にすることによって、低電力モード完了後の高速起動を可能にし得る。

40

【0359】

808では、バス64を経由して通信したいスレーブデバイスが、マスタデバイス52

50

と同期化する。これは、Sワードデータの伝送に対してバス64上のアクティビティを監視することによって、スレーブデバイスによって行われる。ワードモードでは、Sワードのビットが、バス64上で連続的に伝送される。ビットストリームモードでは、Sワードのビットは、連続的に伝送されないが、むしろ他のフレームチャネルからのビットと時間多重化される。しかしながら、両方の場合において、少なくとも1つの実施形態では、スレーブデバイス54は、Sワードを認識するとともに、同期化プロセス中にフレーム長およびフレーム構造を決定する。フレーム構造は、同期化プロセスの一部としてスレーブデバイス54によって認識されるが、同期化（すなわち、ロック）を失うことなく、スレーブデバイス54が1つのフレーム形式から別のフレーム形式へ即時に変化し、それによって、フレーム形式の即時変化を可能にする時間がある場合、次のフレーム形式がどのようなものになるであろうかを示すために、スレーブデバイス54の内側のレジスタが含まれることができる。この情報は、COMMAND SEPARATIONレジスタ中に含まれ得、同期化後にマスタデバイス52によってスレーブデバイス54においてプログラムされることができる。ワードおよびビットストリームモードの両方で、スレーブデバイス54は、バス64上のアクティビティを監視し、Sワードのビットが連続的に送信されるか、または他のデータビットと時間多重化されるかを決定することによって、同期化を試行する。

【0360】

少なくとも1つの実施形態では、Sワードの動的同期部分が決定論的方法を使用して生成されるので、スレーブデバイス54は、所与のSワードの一定同期部分を検索し、Sワードの動的同期部分に着目し、次のSワードの動的同期部分を計算することによって、バス64上のアクティビティを監視する。少なくとも1つの実施形態では、決定論的方法は、例えば、CRCカウンタであり得るが、それに限定されない。CRCカウンタは、典型的には値ゼロに遭遇しないようにプロビジョンを使用する（その場合、ゼロで止められるであろう）。これは、この値についてゲート制御し、ゼロの値が発生する場合にCRCカウンタを別の値に設定することによって行われることができる。別の実施形態では、CRCワードは、適切な出力パターンを生成するようにコード化される論理ゲートを伴うバイナリカウンタからの出力として生成されることができる。少なくとも1つの実施形態では、異なる一定部分が、ワードモードおよびビットストリームモードのためのSワードに対して使用されることができる。少なくとも1つの実施形態では、異なる一定部分はまた、統一ビットストリームフレーム形式が使用されるときにSワードに対して使用されることもできる。

【0361】

同期化プロセスをよりロバストにするために、少なくとも1つの実施形態では、スレーブデバイス54は、Sワードの一定部分を検索し、Sワードの動的部分に注意し、次のSワードの動的部分を数回計算することができる。これら3つのステップは、同期化チェックプロセスと称することができる。この同期化チェックプロセスがより多くの回数で繰り返されるほど、同期化プロセスがよりロバストになるであろう。なぜなら、スレーブデバイス54が、ワードモードが使用されているか、ビットストリームモードが使用されているかを適正に検出し、例えば、ポート等のバス64にアタッチされる他のデバイスによって生成される誤った同期化パターンに同期化するよりもむしろ、マスタデバイス52と同期化する可能性が高くなるからである。しかしながら、同期化チェックプロセスの反復を増加させることと、同期化のための時間を短縮することとの間のトレードオフがある。同期化を達成する種々の方法が、本説明において以降でさらに詳細に議論される。

【0362】

810では、マスタデバイス52は、スレーブデバイスが挿入されているか、または除去されているか、換言すると、スレーブデバイスがバス64と接続しようとしているか、またはバス64から接続解除しつつあるかを検出する。バス64がすでに動作を開始した後、デバイスからバス64への物理的接続が起こる場合、これは、ホット挿入と称される。これは、スレーブデバイス52がバス64にアタッチされる前に受電するか、または

10

20

30

40

50

電力が印加されないときでさえも、別様にバス64に影響を及ぼさないかのいずれかある限り、本明細書で説明される統一バス通信プロトコルのうちの少なくとも1つで可能である。さらに、バス64は、バス64がアクティブであるときでさえも、デバイスが物理的に除去されることを可能にし、これは、ホット除去と称される。少なくとも1つの実施形態では、これは、PINGコマンドを発行することによって、マスタデバイス52によって行われることができる。PINGコマンドが発行された後、マスタデバイス52と同期化し、バス64にロックされている各スレーブデバイスは、XおよびYコマンドワードがバス64を経由して伝送されるときに、あるタイムスロット中でデータを伝送することによって、そのスレーブ状態を伝送することができる(実施例については図9a、9b、および10、ならびに関連説明を参照)。次いで、マスタデバイス52は、どのスレーブデバイスがバス62に接続しようとしているかを決定するために、XおよびYコマンドワード中のスレーブ状態フィールドの値を読み取ることができる。実施形態では、これは、スレーブデバイスがスレーブ状態として「01」、「10」、または「11」の値を書き込んでいる場合に決定されることができる。スレーブデバイスがバスから接続解除する場合、それが占有していたアドレスは、利用可能になり、別のスレーブデバイスに対して使用されることができるか、または、マスタデバイス52およびバス64と再度同期化されるまで、同じスレーブデバイスのために保留されたままとなることができる。

10

【0363】

少なくとも1つの実施形態では、割り込みビット(IRQ)が、Sワードに関連付けられたタイムスロットに書き込まれることができ、それは、スレーブデバイスのうちの1つが注意を必要とすることをマスタデバイス52に通知し、その時点で、マスタデバイス52は、PINGコマンドを発行し、バス64に接続しようとしているスレーブデバイスの状態を決定することができる。少なくとも1つの実施形態では、割り込みビット(IRQ)は、スレーブデバイスの状態がある様式で変化する場合に、自動的に発行されることができる。

20

【0364】

812では、マスタデバイス52は、バス64に接続しようとしており、マスタデバイス52と同期化しているスレーブデバイスにアドレスを割り当てる。実施形態では、これらのスレーブデバイスは、ゼロのデバイスアドレスを有することになり、マスタデバイス52は、REQUEST ID関数を発行することによって、これらのスレーブデバイスを検出する(実施例については図11aおよび関連説明を参照)。この瞬間にバス64にロックされていないスレーブデバイスは、アクティブにされず、REQUEST ID関数に回答しないであろう。少なくとも1つの実施形態では、複数のスレーブデバイスが、電力上昇イベントまたはリセットイベント後にゼロのデバイスアドレスを有することができるので、次いで、マスタデバイス52は、異なるアドレスをこれらのスレーブデバイスに割り当てるために、本説明で以前に説明された動的アドレス割付方法を行うことができる。したがって、例えば、I²C等の他のバス通信プロトコルと異なり、スレーブデバイスのデフォルトアドレスは、統一バス通信プロトコルを用いて固定される必要がない。これは、多くの場合、I²C(7ビットが使用される)において遭遇される、デバイスアドレス衝突に関する問題を有することなく、相対的(例えば、4ビット)アドレス空間の使用を可能にする。この問題を解決するために、I²Cデバイスのデバイス製造業者は、多くの場合、代替的なアドレスのための余分なピンを含むが、これは、余分な費用および空間を追加するか、または同一のチップのプログラムされた変異形を作成し、それによって、運搬上の問題につながる。これは、本明細書で説明される統一バス通信プロトコルの種々の実施形態とともに動的アドレス割付方法を使用することによって、回避されることができる。

30

40

【0365】

814では、マスタデバイス52は、バス64とちょうど同期化され、連結され、所与のデバイスアドレスが与えられているスレーブデバイスを構成する。この構成は、ポートプログラミング、および選択されたフレーム形式に従った選択されたデータチャネルから

50

のデータに対するタイムスロットの割当等の種々の動作を含むことができる。一般に、この構成は、データ伝送のために、同期化されたスレーブデバイスのうちの少なくとも1つの少なくとも1つのポートから、データチャンネルを選択することを伴う。

【0366】

少なくとも1つの実施形態では、マスタデバイス52は、そのポートプログラミングについてスレーブデバイス54に指示する。これは、第1のポートへの入力データを受信するデータチャンネルと一緒にグループ化することと、出力データを第2のポートに伝送するデータチャンネルと一緒にグループ化することを含むことができる。少なくとも1つの実施形態では、グループ化はまた、所与のポートのチャンネルが、同一の周波数でサンプリングされるデータを有するように行われることもできる。マスタデバイス52はまた、番号を、スレーブデバイス54に対して定義されている種々のポートに割り当てることもできる。

10

【0367】

少なくとも1つの実施形態では、たとえいくつかのデータチャンネルがポートに割り当てられなくても、データチャンネルの全てが使用されないこともある。したがって、マスタデバイス52は、データ伝送またはデータ受信のためにアクティブになるであろう、ポートのデータチャンネルの一部分を選択することができる。少なくとも1つの実施形態では、これは、CHANNEL SELECTIONフィールドを使用して行われることができる（実施例については図16および関連説明を参照）。例えば、8つのデータチャンネルを伴うポートは、「00001100」に設定されたCHANNEL SELECTIONフィールドを有することができ、その場合、データチャンネル2および3が、（データチャンネル番号付けが7から始まる場合）アクティブなデータチャンネルであるように選択される。

20

【0368】

少なくとも1つの実施形態では、次いで、マスタデバイス52は、統一バス通信プロトコルとともに使用するために選択されているフレーム形式に基づいて、プログラムされたポートに対してデータが配置されるべき場所（すなわち、どのタイムスロットが使用されるか）をスレーブデバイスに指示する。少なくとも1つの実施形態では、ポートは、それがビットストリームを搬送するか、TDMデータを搬送するかに関して定義され、異なるポートが、異なるデータ形式目的に対して使用される。

30

【0369】

ワードモードの場合、2つのコマンドワードの間の最小距離と、コマンドワードの後に、データが最初に配置されるタイムスロットとが特定されることができる。少なくとも1つの実施形態では、これは、STARTフィールドによって特定されることができる。各データチャンネルのデータ幅も特定することができ、それは、少なくとも1つの実施形態では、LENGTHフィールドを使用することによって行われることができる。

【0370】

ワードモードでは、少なくとも1つの実施形態では、データの転送を微調整することが可能であり、これは、複数のサンプルレートが、1つ以上のスレーブデバイスの1つ以上のポートに対する異なるデータチャンネルに同時に使用されるときに、有益かつ効率的である。例えば、少なくとも1つの実施形態では、サブフレーム中のデータチャンネルに対するデータの間に間隙があり得る（実施例については図21および23ならびに関連説明を参照）。少なくとも1つの実施形態では、サブフレーム中の異なるデータチャンネルの種々のグループ化があり得る（実施例については図21、23、および24ならびに関連説明を参照）。少なくとも1つの実施形態では、サブフレーム中の異なるデータチャンネルのグループ化の反復があり得る（実施例については図21、23、および24ならびに関連説明を参照）。少なくとも1つの実施形態では、単一のデータチャンネルのみが、サブフレーム中で繰り返され得る。少なくとも1つの実施形態では、REPEAT、SUBGROUP、およびSKIPフィールドを使用することによって、サブフレーム中のデータチャンネルのこれらの異なる構成を達成されることができる。これらの特徴のうちの少なくとも2つ

40

50

、およびこれらの特徴の全てを含む、これらの特徴の種々の組み合わせがあり得る、実施形態もあり得る。

【0371】

ワードモードでは、非同期データ転送を効率的にサポートすることも可能である。少なくとも1つの実施形態では、これは、このデータ伝送に使用される第1のフレーム中で、データチャンネルからのデータに先行する2ビットを追加することによって、達成されることができる。ビットのうちの1つは、伝送デバイスが、有効である伝送すべき新しいデータを有するかどうかを示すために使用され得、ビットのうちの1つは、受信デバイスが伝送された新しいデータを受信したかどうかを示すために使用され得る。

【0372】

ワードモードでは、少なくとも1つの実施形態では、例えば、44.1kHzでサンプリングされるデータが48kHz経路を経由して伝送されるとき等に、分数性質であるデータフローをサポートすることも可能であり得る。少なくとも1つの実施形態では、いくつかのフレームにわたるデータ転送を示すために、マスタデバイス52中の相加算器およびXコマンドワード中に位置するビットを使用することによって、分数データフローをサポートすることができる。

【0373】

ビットストリームモードの場合、ワードモードについて行われたように、同様の情報が特定されることができるが、これは、ビットストリームモードで使用される異なるフレーム形式により、わずかに異なって行われる。例えば、ビットストリームフレーム形式を使用することと、統一ビットストリームフレーム形式を使用することとの間で区別がなされることができる。両方の場合において、2つの制御ビットの間の最小距離、ならびにフレームチャンネルの数、フレームチャンネルの長さ、フレームチャンネルへのデータチャンネルの割付、および制御チャンネルとしてのフレームチャンネルの割付が特定されることができる。各フレームチャンネルのデータ幅は、1ビットである。

【0374】

統一ビットストリームフレーム形式では、少なくとも1つのフレームチャンネルが、デジタルワードデータを1度に1ビット伝送するために使用される、仮想フレームチャンネルとして割り付けられる。これは、ビットストリームデータチャンネルおよびワードデータチャンネルの両方を同時にサポートすることができる共通フレーム形式を可能にする。したがって、TDM信号伝達またはTDMデータ転送を、ビットストリームモードでサポートすることができる。実施形態では、これは、TDM WORDS TRANSFERRED IN BITSTREAM MODEフィールドを使用することによって達成されることができる。

【0375】

少なくとも1つの実施形態では、ビットストリームフレーム形式および統一フレーム形式の両方に対するフレームチャンネルのうちの少なくとも1つは、異なるデータチャンネルからのビットが、多重化フレームチャンネルと称することができる共通フレームチャンネル中で多重化される多重化フレームチャンネルとして割り付けられることができる(図28cおよび28dならびに説明の関連節が実施例を提供する)。これは、異なるフレームチャンネルに割り付けられたデータチャンネルが、異なるサンプリングレートを使用するときに行われることができ、それによって、より低いサンプリングレートを伴うデータチャンネルは、多重化フレームチャンネル中で(多重化フレームチャンネルの垂直方向へ)インターレースされることができる。

【0376】

少なくとも1つの実施形態では、ビットストリームフレーム形式および統一フレーム形式の両方について、データチャンネルに割り付けられるフレームチャンネルの数は、データ伝送中の制御フレームチャンネルおよび1つ以上のデータチャンネルによって使用される帯域幅を変化させるように、変化させられることができる。

【0377】

10

20

30

40

50

少なくとも1つの実施形態では、ビットストリームフレーム形式および統一フレーム形式の両方について、データの転送を微調整することが可能であり、これは、複数のサンプルレートがチャネルデータに同時に使用されるときに、有益かつ効率的である。例えば、少なくとも1つの実施形態では、フレーム中のデータチャネルに割り付けられるフレームチャネルの間に間隙があり得る（実施例については図28bおよび関連説明を参照）。少なくとも1つの実施形態では、フレーム中の異なるデータチャネルに割り付けられるフレームチャネルの種々のグループ化があり得る（実施例については26bおよび28dならびに関連説明を参照）。少なくとも1つの実施形態では、フレーム中の異なるデータチャネルのグループに割り付けられるフレームチャネルの反復があり得る（実施例については図26bおよび28bならびに関連説明を参照）。少なくとも1つの実施形態では、サブフレーム中のデータチャネルへのフレームチャネルのこれらの異なる構成および割付は、HSTART、VSTART、HSPACING、およびVSPACINGフィールドを使用することによって、達成されることができる。

10

【0378】

少なくとも1つの実施形態では、ワードモードについて説明されたのと同様に、ビットストリームモードで非同期データ転送を効率的にサポートすることが可能である。これは、フラクショナルフローをサポートしたチャネル/ポートが使用された場合に可能であり得る。例えば、そのようなチャネルは、3.072MHzチャネルで44.1kHz x 64ビットストリームデータを転送するために使用されることができる。

【0379】

20

図53をもう1度参照すると、816で、少なくとも1つのデータの伝送機および少なくとも1つのデータの受信機上の選択されたデータチャネルが、アクティブにされる。少なくとも1つの実施形態では、これは、ACTIVEフィールドに書き込むことによって行われることができる（実施例については図16および関連説明を参照）。これらの選択されたデータチャネルは、1つ以上のスレーブデバイスの1つ以上のポートからの1つ以上のデータチャネルであり得る。

【0380】

818では、選択されたデータチャネルに関連付けられたデータがバス64上で転送される。入力データチャネルである、選択されたデータチャネルについては、これらのデータチャネルからのデータは、バス64を経由して対応するスレーブデバイスから受信される。出力データチャネルである、選択されたデータチャネルについては、これらの選択されたデータチャネルに対するデータは、バス64を経由して対応するスレーブデバイスに送信される。一般に、818でのデータ転送は、バス64に連結される少なくとも2つのデバイス間でデータを伝送することを含み、転送されるデータは、数値データ、制御データ、およびクロックデータのうちの少なくとも1つを備える。

30

【0381】

820では、データが所与の選択されたデータチャネルに送信されるか、またはそこから受信される必要がない場合、選択されたデータチャネルのうちの少なくとも1つを非アクティブにされることができる。少なくとも1つの実施形態では、選択されたチャネルは、対応するポートのACTIVEフィールドを非アクティブにすることによって、非アクティブにされることができる。

40

【0382】

次いで、方法800は、804に戻ることができ、そこで、動作モードまたはフレーム形式が変更される必要があり得るかどうか決定されることができる。例えば、サンプリングレート等のデータチャネルのいくつかのパラメータが変更されるため、例えば、フレーム形式がオンザフライで変更される事例があり得る。他のシナリオでは、データの伝送機および受信機が変化し得る。例えば、オーディオ用途において、1つのシナリオでは、オーディオデータが、ヘッドホンジャックに向けられる一方で、別のシナリオでは、オーディオデータは、受信機、スピーカ、または複合受信機/スピーカに向けられる。いくつかのシナリオでは、1つの信号ソースから別の信号ソースへ徐々に変化すること等のいく

50

つかのミキシング状況を可能にするために、利得変化が複数のデバイスにおいて同時に起こり得る。この時点で、方法 800 は、作用 806 から 820 までループする。実施形態では、このループは、統一バス通信プロトコルが非アクティブにされるまで繰り返し行われることができる。例えば、ループは、外部イベントがオーディオセクション等の特定のアプリケーションを終了させる（例えば、通話の終了）場合に、終了することができ、その場合、デバイスは、低電力モードに設定され、クロックは、アイドル状態に設定されるであろう（例えば、MCLKD = 0000）。

【0383】

少なくとも1つの実施形態では、マスタデバイス52の種々のレジスタは、方法800においてスレーブデバイスについて説明されたように構成されることができる。

10

【0384】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態は、ビットストリームデータ、（例えば、I²SまたはSPDIFに接続するための）デジタルワードデータ、TDMデータ、および制御データ、ならびにI²CおよびI²S能力のうちの少なくとも1つに対する共通インターフェースサポートを提供する。したがって、本明細書で説明される統一バス通信プロトコルの種々の実施形態は、種々の標準インターフェースの間のデジタルハブとして使用されることができ、または場合によっては、本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態は、SLIMbus、McBSP、McPDM、およびMISOバス通信プロトコルのうちの少なくとも1つに取って代わるために使用されることができる。

20

【0385】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態で行われるように、ビットストリームデータとともに制御データを取り扱う能力は、ビットストリームデータに対する制御情報の転送のための標準方法が、従来のビットストリームバス通信プロトコルではまだ合意されていないため、有益である。さらに、従来の制御方法は、場合によってはクリック音のように鳴るオーディオデータへの妨害をもたらす。さらに、従来のビットストリームインターフェースは、2つより多くのビットストリームデータチャンネルがバスを経由して伝送される時に、より多くの端子を必要とする。従来、これは、より多くのデータラインを追加することによって（各データラインがチャンネル数を2つだけ増加させるであろう）、または、例えば、McPDMインターフェースで実装されるように、専用フレーム同期端子を追加し、ビットストリームチャンネルを多重化することによって達成されることができる。しかしながら、複数のビットストリームデータチャンネルに対するデータを取り扱うために、多重化ビットストリームフレームチャンネルが使用されることができ、余分な端子が同期化のために使用されないため、これは、統一バス通信プロトコルには当てはまらない。

30

【0386】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態の側面は、制御データと同時に低遅延を伴うビットストリームデータの等時性転送のサポート、ならびに非同期データストリームに対するサポートである。例えば、ビットストリームモードで達成される低遅延は、ビーム形成、能動雑音消去、近接性感知、漏出補償、一般的制御動作、または他の低遅延用途を含む、種々の用途を可能にする。

40

【0387】

また、本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態では、プロトコルの全ての特徴をサポートするわけではない、少なくとも1つのスレーブデバイスがあり得ることに留意されたい。これは、スレーブデバイスがサポートしない機能を実行するようにそれらに要求しないことにより、統一バス通信プロトコルによって取り扱われる。

【0388】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態の側面は、内部電力消費を増加させることなく（すなわち、チャンネルの数とは無関係の電力消費）、

50

複数のビットストリームデータチャネルに対するサポートを可能にする、スレーブデバイス中のゲートクロックの使用である。この場合、スレーブデバイスは、コマンドデータが利用可能であるとき、およびスレーブデバイスがそれに割り付けられる専用空間にデータを入力または出力するときを除いて、それらの内部クロックを無効にする。これは、たとえビットストリームの数が増加しても、内部クロッキング周波数が同じであり、この特定のフレーム構造をサポートすることができる小さい値であり得ることを意味する。完全システムクロックにおいて作動する必要があり得るスレーブデバイスの一部は、ビットストリーム同期化をチェックするスレーブデバイスのわずかな部分であろう。しかしながら、外部切り替え損失が、より高いシステムクロックとともに増加するであろう。

【0389】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態の一側面は、より効率的なデータパッキング、より良好なクロックサポート、および帯域幅に関するさらなる融通性のうちの少なくとも1つのための広い部類のフレーム構造に対するサポートである。例えば、用途および取り扱われているデータのタイプに応じて、ワードフレーム形式、ビットストリームフレーム形式、または統一フレーム形式を使用することができる。少なくともいくつかの実施形態では、ビットストリームフレーム形式および統一フレーム形式において、ビットストリームデータチャネルは、より良いデータ伝送および帯域幅効率のために、フレームチャネル中で種々の方法で組み合わせられることができる。少なくともいくつかの実施形態では、統一ビットストリーム形式は、より良いデータ伝送および帯域幅効率のために、ビットストリームデータチャネルとともにワードデータチャネルを伝送するために使用されることができる。少なくともいくつかの実施形態では、ビットストリームモードで、種々のチャネルからのデータは、データ伝送および帯域幅効率を向上させるように、一緒にグループ化されて繰り返されること等の種々の方法で組み合わせられることができる。

【0390】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態の側面は、2つのフレーム形式（すなわち、データモード）の間の瞬時的変化（例えば、オンザフライの変化）のサポートである。それは、プロトコルがこの変化に先立ってスレーブデバイスの準備を可能にすることができるからであり、データにおける故障を回避するであろう。これは、種々のデータモードを追跡するために1つより多くのレジスタを使用することによって達成されることができる。例えば、2つの異なるデータモードが、異なるクロック周波数またはフレーム形式を使用されることができる。

【0391】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態の側面は、ハンドシェイキングプロセスを使用しないバス64へのスレーブデバイスのホット挿入およびホット除去のサポートである。スレーブデバイスは、バス64が動作を開始した後にバス64にアタッチされることができる。バス64は、この場合、クラッシュせず、マスタデバイス52は、それがアドレスゼロに存在し、バス64に接続してマスタデバイス52に同期化するために注意を必要とするという通知をスレーブデバイスから受信するであろう。同様に、マスタデバイス52への事前通知を伴わずに、スレーブデバイスは、バス64から物理的に除去されることができる。スレーブデバイスは、それがバス64から接続解除していることを示すように、その状態レベルを変化させることができ、次いで、状態レベルの変化は、マスタデバイス52に通信されるであろう。

【0392】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態の側面は、以下の特徴のうちの少なくとも1つを実装することによる、内部または外部エラーに対するプロビジョンまたはエラー処理である。例えば、スレーブ状態は、例えば、少なくとも1つの実施形態では5ワードのビットで表されることができる、スレーブデバイスからの緊急要求を監視することによって、同期化の損失または別の問題を検出するために、連続的に監視されることができる。サポートされ得る別の特徴は、全てのREAD、WRITE

10

20

30

40

50

E、およびFUNCTION動作が、パリティチェックを含むことができることである。サポートされ得る別の特徴は、全てのREAD、WRITE、およびFUNCTION動作が、いかなるエラーも伴わずに動作が起こったことを確認するために、データおよび/または命令を受信するデバイスからの確認を含むことができることである。サポートされ得る別の特徴は、クラッシュするバスに接続されるデバイス、または違法データスロットに不慮に書き込むこと(本明細書で説明されるように種々のレジスタビットをチェックすることによって検出されることができ)の発見である。例えば、場合によっては、スレーブデバイスは、独力でバックオフし、バス64は、いくつかのフレーム以内で再起動するであろう(例えば、自己回復エラー処理)。したがって、I²Cバス通信プロトコルとは異なり、バス64は、統一バス通信プロトコルでは無期限にクラッシュすることは不可能である。

10

【0393】

ここで、本発明を実施するための形態の次の節は、マスタデバイスに、または統一バス通信プロトコル、ワードフレーム形式、またはビットストリームフレーム形式に従って通信するバスに、スレーブデバイスを同期化することの話題について議論する。

【0394】

いくつかの従来のバス通信プロトコルでは、バス上でデータを伝送するために使用されるフレーム形式のフレーム長は、固定され、同期化を促進するために同一の値にとどまる。例えば、SPDIF/TOSLINK、AC-97、およびSLIMbusバス通信プロトコルは全て、固定フレーム長を使用する。したがって、これらのプロトコルを使用して、限られた数のワイヤを経由してオーディオデータを伝送するとき、固定フレーム長が使用され、特定のクロック周波数およびサンプリングレートを効率的にサポートすることを困難にする。したがって、これらの特定のクロック周波数およびサンプリングレートでの動作は、増加した電力消費、および増加した帯域幅消費をもたらす。例えば、SLIMbusプロトコルは、電気通信用途で使用される標準周波数である、19.2MHzの周波数を効率的にサポートする限定された能力を有する。さらに、固定フレーム長を使用し、非同期的にフレーム形式に適合しないバスを経由して任意のデータを伝送することは困難である。このシナリオは、余分な帯域幅、余分なパツファ、および余分な電力消費を伴うであろう。場合によっては、非同期サンプルレートを使用することが可能であり得るが、これは、余分なシリコン面積および電力消費を必要とし、オーディオ用途のためのオーディオ品質の低下をもたらす。

20

30

【0395】

フレーム形式に可変フレーム長を使用する、I²S、McBSP、McPDM、および類似プロトコル等のいくつかの従来のバス通信プロトコルがある。しかしながら、これらのプロトコルは、同期化目的でフレーム同期信号を伝送するために、バス用の別個のワイヤまたはバスラインの使用を必要とする。これは、余分なハードウェア費用および余分な空間を必要とし、余分なワイヤを追加することが可能である、これらの状況のみで使用されることができ。さらに、それらは、例えば、充電器の場合等で、無線媒体または2つの電力端子を経由したデータ伝送に適していない。

【0396】

場合によっては、可能な解決策は、特別な明確に定義された起動シーケンスを使用することであり、全てのスレーブデバイスが、バスを経由して通信することができるように、起動時にバスに接続され、フレーム長およびフレーム形式情報を受信する。しかしながら、この従来のスキームは、デバイスのホットプラグイン(すなわち、バスがアクティブにされた後の以降の時間に、スレーブデバイスがバスに接続する)、およびデバイスのホット除去(すなわち、バスが非アクティブにされる前にスレーブデバイスがバスから接続解除する)とも称される、バスに動的に接続され、かつバスから動的に接続解除されることができ、スレーブデバイスと適合性がない。また、スレーブデバイスの不具合またはバスの不具合の場合、この従来のスキームでは、特別な起動シーケンスを経由するために、バスが非アクティブにされ、次いで、再びアクティブにされない限り、再度スレーブデバ

40

50

イスがバスと再同期化する方法がない。最終的に、リングベースのバス構造は、故障する単一のデバイスに続いて、全てのデバイスの全不具合を受ける傾向があり、リングベース構造では、バスが起動される前に全てのデバイスがアタッチされる。

【0397】

加えて、費用、空間、および電力の理由で、フレーム同期化信号がビットストリームデータに組み込まれたときに、多重化ビットストリームシーケンスに同期化することができる、従来のバス同期化アルゴリズムがない。多重化ビットストリームは、バスに使用されるワイヤまたはラインの数を削減しながら、能動雑音消去、ビーム形成、および低遅延用途のために有用である。唯一の従来の解決策は、費用、面積、および電力消費を増加させる、フレーム同期信号のための余分なワイヤを追加することである。これに加えて、これらの従来の方法は、リセット、制御、およびデバイス監視のための追加のワイヤ、ならびにサポートする論理を必要とする。バスに使用されるラインの数を削減しながらの同期化の問題は、マルチフォーマットまたは統一バス通信プロトコルが使用されているときに、さらに複雑な問題になり、フレーム長およびフレーム形式自体の両方が、動作中に変化し得、これらのパラメータは、同期化プロセスの開始時に不明である。

10

【0398】

ここで、汎用同期化方法の種々の例示的实施形態が説明され、方法は、フレーム長、フレーム形式、およびフレームタイプが、動作中に変化し得、同期化プロセスの開始時に不明である場合、本明細書で説明されるような統一バス通信プロトコルが使用されるときに、スレーブデバイスをバス64に同期化するために使用されることができる。例えば、フレーム形式は、ワードフレーム形式(すなわち、ワードモードで)、ビットストリームフレーム形式、または統一ビットストリームフレーム形式(すなわち、両方ともビットストリームモードで)であり得る。別の側面では、本明細書で説明される汎用同期化方法の種々の例示的实施形態のうち少なくとも1つは、バスに使用されるバスラインまたはワイヤの削減をサポートする。他の側面が、本明細書の汎用同期化方法の種々の例示的实施形態の説明中に明白となるであろう。

20

【0399】

一般に、本明細書で説明される汎用同期化方法の種々の例示的实施形態は、バス64上で伝送されるデータにおいて少なくとも1つの同期化パターンを検索することを伴う。スレーブデバイス54が同期化パターンを見出すと、スレーブデバイスは、バス64上でマスタデバイス52と通信し、フレーム形式、および統一バス通信プロトコルによって使用されている動作モードに従った動作のために構成される。以下でさらに詳細に説明されるように、同期化パターンを見出すための種々の代替案がある。少なくともいくつかの場合において、汎用同期化方法の種々の例示的实施形態は、スレーブデバイス54の同期エンジンによって実装されることができる。以前に説明されたように、同期エンジンは、目の特定の用途のためにより好適であるどんなものにも応じて、プロセッサ、状態マシン、または他の専用回路を使用して実装されることができる。

30

【0400】

本明細書で説明される汎用同期化方法の少なくとも1つの実施形態では、同期化パターンを検索することは、Sワードの一定同期化シンボルまたは一定同期部分を検索することを含む。同期化パターンのビットは、ワードモードのように、バス64を経由して連続的に伝送され得るか、またはビットストリームモードのように、他のデータビットと多重化またはインターレースされながら1度に1ビット伝送され得る。

40

【0401】

本明細書で説明される汎用同期化方法の少なくとも1つの実施形態では、同期化パターンを検索する前に、ワードモード等の2つの動作モードのうちの一つが仮定されることができる。同期化パターンがフレームの最大許容長内で場所を特定されない場合には、他の動作モードを仮定し得、同期化パターンがもう1度検索される。同期化パターンが場所を特定されない場合には、他の動作モードをもう1度仮定し得、同期化パターンが再度検索される。異なるモードの間で前後に交代するこのプロセスは、同期化(すなわち、ロック

50

)が達成されるまで行われることができる。異なるモードに基づく検索間のこの切り替えは、以下でさらに詳細に説明されるように、よりロバストな検索プロセスをもたらす。

【0402】

本明細書で使用され、文脈に従う場合、「仮定する」または「仮定すること」という動作は、人間が行い得るような考え方を採用することを意味しないことに留意されたい。むしろ、「仮定すること」という動作は、機械が行い得るように、動作または挙動すること、あるいは動作または挙動しようとする、または試行することを指し得る。「動作モードを仮定すること」等の語句は、例えば、必ずしも最初に、その動作または挙動に関連し得る、特定の事実または機能性あるいは任意の他の条件を検証すること、または正当性を立証することなく、動作の Protokol または慣例または規格または方法に従って、動作または挙動することを指し得る。この文脈では、「第1の動作モードを仮定すること」は、第1の動作モードが、有効で機能的であり、実践では正しくないことが判明し得る結果を達成する良好な方法であるかのように、口語的に動作することと考えられ得る。したがって、方法の特定の作用が、特定のパラメータの特定の値を仮定すること、または特定の条件を仮定することとして表されるとき、それは、概して、特定のパラメータの初期値または特定の条件が有効であるものとして選択されることを意味する。特定の実施形態に応じて、次いで、この選択された値は、それが正しいかどうかを確認するようにチェックされ得、もしそうでなければ、次いで、例えば、別の値を選択すること等の別の措置が講じられ得る。

10

【0403】

本明細書で説明される汎用同期化方法の少なくとも1つの実施形態では、並行実装を使用することができ、動作モードを仮定する一方の実装は、ワードモードであり、動作モードを仮定する他方の実装は、ビットストリームモードである。次いで、実装は、同期化が達成されるまで、仮定された動作モード下で同時に同期化パターンを検索する。この場合、同期化パターンが、所与の実装について場所を特定されないとき、同一のモードが仮定され得、検索が再度実行される。再度、これは、以下でさらに詳細に説明されるように、よりロバストな検索プロセスをもたらす。

20

【0404】

本明細書で説明される汎用同期化方法の少なくとも1つの実施形態では、同期化パターンの第1のインスタンスの場所が特定されると、使用されているフレーム長を決定するために、同期化パターンの次のインスタンスについて、新しい検索が行われるであろう。いくつかの実施形態では、これは、決定されたフレーム長が正しいことを検証するために、数回行われ得る。フレーム長が決定されると、後続の検索は、決定されたフレーム長を使用して、同期化パターンを検索するであろう。少なくとも1つの実施形態では、バス64上でデータの全フレーム中に同期化パターンが伝送されることができるとに留意されたい。

30

【0405】

本明細書で説明される汎用同期化方法の少なくとも1つの実施形態では、マスタデバイス52とスレーブデバイス54との間の誤った同期化の可能性を低減させるために、同期化パターンは、一定同期部分および動的同期部分を有することができる。動的同期部分を生成するために、決定論的方法が使用されることができ、動的同期部分は、バス64上で伝送される2つの連続ワードの動的部分の間等、2つの連続する同期化パターン間で異なる。最初に一定同期部分の場所が特定されると、対応する現在の動的同期部分(すなわち、現在の動的同期部分)が読み取られる。次いで、決定論的方法が、現在の動的同期部分に基づいて、次の期待動的同期部分を計算するために使用される。次に一定同期部分の場所が特定されると、対応する動的同期部分が読み取られ、誤った同期化パターンに仮定されるように、真の同期化パターンの場所が特定されたことを検証するために、計算された上記の期待動的同期部分と比較される。一定同期部分および動的同期部分の両方に合致する誤った同期化パターンの確率は、単に一定同期部分を検索するより低いので、この場合、大抵の誤った同期化パターンは、無視されるであろう。この検索プロセスは、誤

40

50

ために使用され得る。他の実施形態では、既知の状態で起動され得、バスは、誤った同期パターンが存在しないが、この実施形態は、バスが起動した後に別のデバイスが同期化を取得することを可能にしないであろう。したがって、このスキームでは、新しいデバイスがバスに接続しようとする場合に、バスが再起動する必要があるであろう。したがって、この解決策が選択される場合、いかなるオーディオ妨害も伴わないホットプラグインが可能ではない。

【0410】

2つの一定同期部分の合致に加えて、動的同期部分に対する合致があることをチェックすることに加え、少なくともいくつかの実施形態では、2つの一定同期パターン間の距離が一定のままであるかどうかをチェックすることも可能である。これは、有効フレーム形式に当てはまる。したがって、有効な動的同期部分を伴う2つの同期化パターン間の距離もまた、一定の距離（例えば、2つの同期化パターンの開始または終了間のビットの数で）を有することをチェックすることが可能である。そのようなチェックを行うために、比較されることができ2つの距離を計算することができるよう、少なくとも3つの有効な一定同期部分が場所を特定される。

10

【0411】

本明細書で説明される同期化方法の少なくとも1つの実施形態では、一定同期部分の場所が特定されているが、1つ以上のエラーがあるとき（ちょうど見出された同期化パターンに基づいて決定される現在のフレーム長が、以前に場所が特定された同期化パターンに基づいて決定された以前のフレーム長と同一ではない等）、ランダム成分が使用されることができ、無効な同期位置を破棄するために、この追加の情報が使用されることができ

20

【0412】

本明細書で説明される汎用同期化方法の少なくとも1つの実施形態では、ビットストリームモードで同期化パターンを検索する汎用同期化方法の部分は、同期化パターンのビットが存在しないと仮定される、チャンネルからのデータをスキップするために、クロックゲーティングを利用することができ（例えば、1つおきのビットのみからの情報を使用して、ロックがすでに達成されている場合に、1つおきのビットを無視する）。クロックゲーティングを使用することの1つの利益は、それが電力消費の削減を可能にし、それによって、同期化方法をよりエネルギー効率的にするであろうことである。なぜなら、いくつかのフレームチャンネルが同期化パターンのビットを有すると仮定されないため処理されないからである。クロックゲーティングを使用することの別の利益は、以下でさらに詳細に説明されるように、クロックゲーティングが、本質的に、ビットストリームフレームデータがワードデータと同様に処理されることを可能にするので、共通検索構造が、動作のワードおよびビットストリームモードの両方の下で同期化パターンを検索するために使用されることができることである。

30

【0413】

一般に、本明細書で説明される種々の汎用同期化方法は、同期化までの時間を短縮するために、上限を採用するであろう。例えば、バス64がワードモードで動作し、同期化パターンを検索していると仮定するとき、上限をフレーム長に課することができる。したがって、同期化パターンの場所を特定するたように検索されるビット数が、最大定義フレーム長より長く、同期化パターンが見出されていない場合、検索は、ビットストリームモードに切り替わり、同期化エンジンは、同期シンボルパターンが他の情報と多重化されていることを仮定し、同期化を達成するために、正しいチャンネル位置、チャンネル内の位置、およびチャンネルの数を検索するであろう。代替として、少なくとも1つの他の実施形態では、検索方法は、ビットストリームモードに切り替わる前に、ワードモードで1回以上、再度試行し得る。

40

【0414】

少なくとも1つの実施形態では、バスがビットストリームモードで動作していると仮定するとき、同期化パターンを検索する前に、フレーム形式でのフレームチャンネルの数に上

50

限を課すことができる。例えば、フレームチャネルの最大数が A_{max} であると仮定される場合には、本明細書で説明される同期化方法のための検索時間は、 A_{max}^3 にほぼ比例する。 A_{max} を 16 以下に設定することが、同期化を取得するための許容可能な時間量をもたらすことが分かっている。しかしながら、同期化を取得するための時間が容認可能である、より高いクロック周波数、より高い処理能力、または両方の条件下で、 A_{max} のより高い値を使用することが可能であり得る。全ての可能な数のビットストリームチャネル、およびこれらのチャネルの各々の内側の同期化位置を試行した後に、同期化が取得されていない場合、同期化方法は、再度、ワードモードに戻り、検索プロセスを再び新たに始め得る。これは、マスタデバイスが同期化パターンを伝送する準備がまだできていない、またはマスタデバイスが何らかの機能不全により、または節電目的でアイドル状態であるため、マスタデバイスがバスにアタッチされる前にスレーブデバイスがバスにアタッチされ、スレーブデバイスがクロック信号を受信しているが、いかなる同期化パターンも受信していない場合に有益であり得る。

10

【0415】

本明細書で説明される種々の同期化方法の少なくとも1つの実施形態では、例えば、クロック信号上の故障により、スレーブデバイス54がマスタデバイス52とのロックから外れる（すなわち、同期化から外れる）場合、スレーブデバイス54は、完全でより長い同期化検索を回避するために、以前に決定されたフレーム長および同一のフレーム形式を仮定して、同期化を再獲得しようとすることができる。これが成功しない場合、スレーブデバイス54は、この同一の仮定を使用して、同期化を数回再試行することができる。これが依然として成功しない場合には、スレーブデバイス54は、マスタデバイス52が、バス64上で通信するために異なるフレーム形式または動作モードを使用することに切り替わっていると仮定することができ、次いで、スレーブデバイスがマスタデバイスと同期化されることを可能にするために、完全同期化検索が使用されることができ

20

【0416】

代替実施形態では、本明細書で説明される汎用同期化方法の種々の例示的实施形態のステップが、異なる順番で実装され得ることに留意されたい。例えば、最初にワードモードを仮定して同期化パターンの検索を実行するよりもむしろ、最初にビットストリームモードを仮定し得る。

【0417】

本明細書で説明される種々の同期化方法のうちの少なくとも1つの利益は、場合によっては、6MHzクロックレートを仮定して、約60ミリ秒以下等で極めて迅速に同期化を達成できることである。平均ロック時間は、この値よりはるかに速くあり得る。いくつかの実施形態では、検索時間を減少させるために、チャネルおよびフレーム長の数の全ての組み合わせが可能にされるわけではない。

30

【0418】

一側面では、本明細書で説明される汎用同期化方法の種々の例示的实施形態のうちの少なくとも1つの利益は、動作中のバス64へのスレーブデバイスのホットプラグインおよびホット除去に対するサポートである。したがって、スレーブデバイス54は、フレーム形式および動作モードを独力で決定し、それがアタッチされ、バス通信プロトコルに従って通信する準備ができていることをマスタデバイス52に伝えることができる。これは、スレーブデバイス54が、動作モード、フレーム形式、サブフレーム長、およびフレームの開始位置を決定することを伴うことができる。

40

【0419】

一側面では、本明細書で説明される汎用同期化方法の種々の例示的实施形態のうちの少なくとも1つの利益は、スレーブデバイス54、バス64、およびこれらのシステム構成要素の間の物理的リンクを表すハードウェアが、物理的に破損されない限り、スレーブデバイス54が起動する条件、またはバス64の初期または以降の条件にかかわらず、スレーブデバイス54がバス64と同期化し、バス通信プロトコルに従って通信できることである。これは、バス上で発生する雑音の場合、ならびに内部レジスタ状態を変化させるア

50

ルファ粒子、またはデバイス自体の制御を超える内部状態の他の変化等のランダムなイベントによるデバイスの機能不全の場合に、同期化のための増加したロバスト性を提供する。さらに、デバイスが不良な実装により故障する場合、この故障後に、このデバイスへのアクセスを得ることが依然として可能である。なぜなら、内部同期化エンジンが、（同期化の欠如、または繰り返されるバス衝突のいずれかによる）エラー後にデバイスがバス64からバックオフすることを確実にし、次いで、同期化に再度戻ろうとするからである。マスタデバイス52からの助けを借りずに、スレーブデバイス54がそれ自体をバス64にアタッチすることができるという事実が、本システムをよりロバストにする。

【0420】

一側面では、本明細書で説明される汎用同期化方法の種々の例示的实施形態のうちの少なくとも1つの利益は、バスの動作中にフレーム形式パラメータおよび/または動作モードが変更されることができ、スレーブデバイス54がまだバス64にアタッチされていない場合に依然として同期化を達成できることである。

【0421】

汎用同期化方法の種々の実施形態の以下の説明では、説明される同期化パターンは、Sワードであり得、（一定同期部分としても知られている）同期定数は、SワードのCONSTANT SYNC SYMBOLフィールド中のデータであり得、動的同期部分は、SワードのDYNAMIC SYNC SYMBOLフィールド中のデータであり得ることを理解されたい。代替として、同期化パターンが、CONSTANT SYNC SYMBOLフィールドからのデータ、DYNAMIC SYNC SYMBOLフィールドからのデータ、またはCONSTANT SYNC SYMBOLフィールドおよびDYNAMIC SYNC SYMBOLフィールドからのデータの組み合わせを備え得る、実施形態があり得る。他の実施形態では、一定同期シンボルまたは動的同期シンボルは、同期化目的で、単独で、または他の組み合わせで使用されることができ、例えば、フレーム内で1回以上使用されることができ、

【0422】

同期化は、いくつかの異なる方法で達成されることができ、ここで説明される実施形態は、低ゲート総数で実装されることができ、ここに留意されたい。代替として、より高いゲート総数を使用することによって、より高速の同期化スキームが実装されることができ

【0423】

ここで図54を参照すると、汎用同期化方法900の例示的实施形態の略図が示されている。汎用同期化方法900の側面のうちの1つは、バス64を経由して通信するために使用される統一バス通信プロトコルに使用されている動作モードおよびフレーム形式にかかわらず、バス64がアクティブにされた後に、スレーブデバイス54がマスタデバイス52に同期化するための能力である。

【0424】

902では、方法900が、統一バス通信プロトコルに対する第1の動作モードを仮定する。したがって、方法900は、正しい動作モードがワード形式またはビットストリーム形式のいずれかであることを仮定し、次いで、この仮定に基づいて、続いて同期化パターンの場所を特定することに進む。本実施例では、方法900は、最初に、動作のワードモードを仮定する。

【0425】

904では、方法900が、第1の動作モードに従って、バス64を経由して伝送されたデータ中の1つ以上の場所で同期化パターンを検索する。同期化パターンが、一定同期部分および動的同期部分を備える場合、作用904は、一定同期部分を検索することを含む。少なくとも1つの実施形態では、一定同期部分は、第1または第2の動作モードについて異なり得る。代替として、別の実施形態では、同期化パターンが固定部分および動的部分を備える場合には、作用904は、動的同期部分を検索することを含む。

【0426】

10

20

30

40

50

代替として、少なくとも1つの実施形態では、方法900は、動的同期部分が決定論的方法によって決定されることができ、同期化パターンが一定同期部分を備えるかどうかに関わらず、同期化パターン全体を検索することができる。少なくとも1つの実施形態では、動的同期部分は、2つの連続する同期化パターンの動的同期部分が異なるが決定可能であるように、決定論的方法によって生成されることができ、少なくとも1つの実施形態では、決定論的方法は、CRCジェネレータ、または以前に説明された変形例のうちの1つを採用し得る。

【0427】

少なくとも1つの実施形態では、使用されるいくつかの同期化パターンがあり得、作用904は、伝送されたデータをこれらの同期化パターンの各々と比較することによって、これらの同期化パターンのうちの1つに対して検索することを含むことができる。

10

【0428】

少なくとも1つの実施形態では、異なる同期化パターンは、定義された順番で起こることができ、方法900は、定義された順番を使用して、異なる同期化パターンを検索することができる。

【0429】

904では、ワードモードが仮定されるとき、方法900は、同期化パターンの開始タイムスロットとして、伝送されたデータに対するあるタイムスロットを選ぶことを含む。次いで、長さが、検索されている同期化パターンと同一の長さであるように、開始タイムスロットから終了タイムスロットまでの伝送されたデータが、バストラフィックから読み取られ、検索されている同期化パターンと比較される。この例示的实施形態では、検索されている同期化パターンは、一定同期部分である。他の代替実施形態では、異なるデータ長が読み取られ、同期化パターン全体、動的同期部分のみ、または一定同期部分および動的同期部分の両方を含む、同期化パターンの異なる部分と比較されることができ、いかなる合致もない場合には、開始タイムスロットを1つのタイムスロットだけ前進させ、プロセスを繰り返すことができる。並行実装では、並行して、いくつかの開始タイムスロットを選び、バストラフィックからデータを読み取り、読み取られたデータを(ちょうど説明されたような)同期化パターンの一部分または全体と比較することが可能であり得る。

20

【0430】

904では、ビットストリームモードが仮定されるとき、方法900は、同期化パターンの開始タイムスロットとして、伝送されたデータに対するあるタイムスロットを選ぶことを含む。次に、ビットストリームフレームチャネルの数が仮定される。次いで、ビットストリームフレームチャネルの仮定された数と各フレームチャネルに対するデータ長とに基づいて、フレーム長を計算することができる。例えば、制御フレームチャネルがSワードを備え、その後、各ワードが16ビットである、XワードおよびYワードが続く、以前に説明されたビットストリームフレーム形式を使用する場合、各フレームチャネルに対するデータ長は、48ビットであり、フレーム長は、ビット単位で $48 * (\text{仮定ビットストリームフレームチャネルの数})$ として計算される。いくつかの実施形態では、S、X、およびYワードの間にいくらかの間隔があり得る。ビットストリームフレームチャネルの最大数が定義され、検索は、伝送されたデータを反復して検索することを含む。これは、最初に、多重化されるチャネルのある数を仮定することによって行われる。次いで、有効な同期化パターンが取得されるか、または取得されないまで、全ての有効な位置でこれらのチャネルの各々を検索することが可能である。いかなる同期化パターンも見出されなかった場合、次のチャネルが試行され、全てのチャネルが試行された場合、多重化されると仮定されたチャネルの数が増加させられ、同一の手順が繰り返されるであろう。このプロセスは、バス64に対して定義されるチャネルの最大数を超えるまで継続され得、その場合、本プロセスは、再び新たに始まるか、またはワードモードで再度始まるかのいずれかであり得る。この検索は、1つのビットストリームフレームチャネルを仮定することによって始まり得る。本説明のこの部分では、ビットストリームフレームチャネルという用語は、制御フレームチャネル、ビットストリームフレームチャネル、仮想フレームチャ

30

40

50

ネル、および多重化フレームチャネルを対象とするか、または含むように意図されていることに留意されたい。

【0431】

例えば、ビットストリームフレームチャネルの最大数が3であると仮定すると、1つのビットストリームフレームチャネルがあると仮定することによって、検索が始まり、これは、(制御ワードの間の空間も許容されない限り)フレームがいかなるデータも伴わないフレーム制御チャネルのみを備える場合であろう。この場合、同期化パターンのビットは、ビットごとに発生する。以前に説明されたSワード、Xワード、およびYワードスキームを仮定すると、この場合にフレームに対して伝送されるビットは、Sワードビット15、Sワードビット14、・・・、Sワードビット0、Xワードビット15、Xワードビット14、・・・、Xワードビット0、Yワードビット15、Yワードビット14、・・・、およびYワードビット0を備えるはずである(制御ワードの数、ビット数、ならびにこれらのビットおよびワードの順序付けが、代替実施形態では異なり得ることに留意されたい)。開始タイムスロット位置から始まり、同期化パターンに合致するかどうかを決定するために、次のM個の伝送されたデータビットが読み取られ、Mは、検索されている同期化パターンのビット数である。いかなる合致もない場合には、開始タイムスロット位置が、1位置、前方に移動させられ、初期の最初のビットが破棄され、次の伝送されたビットが読み取られ、Mビット比較が再度繰り返される。いかなる合致もない場合、開始タイムスロット位置を移動させ、比較を行うこのプロセスは、同期化パターンの場所が特定されるか、または最大フレームチャネル長に達するまで繰り返される。同期化パターンが見出されていない場合には、2つのビットストリームフレームチャネルがあると仮定して、この検索プロセスが繰り返される。1つのビットストリームフレームチャネルに対する検索プロセスは、ワードモード下で行われる検索と同一であることに留意されたい。代替的な並行実施形態では、同期化パターンとの比較のために使用される、伝送されたデータを読み取って通過するために、FIFO構造を使用することができるが、通常は、シリアルシフトレジスタが比較のために使用され得る。

【0432】

ここで、フレーム形式で2つのビットストリームフレームチャネルがあると仮定すると、この場合、同期化パターンのビットが2ビットごとまたは2つのタイムスロットごとに発生するため、方法900は、伝送されたデータの2ビットごとに検索する。以前に説明されたSワード、Xワード、およびYワードスキームを仮定すると、この場合、フレームに対して伝送されるビットは、Sワードビット15、データチャネル1ビット0、Sワードビット14、データチャネル1ビット1、・・・、Yワードビット0、およびデータチャネル1ビット47を備えているはずである。もう1度、開始タイムスロット位置から始まり、同期化パターンに合致するかどうかを決定するために、次のM個の伝送されたデータビット(各々1ビットによって分離される)が読み取られる。いかなる合致もない場合には、開始タイムスロット位置が、2位置、前方に移動させられ、次のM個の伝送されたデータビットが読み取られ、比較が繰り返される。これは、ワードモードで合致を見出すために使用されるアルゴリズムが再利用されるが、この時、1ビットおきのみアクティブであるようにクロックゲート制御(またはクロック有効化)される場合、ハードウェアにおいて的確に行われることができる。このようにして、1ビットおきにスキップすることは、次の位置で検索し、1つおきの位置に対してクロックゲーティングを追加することと同等である。これは、2ビットをスキップすることと同等である。したがって、チャネルの数がNであると仮定され、N-1個の位置でクロックを無効にすることによって、N-1個のチャネルをスキップし、調査される次のビットが、調査された最後の位置の後のN個目の位置であるビットである、単純な統一アルゴリズムが作成されることができ。いかなる合致もない場合には、開始タイムスロット位置を移動させ、比較を行うこのプロセスは、同期化パターンの場所が特定されるか、または最大フレーム長に達するまで繰り返される。最大フレーム長は、ビットストリームフレームチャネルの仮定された数で乗算される最大チャネル長である。同期化パターンが見出されていない場合には、2つのビッ

10

20

30

40

50

トストリームフレームチャンネルがあると依然として仮定するが、1ビットだけ開始位置を遅延させて、この検索プロセスが繰り返される。この遅延は、単一のクロックサイクルをスキップし、それによって、全ての検索を次のフレームチャンネルまで移動させることによって行われることができる。前述の検索プロセスは、同期化パターンが見出されるか、または最大フレームチャンネル長に達するまで繰り返される。同期化パターンが見出されていない場合には、3つのビットストリームフレームチャンネルがあると仮定して、この検索プロセスが繰り返される。

【0433】

ここで、フレーム形式で3つのビットストリームフレームチャンネルがあると仮定すると、この場合、同期化パターンのビットが3ビットごとまたは3つのタイムスロットごとに発生するため、方法900は、伝送されたデータの3ビットごとに検索する。以前に説明されたSワード、Xワード、およびYワードスキームを仮定すると、この場合、フレームに対して伝送されるビットは、Sワードビット15、データチャンネル1ビット0、データチャンネル2ビット1、Sワードビット14、データチャンネル1ビット1、データチャンネル2ビット1、・・・、Yワードビット0、データチャンネル1ビット47、およびデータチャンネル2ビット47を備えているはずである。もう1度、開始タイムスロット位置から始まり、同期化パターンに合致するかどうかを決定するために、次のM個の伝送されたデータビット（それぞれ、2ビット、または1を引いた仮定ビットストリームフレームチャンネルの数によって分離される）が読み取られる。いかなる合致もない場合には、開始タイムスロット位置が、3位置、前方に（または前方に仮定ビットストリームフレームチャンネルの数で）移動させられる。この場合に前方に移動するために、少なくとも1つの実施形態では、ワードモード検索アルゴリズムへのクロック信号は、典型的には、2つのクロック周期にわたって無効にされるか、またはゲート制御され得、それによって、次のクロック周期を次のビットに等しくする（それによって、同一のアルゴリズムを再利用する）。次いで、第1の伝送されたビットが破棄され、次いで、伝送されたデータビットが読み取られ、比較が繰り返される。いかなる合致もない場合には、開始タイムスロット位置を移動させ、比較を行うこのプロセスは、同期化パターンの場所を特定されるか、または最大フレーム長に達するまで繰り返される。同期化パターンが見出されていない場合には、3つのビットストリームフレームチャンネルがあると依然として仮定するが、1ビットだけ開始位置を遅延させて、この検索プロセスが繰り返され、1クロックサイクルをスキップすることによって再度行われることができる。前述の検索プロセスは、同期化パターンが見出されるか、または最大フレームチャンネル長に達するまで繰り返される。同期化パターンが見出されていない場合には、3つのビットストリームフレームチャンネルがあると仮定するが、1ビットだけ開始位置を遅延させて、この検索プロセスが繰り返される。前述の検索プロセスは、同期化パターンが見出されるか、または最大フレームチャンネル長に達するまで繰り返される。本実施例では、3つのチャンネル間隔を使用して、本方法がフレームチャンネルの全てを通して検索されているため、同期化が取得されている場合がある。しかしながら、同期化パターンが見出されておらず、バスに許容されたチャンネルの最大数を超えている場合、モードはワードモードに変更され、または単一のチャンネルを再度仮定して、プロセス全体が再び新たに開始される（これは、マスタデバイス52がいかなる同期化情報も送信していなかった場合に有益であり得る）。

【0434】

一般に、ビットストリームモードの仮定の下で検索するとき、検索は、最大A個のビットストリームフレームチャンネル、Qビットのチャンネル長を仮定すること、および同期化パターンを検索するためにデータの数回の経由を潜在的に行うこと（並行ハードウェアが存在する場合、単一の経由が使用され得る）を含む。フレームチャンネルの現在の数Cが設定され、現在のフレームチャンネル指数Bが1に設定される。TS0の開始タイムスロットが定義され、B番目の経由の下で、同期化パターンは、B番目のビットストリームフレームチャンネル中にあると仮定される。例えば、Aが4であると仮定すると、次いで、Cが1に設定され、Bが1に設定され、第1のビットストリームチャンネルが検索される。同期化パ

10

20

30

40

50

ターンが見出されない場合には、2つのビットストリームフレームチャンネルがあると仮定され、Cが2に設定され、Bが1に設定され、開始タイムスロットが1だけ遅延させられ、第1のビットストリームチャンネルが検索される。同期化パターンが見出されない場合には、Bが2に設定され、1つのタイムスロットをスキップし、第2のチャンネル中で開始することによって、検索が第2のビットストリームフレームチャンネルについて繰り返される。同期化パターンが見出されない場合には、3つのビットストリームフレームチャンネルがあると仮定され、Cが3に設定され、Bが1に設定され、開始タイムスロットが再度1だけ遅延させられ、第1のビットストリームチャンネルが検索される。同期化パターンが見出されない場合には、Bが2に設定され、検索が第2のビットストリームフレームチャンネルについて繰り返される。同期化パターンが見出されない場合には、Bが3に設定され、検索が第3のビットストリームフレームチャンネルについて繰り返されることが仮定される。同期化パターンが見出されない場合には、4つのビットストリームフレームチャンネルがあると仮定され、Cが4に設定され、Bが1に設定され、開始タイムスロットが再度1だけ遅延させられ、第1のビットストリームチャンネルが検索される。同期化パターンが見出されない場合には、Bが2に設定され、検索が第2のビットストリームフレームチャンネルについて繰り返される。同期化パターンが見出されない場合には、Bが3に設定され、検索が第3のビットストリームフレームチャンネルについて繰り返される。同期化パターンが見出されない場合には、Bが4に設定され、検索が第4のビットストリームフレームチャンネルについて繰り返される。同期化パターンが見出されない場合には、 $B = C = A$ であるため、ビットストリームフレームチャンネルの最大数が検索されており、方法900は908に移動する。代替として、方法900は、同期化パターンの場所が特定されると、上記のアクションの全てを経由する必要はない。

【0435】

異なるデータ長を読み取り、同期化パターンの異なる部分と比較することができるワードモードについて前述されたように、ビットストリームモードを検索するための種々の代替実施形態があり得る。加えて、並行して、いくつかの開始タイムスロットを選び、バストラフィックからデータを読み取り、読み取られたデータを同期化パターンの一部分または全体と比較することが可能であり得る並列実装が使用される実施形態があり得る。実施例として、ビットストリームの数が $2N$ に限定される場合、同期化パターンの場所が特定されるか、またはビットストリームフレームチャンネルの最大数が検索されるまで、最初に単一のチャンネルを読み取り、これが完了したときに、並行して（各々1つのスロットによってオフセットされる）2つのチャンネルを読み取り、これが完了したときに、並行して3つのチャンネルを読み取ること等によって、検索時間を約 $(N + 1)$ 倍短縮することが可能である。

【0436】

ワードモードおよびビットストリームモードの下で検索することの間の類似性により、少なくとも1つの実施形態では、ビットストリームモードの下で検索するときクロックゲーティングを使用することによって、動作のワードモードおよびビットストリームモードの両方の下で同期化パターンを検索するための共通検索構造（すなわち、コンピュータコードまたは状態マシン等の論理構造）を使用することが可能である。クロックゲーティングは、同期化パターンのビットが存在すると仮定されないフレームチャンネルからのデータをスキップするために使用される。例えば、3つのビットストリームフレームチャンネルがあると仮定すると、TS0の開始タイムスロットを仮定して、第1の経路の下で、同期化パターンは、タイムスロットTS0、TS3、TS6等で伝送されるビットを有する、第1のビットストリームフレームチャンネル中に存在すると仮定される。したがって、クロックゲーティングは、これらのタイムスロットで伝送されるデータを選び、タイムスロットTS1、TS2、TS4、TS5等で伝送されるデータを無視するために使用されることができる。同期化パターンが第1の経路中に見出されなかった場合、始点が1つのタイムスロットだけ遅延させられ、検索が繰り返される。したがって、同期化パターンは、タイムスロットTS1、TS4、TS7等で伝送されるビットを有する、第2のビットスト

10

20

30

40

50

リームフレームチャンネル中に存在すると仮定される。したがって、クロックゲーティングは、これらのタイムスロットで伝送されるデータを選び、タイムスロットTS0、TS2、TS3、TS5等で伝送されるデータを無視するために使用されることができる。同期化パターンが第2の経路中に見出されなかった場合、始点が1つのタイムスロットだけ遅延させられ、検索が繰り返される。したがって、同期化パターンは、タイムスロットTS2、TS5、TS8等で伝送されるビットを有する、第3のビットストリームフレームチャンネル中に存在すると仮定される。したがって、クロックゲーティングは、これらのタイムスロットで伝送されるデータを選び、タイムスロットTS0、TS1、TS3、TS4等で伝送されるデータを無視するために使用されることができる。

【0437】

いくつかの実施形態では、ビットストリームモードで並列実装を使用することによって、検索時間を短縮することが可能であり得る。例えば、一度に1つのビットストリームフレーム形式を仮定して、伝送されたデータを検索する代わりに、同時に複数のビットストリームフレーム形式を仮定して、伝送されたデータを検索することが可能であり得る。これは、より高いゲート総数および複雑性をもたらすが、同期化を達成する短縮された時間をもたらすであろう。

【0438】

906では、説明されるような特定の実施形態に依存する、同期化パターンが見出された場合、方法900は910へ移動する。しかしながら、同期化パターンが見出されていない場合、方法900は908へ移動する。動作モードとしてワードモードを仮定するとき、同期化パターンの場所が早期に特定されていない限り、最大フレーム長までを含む種々の開始タイムスロット位置に対して、伝送されたデータで検索作用が行われる。動作モードとしてビットストリームモードを仮定するとき、最大フレーム長までを含む、種々の開始タイムスロット位置に対して、伝送されたデータで検索作用が行われ、これは、同期化パターンの場所が早期に特定されていない限り、フレームチャンネルの最大数までの任意の数のフレームチャンネルについて繰り返される。いくつかの実施形態では、同期化を達成する時間を短縮するために、フレーム長およびチャンネル数の限定された一部のみを検索することが選択され得る。

【0439】

ビットストリームモードまたはワードモードでは、検索は、新しい伝送されたデータ上で行われることができるか、または伝送されたデータの一部分が保存され、同期化パターンが見出されるか、または検索限界に達するまで、検索が保存されたデータ上で反復されることができる。これら2つの実装の間の選択は、スレーブデバイス54が同期化パターンの検索に専念させることができる、メモリの量に依存する。例えば、1つだけのバッファが検索のために使用されることができる場合、説明された検索の第1の方法が使用される。この第1の方法の利点は、より小さいチップサイズ、より低い費用、およびより単純な実装である。

【0440】

908では、第1の動作モード(すなわち、ワードモード)を仮定して同期化が取得されていないため、統一バス通信プロトコルのために第2の動作モード(すなわち、ビットストリームモード)が仮定される。次いで、方法900は、第2の動作モードに従って、伝送されたデータ上で、検索する作用904および906、ならびに取得する作用910を行う。より一般的に、作用908は、他のフレーム形式に切り替わることを含む。したがって、検索する作用904および906がビットストリームモードの下で成功していない場合には、908で、もう1度ワードモードを仮定し、同期化パターンが904および906で再度検索される。異なるフレーム形式の間のこの切り替えは、同期化が取得されるまで繰り返される。通常の条件下で、スレーブデバイスは、異なる状態であるようにプログラムされるまで、この検索を行い続けるであろう。たとえすでに1回完了していたとしても、スレーブデバイスが検索を再度行い続けるべきである理由は、マスタデバイス52が、何らかの機能不全により一時的に機能停止し得るか、またはスレーブデバイスより

10

20

30

40

50

も電源を入れることが遅くあり得ることである。したがって、ロバスト性のために、スレーブデバイスは、単純に検索を継続するべきである。検索を停止することが所望される場合、マスタデバイス52は、随意に、バスクロックをゲート制御し、それによって、スレーブデバイスが同期化の検索を繰り返し行うことを停止させることができる。

【0441】

910では、方法900は、使用される特定の実施形態に依存する、少なくとも1つの同期化規則に従って、場所を特定された同期化パターンが検証された場合に同期化を取得することを含む。一般に、少なくとも1つの同期化規則は、910で検証を1回以上行うことを伴うことができる。

【0442】

例えば、少なくとも1つの実施形態では、同期化パターンの場所が特定されると、検索する作用904および906、ならびに取得する作用910は、統一バス通信プロトコルによって使用されているフレーム長を決定するために、次の同期化パターン（すなわち、連続同期化パターン）を検索することを含む。次いで、同期化規則は、決定されたフレーム長が正しいことを検証するために、検索する作用904および906、ならびに取得する作用910を数回行うことを含むことができる。例えば、2つのフレーム長が計算されるように、3つの連続同期化パターンが見出される場合には、2つのフレーム長が異なる場合にエラーがある。代替として、2つのフレーム長が同一である場合には、これは、場合によっては検証として使用され得る。フレーム長が見出されると、フレーム長を3で除算することによって、例えば、2進分割によって、またはこのカウンタがフレーム長に等しくなる（次いで、増分の数がサブフレーム長に等しくなる）までカウンタを3だけ増分することによって、コマンドワードの間の間隔を見出すことができる。

【0443】

少なくとも1つの実施形態では、検証は、同期化パターンの動的部分を使用することを含む。例えば、検索する作用904および906、ならびに取得する作用910は、現在場所を特定されている同期化パターンに対する一定同期部分に関連付けられる現在の動的同期部分を読み取ることを含み得、同期化規則は、決定論的方法および現在の動的同期部分に基づいて、期待動的同期部分を計算することと、次の一定同期部分の場所を特定することと、誤った同期化の可能性を低減させるために、次の一定同期部分に関連付けられた動的同期部分を期待動的同期部分と比較することとを含む。

【0444】

期待動的同期部分が次の動的同期部分に等しくない場合には、2つの同期化パターン（すなわち、現在および以前の同期化パターン）のうちの少なくとも1つが不正確であり得る。少なくとも1つの実施形態では、期待動的同期部分が次の動的同期部分に等しくない場合には、現在場所が特定されている同期化パターンは無効であると仮定されることができる。少なくとも1つの代替実施形態では、期待動的同期部分が次の動的同期部分に等しくない場合には、以前に場所が特定された同期化パターンは無効であると仮定されることができる。これらの選択のうちのいずれかの間で選択することは、以下でさらに詳細に説明されるように、決定プロセスにランダム成分を含むこと等の追加の情報に基づいて行われることができる。

【0445】

少なくとも1つの代替実施形態では、同期化規則は、方法900に、計算された動的同期部分に合致する動的同期部分を有する同期化パターンに対して検索し、この場所が特定された同期化パターンが真のパターンであると見なし続けさせることを含むことができる。計算された動的同期部分に合致する動的同期部分を有するかどうかを決定するために、検索される同期化パターンの数に制限（FL）を課すことができる。この制限が到達された場合、方法950は、最後のFLの場所が特定された同期化パターンのうちのいずれかを真の同期化パターンとしてランダムに選択するか、または次に来る同期化パターンのうちの1つを真の同期化パターンとしてランダムに選択し、次のFLの場所が特定された同

10

20

30

40

50

期化パターンを見ることによって、更新された動的同期部分に基づいて次の同期化パターンを検索し始めることができる。

【0446】

少なくとも1つの実施形態では、決定論的方法は、連続同期化パターンがCRCの連続状態に基づく動的部分を有するように、15個の状態を定義する4ビットを有するCRCジェネレータ(すなわち、LRSを利用するフィードバックを伴うシフトレジスタ、すなわち、線形反復シーケンス)を使用することを含み得る。1つの状態が現在の動的部分のために配置されている場合、決定論的方法を使用して、次の動的部分のための次の状態が計算されることができる。少なくとも1つの代替実施形態では、異なるビット数を伴うCRCを使用することができる。少なくとも1つの代替実施形態では、限定されないが、8、16、または異なる数のビットを利用するLRSTポロジーに基づくCRCジェネレータ等の他の決定論的方法を使用することができる。別の実施例として、単純な加算器を使用することができるが、カウンタが1だけ増分された場合に、全反復中に変化するビット数が限定されるであろうという不利点を有するであろう。一般的な実施形態では、ジェネレータ $X_{n+1} = F(X_n)$ を使用して、一連の値が求められ得、ジェネレータ関数Fは、値 X_n から異なる値 X_{n+1} までのマッピングをもたらし、これは、同時に全ステップで可能な限り多くのビットをランダムに変化させながら、所与の間隔内の可能な限り多くの値が利用されるように、数学関数を利用して行われ得る。これを達成する候補実施例は、閉鎖間隔にわたって一様に分布するランダムな離散数字を生成するように書かれる、数学アルゴリズムである。これらのランダムな数字は、全てではないがほとんどの無効な同期位置を排除し、それによって、同期化プロセスを加速することを目的としている、単一または有限数の同期定数に加えて、バス64上で伝送され得る。

【0447】

例えば、CRCジェネレータが、動的同期部分を計算するために、以下の演算、すなわち、 $\{S_4, S_3, S_2, S_1\} = \{S_3, S_2, S_1, (S_4 \text{ XOR } S_3)\}$ を行うと仮定されたい。本実施例では、動的同期部分は、リセット時に「1111」に初期化されることができる。少なくとも1つの実施形態では、「0000」状態は使用されず、それに遭遇した場合には、次のサイクル中に「1111」状態をもたらすものとする。この理由は、CRCカウンタがLRS構造を使用する少量の論理とともに実装される場合、CRCカウンタがエラーによってゼロ状態に遭遇すれば、この状態から抜け出せないであろうということである。したがって、「0000」というこの値が、ゼロとは異なる少なくとも1ビットを伴う値に自動的にマップされない限り、(少なくとも伝送機については)回復するために苦勞し得る。同様に、4とは異なるビット数がCRCジェネレータで使用される場合、ジェネレータをとどまらせるであろう値(典型的にはゼロ)をチェックし、これらを別の値にマップする必要がある。次いで、この実施例に対する16進数でのシーケンス全体は、 $\{F, E, C, 8, 1, 2, 4, 9, 3, 6, D, A, 5, B, 7\}$ となるであろう。疑似ランダム値である、動的同期部分は、より迅速なロック時間を可能にし、スレーブデバイス54が、ランダムバストラフィックによって生成され得る、誤った同期化パターンにロックする可能性を低減させる。

【0448】

一般に、検証は、同期化プロセスをエラーに対してよりロバストにするように、数回行われ得る。例えば、同期化規則は、誤った同期化の可能性をさらに低減させるために、種々の実施形態の計算する作用、場所を特定する作用、比較する作用を1回以上繰り返すことを含み得る。別の実施例として、CRCジェネレータを使用した検証は、2回、7回、または8回、あるいは例えば、CRCジェネレータによって生成することができる状態の数まで行われ得る。少なくとも1つの実施形態では、スレーブデバイス54がマスタデバイス52と連続的にロックしていることを確実にするために、スレーブデバイス52は、同期化が確立された後の全フレーム中、CRCチェックを行い、同期一定部分の発生をチェックし続け得る。

【0449】

10

20

30

40

50

少なくとも1つの実施形態では、同期化規則は、バストラフィック上の雑音が多い条件下で、およびバストラフィックが同期化パターンに密接に類似するときに、同期化の可能性を向上させるために、場所を特定する作用904および906、ならびに取得する作用910中でランダム成分を追加することを含み得る。

【0450】

例えば、少なくとも1つの実施形態では、ランダム成分は、ある期間にわたってバストラフィックのパリティを読み取ることと、現在場所が特定されている同期化パターンが真のパターンであるか、または誤ったパターンであるかどうかに関してランダムな選択を行うためにパリティ情報を使用することを含む。このようにして、同期化方法は、例えば、1つおきの場所が特定された一定同期化パターンが有効であり、他の1つおきのものが無効であり、それによって、全てのCRCチェックを無効にする場合に、サイクルから抜け出せなくなるであろう。少なくとも1つの実施形態では、期間は、伝送されたデータの1つのフレームであり得る。

10

【0451】

別の実施例として、少なくとも1つの実施形態では、同期化パターンが間違った場所（すなわち、フレームの始めではない）で場所を特定され、それによって、同期化プロセス中に1つ以上のエラーをもたらすときに、ランダム成分が使用され得る。例えば、これらのエラーは、検索されている同期化パターンのように見え得るが、実際には誤った同期化パターンである、バス64上の伝送されたデータをもたらす状況があるという事実によるものであり得る。誤った同期化パターンが、真のランダムなデータトラフィックの結果である場合、それらは通常、短時間にわたって同期化プロセスを妨害するのみであり、それらが消滅するとき、方法900は、真の同期化パターンを見出し、スレーブデバイス54は、バス64に同期化されるであろう。しかしながら、一定同期部分を検索するとき、例えば、一定同期部分に合致する、温度センサによって提供される一定温度データ等の一定のアクティビティをもたらす状況があり得る。これは、たとえ一定同期部分が多くビットの長さであることが可能であっても起こり得る。この場合、これらの誤った同期化パターンは、同期化プロセスを妨害するであろう長時間にわたって、バス64上で伝送されたデータ中に出現し得る。

20

【0452】

ここで図55aおよび55bを参照すると、それぞれ、真の同期化パターンが伝送されるデータ中で送信される第1のシナリオ、ならびに真および偽の同期化パターンが伝送されるデータ中で送信される第2のシナリオの略図が示されている。本実施例では、（制御データ以外の）伝送されるデータは、オーディオデータである。しかしながら、本実施例はまた、計装データ（すなわち、センサデータ）等の他のタイプのデータにも当てはまり得る。図55aでは、制御データは、オーディオデータを備えるデータ部分によって分離される、Sワード、Xワード、およびYワードを備える。Sワードは、真の同期化パターンを備える。図55bでは、制御データは、図55aに示されたものと同じ一般構成要素を備える。しかしながら、ここでは、データ部分中で送信される、3つの誤った同期化パターン（すなわち、3つの誤ったSワード）がある。代替的な場合において、誤った同期化パターンは、データ部分の始めまたは終わりのより近くに存在することができ、サブフレームのうちのいくつかの中のみであり得、および/またはおそらく連続データ部分内の異なる場所にあり得る。誤ったSワードは、真の同期化パターンに対応するデータビットを含む。しかしながら、それらは、フレームの始めにおける正しい場所（すなわち、正しい同期位置）になく、したがって、誤っている。したがって、これらの誤ったSワードは、真の場所が把握されると、それらの場所に基づいて決定され得る。代替として、これらの誤ったSワードは、それらが前述のように期待動的同期部分に合致するかどうかを決定するために、これらの誤ったSワードの動的同期部分をチェックすることによって決定されることができる。

30

40

【0453】

誤った同期化パターンの状況は、汎用同期化方法900でランダム成分を使用して取り

50

扱うことができる。ランダム成分は、ランダム変数を使用することによって取得される。少なくとも1つの例示の実施形態では、ランダム変数は、ある期間にわたる伝送されるデータのパリティであり得る。例えば、パリティは、最後の同期化パターン以降の1の数に偶数である場合には1に設定されることができ、そうでなければ0に設定され、または逆も同様である。パリティは、2を法とするビットの総数の整数値であるが、他の類似ランダム量を定義することができ、例えば、4を法とするビットの総数の整数値（すなわち、加算するときの最後の2ビット）である。少なくとも1つの実施形態では、最後のフレーム同期化パターン以降の全てのバストラフィックに基づく、追加のCRCジェネレータが使用され得る。CRCジェネレータは、CRCジェネレータ自体からの出力を入力として取り、それとバス64からの入力とのXOR演算を行うことができる（レジスタ値ゼロを何らかの他の値にマップするように、特別な規定が作成されるべきである）。バストラフィック中の1または0の数を数えることは、量子化プロセス自体が、物理的測定に存在する熱雑音（例えば、量子・電気雑音または量子・機械雑音）により、ランダムなプロセスであるため、ランダムなプロセスである。次いで、この計算に基づいて、ランダムな決定が行われる。少なくとも1つの実施形態では、決定は、以前の同期化パターンを真のパターンとして保つこと、および次の真の同期化パターンを検索し続けること、または現在の同期化パターンを真のパターンとして保つこと、および次の真の同期化パターンを検索し続けることのいずれかであり得る。この決定は、計算されたパリティが0に等しいかどうかをチェックすることによって行われることができる。代替として、この決定は、計算されたパリティが1に等しいかどうかをチェックすることによって行われ得る。各場合において、同期位置の変化の確率は1/2である。一般的な場合において、ランダム変数に基づいて現在の位置にとどまる確率は、 p となり、新しい位置に変化する可能性は、 $(1-p)$ であろう。以前の同期化パターンまたは現在の同期化パターンが真の同期化パターンとして保持されるかどうかにかかわらず、この方法論を用いて、遅かれ早かれ正しい真の同期化パターンが選択されるであろう。少なくとも1つの実施形態では、現在の同期化パターンを真の同期化パターンとして単に保持する代わりに、方法900が、次の場所が特定された同期化パターンを真の同期化パターンと見なす前に、現在の同期化パターンから始まる、ランダムであるが有限数の同期化パターンが破棄されることができ

【0454】

真の同期化パターンとしての以前に場所が特定された同期化パターンの選択は、 p という確率を有するものとして表されることができ、真の同期化パターンとしての現在場所が特定されている同期化パターンの選択は、選択されたランダム変数（すなわち、一実施形態で議論されるようなパリティ）に基づいて $1-p$ という確率を有するものとして表されることができ。次いで、方法900は、次の一定同期部分を検索し、関連動的同期部分を期待動的同期部分と比較し続ける。場合によっては、一定同期シンボルのように見える、多くのバストラフィックがある場合、方法が2つのフレーム開始位置の間の正しい同期位置にとどまる可能性は、 p^N であり、 N は、誤った同期位置の数である。しかしながら、正しい同期位置が見出されると、検索方法は、誤った位置を破棄し、正しい位置のみを見るであろう。

【0455】

少なくとも1つの実施形態では、検索プロセスを完全に決定論的にしようとして、有効な同期定数に関連付けられる最後のCRC値のリストを作成し、それによって、このリストからの任意の位置に合致する動的同期部分を有さない、以降の同期定数合致を迅速に排除することによって、この同期化検索を加速することが可能であり得る（これは、バス64上の全ての起こり得る妨害に対処するほど十分に長いリストを伴い得る）。

【0456】

少なくとも1つの実施形態では、2つの場所が特定された一定同期部分の間いくつかのタイムスロット（すなわち、同期分離値）があるかを観察することによって、多くの誤った同期化パターンを排除することが可能であり得る。次の場所が特定された一定同期部分に対する間隔の下端は、少なくとも（ワードモードで）同期分離値となるはずである。検

10

20

30

40

50

索方法がLという2つの同期シンボルの間の距離にすでに遭遇している場合、距離が2つの以降で見出された同期シンボルの間のLより短ければ、これらの同期シンボルのうちの少なくとも1つは誤っている（同期シンボルという用語は、同期パターンまたは同期部分という用語に類似することに留意されたい）。

【0457】

少なくとも1つの代替実施形態では、真の同期化パターンが見出されると、本方法は、最大フレーム長に達するか、または場所が特定された同期化パターンの動的同期部分が期待動的同期部分に合致するまで、前方に検索し続けることを含む。この後者の条件が有効である場合には、本方法は、この現在の同期化パターンと以前の同期化パターンとの間の間隔を使用して、将来の同期化パターンを検索し続ける。これは、誤った同期位置であること、または（一定同期部分および動的同期部分の両方がチェックに合格するため）有効な同期位置の良好な候補であることが証明されるまで、所与の第1の同期位置を維持し続けるであろうという点で、以前の方法とは異なる。これは、不確かであるときに2つの同期位置の間でランダムに選択する技法とは対照的である。所与の位置が有効な同期位置ではあり得ないことが証明される場合、同期化方法は、可能性として考えられる有効な同期シンボルをもたらす、次の同期位置まで移動するか、または最大フレーム長を超え得る。この同期化方法が、検証に使用される同期化パターンの数を見出せない場合には、本方法は、最後に場所が特定された同期化パターンを真の同期化パターンとして使用し始めることができるが、これは、行うためにいくらかのメモリを伴い得る。後者の条件が真であり、最大フレーム長に達する場合、同期化方法が新たに始まり、次の真の同期化パターンを検索し始める。この方法は、より高速の同期化を提供すると考えられ得るが、本方法が、（素数の位置、例えば、11個の位置をスキップするために役立つが）ランダムデータの構造により、真の同期位置をスキップし続け得るため、同期化が達成されない場合があるリスクがある。したがって、この方法が機能するために、全ての同期位置を記憶し、この情報と連動すること、またはいくらかのランダム性を本方法に追加すること、例えば、合致する一定同期シンボルおよび合致する動的同期シンボルを再度検索し始める前に、ランダムな数の同期シンボルをスキップすることのいずれかが必要であり得る。本方法は、並列ハードウェア実装に最も適していると考えられる。これが利用可能である場合、いくつかの検索エンジンを実装し、それによって、アルゴリズム全体を非常に速く確実に稼働させることが可能である。N個全ての同期エンジンが起動されるまで、第1の同期エンジンは、最初に求められた同期定数への合致する動的同期シンボルを探し得、第2の同期エンジンは、第2の同期位置への合致する動的同期シンボルを探し得る等である。本方法は、いかなる問題にも遭遇することなく、2つのフレーム開始位置の間に最大でN-1個の誤った同期位置を許容するであろう。

【0458】

少なくとも1つの代替実施形態では、最大データ長内で見出されている、全ての有効な一定同期部分に対応する動的同期部分から表を作成することができる。これらの場所が特定された動的同期部分の位置もまた、表に記録される。次に、期待動的同期部分に合致する、表中の動的同期部分の位置は、位置のリスト中に記録されることができる。位置のリスト中の連続位置の分離に反復がある場合には、これらの位置の間の距離は、実際のフレーム長となり、これらの位置は、同期化を取得するために使用されることができる。この方法に関する主要な問題は、ハードウェアオーバーヘッドであり、別の面では、ロバスタな同期化を提供する。

【0459】

上記の変形例のいずれかでは、同期化が達成される場合には、方法900が作用912を行う。同期化が達成されず、動作モードが依然として正しいと仮定される場合には、方法900は、検索する作用904および906を再度行い、1つ以上のパラメータをリセットし得る。

【0460】

本明細書で説明されている同期化規則のうちの少なくとも2つを使用する実施形態があ

10

20

30

40

50

り得ることに留意されたい。また、本明細書で説明されている同期化規則の種々の組み合わせを使用する実施形態もあり得ることに留意されたい。さらに、本明細書で説明されている同期化規則のうちの少なくとも1つを使用する実施形態があり得ることに留意されたい。

【0461】

912では、同期化がスレーブデバイス54によって達成されている。統一バス通信プロトコルの特定の実装に応じて、スレーブデバイス54は、ロック（すなわち、同期化）を達成したことを信号伝達するように、何らかの情報をマスタデバイス52に返送することを可能にされ得る。その時点で、マスタデバイス52は、統一バス通信プロトコルに従って通信し、有効にされる、少なくとも1つのポートの1つ以上の入力チャネル、1つ以上の出力チャネル、または1つ以上の入力および出力チャネルを有することができるようにスレーブデバイス54を構成するために、例えば、種々のパラメータデータをスレーブデバイス54に送信すること等の種々のアクションを行い得る。少なくとも1つの実施形態では、スレーブデバイス54は、バス64にアタッチされたいかなるポートも有さず、バス64は、スレーブデバイス54内のレジスタREAD/WRITE動作またはFUNCTIONSのアクティブ化等の制御目的で使用されるにすぎないであろう。代替実施形態では、スレーブデバイス54によって、または例えば、マスタデバイス52およびスレーブデバイス54を伴うイベントの組み合わせによって、設定も行われることができる。少なくとも1つの実施形態では、別のマスタデバイスが、スレーブデバイス設定の全てまたはいくつかを設定することに関与することができる。

【0462】

同期化が達成されると、方法900が同期化を維持するように行われ続ける、実施形態があり得ることに留意されたい。エラーがあり、同期化パターンの1つのインスタンスのみに対して同期化が失われる場合において、方法900は、そのパラメータの全てがリセットされる完全同期化プロセスを方法900が受ける必要がないように、動作モードおよびフレーム形式パラメータが同一であると仮定することを伴うことができる。例えば、フレーム長およびフレーム形式が決定されているため、同期化が達成されると、同期化を維持するとき、伝送されるデータ中の他の位置は、それらが、最後の同期化パターンからの決定されたフレーム長の距離の近傍にない限り調査されない。これは、方法900が、同期化パターンの中で伝送されるデータブロック中で発生する、任意の誤った同期化パターンを無視することを可能にする。代替として、同期化が長期間にわたって失われる場合には、最初から同期化プロセスを開始するように、そのパラメータが完全にリセットされた状態で、方法900が再開され得る。実施例として、スレーブデバイス54が、最大限可能なフレーム長より長く同期化パターンを受信していない場合、不具合があり、スレーブデバイス54は、アクティブである任意のポートを非アクティブにし、同期化プロセスを再開するべきである。

【0463】

また、いずれかの動作モードが、方法900のために最初に仮定され得ることに留意されたい。例えば、第1の動作モードは、ワードモードであると仮定され得、第2の動作モードは、ビットストリームモードであると仮定され得る（ワードモードを仮定して同期化が達成されない場合）。しかしながら、代替実施形態では、第1の動作モードは、ビットストリームモードであると仮定され得、第2の動作モードは、ワードモードであると仮定され得る。

【0464】

ここで図56を参照すると、統一バス通信プロトコルを使用して通信するマスタデバイスに同期するための汎用同期化方法950の別の例示的实施形態の略図が示されている。汎用同期化方法950は、並列実装を利用する。952で、方法950が、統一バス通信プロトコルのために第1および第2の動作モード（すなわち、ワードモードおよびビットストリームモード）の両方を仮定し、仮定動作モードのうちの1つの下で同期化パターンを同時に検索する、2つの別個の同期エンジン（すなわち、一方の同期エンジンは、ワー

10

20

30

40

50

ドモードの仮定の下で同期化パターンを検索し、他方の同期エンジンは、ビットストリームモードの仮定の下で同期化パターンを検索する)を使用することを除いて、汎用同期化方法950は、方法900について説明される種々の代替案とともに、汎用同期化方法900に類似する。したがって、作用954は、第1および第2の動作モードに従って伝送されるデータ中の1つ以上の場所で、同期化パターンを同時に検索することを含む。同様に、作用960は、同期化パターンが第1および第2の動作モードのうちの1つのための少なくとも1つ同期化規則に従って検証された場合、同期化を取得することを含む。したがって、作用954、956、960、および962は、作用904、906、910、および912に類似する。

【0465】

958で、同期化が取得されない場合、方法950が、統一バス通信プロトコルに従って、伝送されたデータ上で第1および第2の動作モードの両方に従って検索および取得する作用を実行するので、作用958は作用908とは異なる。同時に両方の動作モードに従って検索するために並列実装が使用されるので、動作モードの間での切り替えは、方法950については行われぬ。

【0466】

作用958から作用954への折り返しは、よりロバストな同期化方法をもたらす。なぜなら、検索の第1のインスタンスにおいて、検索の結果に影響を及ぼしたエラーが発生した可能性があり得るため、同一の検索プロセスが両方の動作モードで繰り返されることができ、異なる結果が達成されることができ得るためである。例えば、バスエラーが生じた可能性があり得、マスタデバイス52が適正に起動または初期化されていない場合があり、または同期化方法950の以前のインスタンスが実行されていたときにバス衝突等が生じた可能性があり得る。場合によっては、同期化方法950が行われるときに、スレーブデバイス54とバス64との間に欠陥のある物理的接続があり得、例えば、最初にスレーブデバイス54をバス64に物理的に接続するときに、適正な物理的接続がなかった場合があり、したがって、適切な電氣的接続が起こるために遅延があった。これらの条件の全てが、同期化方法950の反復において、解消するか、または解決されるであろう。したがって、同期化方法950を繰り返すことにより、よりロバストな同期化プロセスを提供する。同じことが同期化方法900に言える。したがって、作用908で他のモードに切り替わる前に、検索が仮定動作モードの下で1回以上繰り返される、同期化方法900の代替実施形態があり得る。

【0467】

ここで図57を参照すると、統一バス通信プロトコルがワードモードの下で動作しているときに同期化パターンの場所を特定して検証するために、汎用同期化方法とともに使用されることができ得るワードモード同期化方法1000の例示的实施形態の略図が示されている。しかしながら、方法1000は、以下でさらに詳細に説明されるように、クロックゲーティングが使用される場合、ビットストリームモードの下で動作するときにも使用されることができ得る。

【0468】

1002では、方法1000のインスタンス化が初期化される。方法1000への関数呼び出しが、方法1000のインスタンス化を引き起こす。1004では、同期化パターンを検索し、場所を特定し、検証するために使用される、パラメータ、状態、およびカウンタが初期化される。

【0469】

1006では、方法1000がクロックと同期化する。換言すると、方法1000の残りの作用が、バス64のための新しいクロックサイクルで開始する。例えば、作用1006は、クロック信号が低の論理値から高の論理値まで上昇する(すなわち、上昇クロック信号)場合、方法1000の残りの作用が行われるように、構成されることができ得る。代替として、方法1000は、下降クロック信号に同期化されることができ得る。さらに別の実施形態では、本デバイスは、立ち上がりおよび立ち下がりエッジの両方に同期化するこ

10

20

30

40

50

とができる。

【0470】

1007では、フレーム内の現在の位置が増分させられる。

【0471】

1008では、方法1000は、バス64上の伝送されたデータの一部が、方法900および950について説明されたように、一定同期部分に合致するかどうかを決定する。合致がない場合には、方法1000は1011へ進む。合致がある場合には、方法1000は1010へ進む。

【0472】

1010では、方法1000は、カウンタ状態が0に等しいかどうかを決定する。状態は、どの程度まで方法1000が同期化プロセスに沿っているかを数えるために使用される。例えば、状態値が最大状態値に等しい場合には、同期化が達成されている。状態が、現在、0に等しい場合には、これは、一定同期部分の第1の場所に対応し、方法1000は1012へ進む。状態が、現在、0に等しくない場合には、これは、一定同期部分の後続の場所に対応し、方法1000は1014へ進む。

10

【0473】

1011では、方法1000は、少なくとも2つの合致する同期シンボル(すなわち、同期化パターン)が取得されているかどうか、および現在の検索位置がフレーム開始(すなわち、フレームの開始)であるかどうかを決定する。このような場合には、方法1000は、この位置で同期化パターンを見出すことを期待し、エラーがあり、その場合、方法1000は、1050へ進んで、欠落した同期化パターンがあるかどうかをチェックするであろう。そうでなければ、方法1000は、1070へ進み、1006に戻る前にフレームオーバーランをチェックする。

20

【0474】

1012では、方法1000は、状態を1に設定し、決定論的方法および現在の同期動的部分を使用することによって、次の一定同期部分について期待される同期動的部分を計算する。CRCカウンタを伴う決定論的方法の一実施例は、以前に説明された。次いで、方法1000は、1070へ進んでフレームオーバーランをチェックし、次いで、1006に戻って次の一定同期部分を検索する。

【0475】

次いで、1014では、方法1000は、一定同期部分が以前に1度すでに見出されているかどうか、または現在の検索位置が期待フレーム開始位置にあるかどうかをチェックする。この条件が真である場合、方法1000は1016へ進み、そこで、現在の位置がフレームの開始に設定される。1014での比較が偽である場合、現在の場所が特定された一定同期部分は無視され、方法1000は、1070へ進んでフレームオーバーランをチェックし、次いで、1006に戻る。

30

【0476】

次いで、1018では、方法1000は、計算された動的同期部分が、現在場所が特定されている一定同期部分に関連付けられる動的同期部分に合致するかどうかをチェックする(この場合、関連付けられるという用語は、一定同期部分に関連付けられる動的同期部分が、同一の同期化パターンまたはSワードの一部であることを意味する)。1018での比較が真である場合には、方法1000は1020へ進む。1018での比較が偽である場合には、方法1000は1028へ進む。

40

【0477】

1020では、計算された動的同期部分と場所が特定された動的同期部分との間に合致がある場合、方法1000は、現在場所が特定されている一定同期部分を有効と見なし、フレーム長を更新し、次の動的同期部分を計算する。フレーム長は、以前の一定同期部分を含んだ同期化パターンの開始と、現在の一定同期部分を含む同期化パターンの開始との間の距離を計算することによって更新される。次いで、方法1000は1022へ進む。

【0478】

50

次いで、1022では、方法1000は、状態が、同期化を達成するために同期化ワードを見出すことが成功している状態の数である、最大状態に等しいかどうかをチェックする。最大状態の値は、誤ったロックの低い確率とロックを取得するためのより長い時間との間の妥協として選択される。最大状態パラメータをより高い数に設定することは、同期化が達成されているように、十分な同期化パターンの場所が特定されていることを方法1000が決定できる前に、方法1000のさらなる反復をもたらす。したがって、1022での比較が真である場合には、方法1000は1024へ進み、その時点で同期化が達成されている。この場合、方法1000は、将来の同期化パターンをチェックすることによって同期化を維持するために、1070へ、次いで、1006へ進む。1022での比較が偽である場合には、方法1000は1026へ進み、そこで、状態カウンタが1だけ増分させられる。この場合、次いで、方法1000は、1070および1006へ進んで、同期化を達成することを目指すために次の一定同期部分を検索する。

【0479】

1028では、方法1000は、最初に状態カウンタの値をチェックする。いくつかの有効な一定同期化シンボルの場所が特定されている場合、現在の動的同期シンボルエラーは無視され、方法1000は、続行して以前の位置（すなわち、最後に場所が特定された有効な同期化パターンの位置）を基準として保つであろう。次の動的同期シンボルは、無効と見なされる、現在の場所が特定された一定同期部分に関連付けられるものではなく、最後の場所が特定された一定同期部分に関連付けられるものに基づくであろう。代替として、ほんの数個の同期シンボルの場所の特定しか成功していない場合、方法1000は、同期化を取得することに近づくために、バス64上の伝送されたデータのランダム性が使用される、1029へ進むであろう。現在場所が特定されている一定同期部分に関連付けられる動的同期部分が、以前に場所が特定された一定同期部分に関連付けられる動的同期部分および決定論的方法に基づく、計算された動的同期部分に合致しないため、この比較が行われる。したがって、以前の一定同期部分および現在の一定同期部分のうちの1つは、無効であり、作用1029は、以前の一定同期部分を有効な一定同期部分として保つか、または現在の一定同期部分を有効な一定同期部分として保つかを決定しようとする。作用1029は、（例えば、パリティを使用することによる方法900および950について以前に説明されたような）バストラフィックのランダム性、ならびに状態カウンタの値に基づいて、この決定を行おうとする。

【0480】

1028で、状態カウンタの値が最大状態値に近い（例えば、状態カウンタの値が、一実施形態では最大状態値より1小さい、または別の実施形態では最大状態値より2小さい、または別の実施形態では最大状態値の少なくとも半分である）と決定される場合には、方法1000は、同期化を達成する寸前であり、よって、方法1000は、現在の一定同期部分を1回限りのエラーと見なし、以前の場所が特定された一定同期部分が有効であると仮定し、1032へ移動し、そこで、状態カウンタの値を1だけ減少させ、次いで、1006へ進むことによって、次の一定同期部分をチェックする。しかしながら、1029でのランダム性に基づくチェックが偽であり、状態カウンタの値が、方法1000が同期化を達成する寸前ではないことを示す場合には、以前の一定同期位置を有効な位置として保つか、または新しい一定同期位置を有効な位置として使用するか、ランダムな決定が行われる。次いで、方法1000は、1030へ進み、以前のフレーム長および状態情報を破棄し、次の動的同期部分を計算し、次いで、1070および1006へ進んで、次の一定同期部分を検索し続ける。

【0481】

ここで図58aを参照すると、欠落した同期化パターンを取り扱うために使用され、かつワードモード同期化方法1000とともに使用されることができ、方法1050の例示的实施形態の略図が示されている。具体的には、方法1050は、同期化が一瞬失われた場合、方法1000がそのパラメータを最初期化してワードモードの下で同期化パターンを探すべきかどうか、または方法1000が古い検索パラメータを継続すべきかどうか

10

20

30

40

50

を決定しようとする。

【0482】

1052では、方法1050が、フレームの始めに潜在的な欠落した同期シンボルのインスタンスを取り扱うように呼び出される。

【0483】

1054では、現在のフレーム位置が、欠落した同期シンボルのインスタンスおよび方法1000のリセットの両方を網羅するためにリセットされる。

【0484】

1056では、方法1050は、状態カウンタの現在の値が最大状態値に等しいかどうかを決定する。この条件が真である場合には、一時的であり得る、発生した1つだけのエラーがあった可能性があり、その場合、方法1050は1058へ進み、そこで、状態カウンタの値が1だけ減少させられ、次の動的同期部分が計算される。次いで、方法1050は1062へ進み、そこで、方法1000に戻り、次の同期化パターンを探し続ける。しかしながら、1056での条件が偽である場合には、方法1050は1060へ進み、その時点で、状態カウンタおよびフレーム長パラメータは、0にリセットされ、方法1050は1062へ進み、そこで、方法1000に戻って、依然としてワードモードを仮定して、始めから同期化プロセスを再開する。

【0485】

ここで図58bを参照すると、現在の検索位置がフレームまたはチャネルの最大長を超えるかどうかをチェックするために方法1000とともに使用され得る、方法1070の例示的实施形態の略図が示されている(すなわち、方法1070はフレームオーバーランをチェックする)。

【0486】

1074では、方法1070は、現在のモードがビットストリームモードであり、かつ現在のフレームチャネル検索が完了しているかどうかを決定する。この比較が真である場合には、方法1070は、1080へ進んで、状態カウンタ、フレーム長、および検索位置をリセットする。次いで、方法1070は1082へ進み、そこで、方法1070を呼び出した検索方法に戻ることによって、新しい検索が開始される。1074での検索が真ではない場合には、方法1070は1076へ進む。

【0487】

1076では、方法1070は、現在の検索位置が最大限可能フレーム長を超えるかどうかを決定する。この比較が真である場合には、いかなる同期化パターンもワードモードで見出されておらず、方法1070は1078へ進む。1076でのこの比較が真ではない場合、依然として、同期化パターンを見出す可能性があり、方法1070は1082へ直接進み、そこで、方法1070を呼び出した検索方法に戻る。

【0488】

1078では、方法1070が、仮定動作モードをビットストリームモードに変更し、1080へ、次いで、1082へ進むことによって新しい検索を開始する。

【0489】

1080では、現在の位置、フレーム長、および状態値をリセットすることによって、新しい検索が開始される。

【0490】

1082では、方法1070が終了し、それを呼び出した方法に戻る。

【0491】

ここで図59を参照すると、統一バス通信プロトコルがビットストリームモードの下で動作しているときに、同期化パターンの場所を特定して検証するために汎用同期化方法とともに使用されることができ、ビットストリームモード同期化方法1100の例示的实施形態の略図が示されている。

【0492】

1102では、方法1100のインスタンス化が初期化される。方法1100への関数

10

20

30

40

50

呼び出しが、方法 1 1 0 0 のインスタンス化を引き起こす。1 1 0 4 では、同期化パターンを検索し、場所を特定し、検証するために使用される、パラメータ、状態、およびカウンタが初期化される。例えば、初期化は、ビットストリームフレームチャンネルの最大数および最大フレーム長を定義することを含む。1 1 0 6 では、方法 1 1 0 0 は、方法 1 1 0 0 の残りのアクションがバス 6 4 のための新しいクロックサイクルで開始するために、以前に説明されたようにクロックと同期化する。

【 0 4 9 3 】

1 1 0 8 では、方法 1 1 0 0 は、バス 6 4 上の伝送されたデータの一部が、方法 9 0 0 および 9 5 0 のビットストリーム部分について説明されたように、一定同期部分に合致するかどうかを決定する。これは、ビットストリームフレームチャンネルの現在仮定されている数を通して、1 度に 1 つのチャンネルで検索することによって行われる。代替として、並列実装では、これらのチャンネルの全てを同時に検索することができる。

10

【 0 4 9 4 】

1 1 1 0 では、方法 1 1 0 0 は、有効な一定同期部分の場所が特定されているかどうかを決定する。これは、現在場所が特定されている一定同期部分の動的同期部分を計算された動的同期部分と比較することによって行われることができる。条件が真である場合には、方法 1 1 0 0 は 1 1 1 3 へ進む。条件が偽である場合には、方法 1 1 0 0 は 1 1 1 2 へ進む。1 1 1 0 での比較は、一定同期部分の第 1 の場所では行われず、その場合、一定同期部分が有効であると仮定される。

【 0 4 9 5 】

20

1 1 1 3 では、方法 1 1 0 0 は、現在の場所が特定されている一定同期部分に関連付けられる現在の動的同期部分および決定論的方法に基づいて、次の動的同期部分を計算する。ビットストリーム定義が保持され、検索位置が次の一定同期部分を検索するようにリセットされる。

【 0 4 9 6 】

1 1 1 4 では、方法 1 1 0 0 は、状態カウンタの値が最大状態値に等しいかどうかを決定する。もしそうであれば、同期化が達成されている。しかしながら、方法 1 1 0 0 は、同期化が維持されていることを確実にするように、同期化パターンを検索し続ける。したがって、方法 1 1 0 0 は 1 1 1 6 へ移動する。方法 1 1 0 0 は、ビットストリームフレームチャンネルの現在仮定されている数を保ち、次の一定同期部分を検索するように検索位置をリセットし、1 1 0 6 へ進む。

30

【 0 4 9 7 】

この時点で、方法 1 1 0 0 における変形例は、どのビットストリームフレームチャンネルにおいて最後の一定同期部分の場所が特定されたかに着目することと、検索の次の反復のための検索時間を短縮するために、一定同期部分の次の位置を推定するためにこの情報を使用することを含む。これは、少なくとも 1 つの一定同期部分の場所が特定された場合に行われ得る。代替として、これは、2 つ以上のフレームにおいて同じフレームチャンネル場所で一定同期部分を見出した後に行われ得る。

【 0 4 9 8 】

1 1 1 4 での条件が偽であった場合には、方法 1 1 0 0 は 1 1 1 8 へ進み、その時点で、状態カウンタの値を 1 だけ増加させ、1 1 0 6 へ進む。この場合、場所が特定された一定同期部分は有効であったが、同期化を達成するように場所が特定された十分な有効一定同期部分がなかったため、方法 1 1 0 0 は、検索をもう 1 度繰り返して、次の一定同期部分を検索する。

40

【 0 4 9 9 】

1 1 1 2 では、一定同期部分が見出されていなかった場合には、仮定ビットストリームフレームチャンネルの数が 1 だけ増加させられ、検索位置がリセットされる。次いで、方法 1 1 0 0 は 1 1 2 0 へ進み、そこで、ビットストリームフレームチャンネルの仮定数がビットストリームフレームチャンネルの最大数より大きいかどうかを決定する。この条件が偽である場合には、方法 1 1 0 0 は 1 1 0 6 へ進み、そこで、ビットストリームフレームチャ

50

ネルの新たに定義された数に対するビットストリームフレームチャネル中で同期化パターンを検索する。1120での条件が真である場合には、方法1100は1122へ進み、そこで、動作モードがビットストリームモードではないと仮定し、ワードモードに切り替わり、1124へ進み、そこで、続けてワードモードの下で同期化パターンを検索する検索方法を使用する（そのような方法の実施例の説明については図57を参照）。

【0500】

ここで図60を参照すると、統一バス通信プロトコルがワードモードの下で動作しているときに、同期化パターンの場所を特定して検証するために汎用同期化方法とともに使用されることができる、汎用同期化方法1150の例示の実施形態の略図が示されている。しかしながら、方法1150はまた、クロックゲーティングを使用することによって、ビットストリームモードの下で同期化パターンを検索するために使用することもできる。

10

【0501】

1152では、方法1150のインスタンス化が初期化される。

【0502】

1153では、方法1150は、フレームオーバーランの可能性、すなわち、方法1153を呼び出すことによって、現在の最大フレーム長内でいかなる同期シンボルにも遭遇していないことをチェックする（図61b参照）。方法1153への関数呼び出しが、1212で方法1153のインスタンス化を引き起こす。

【0503】

ここで図61bを参照すると、1214で、方法1153は、検索のために仮定される動作モードがビットストリームモードであるかどうか、および現在のチャネルを通じた検索が完了していることを決定する（これは、再同期フラグが真に設定されるときに示される）。この条件が真である場合、方法1153は、新しい検索を準備するために1216へ進む。1216では、現在の検索位置（L）は、フレームの開始に設定され（すなわち、L=0）、現在のフレームの長さ（L_{old}）は、0にリセットされ、状態カウンタは、0にリセットされる（ビットストリームモードでは、チャネル位置が同時に増分させられる）。次いで、方法1153は、1222へ、次いで、方法1150の1154へ進む。1214での条件が真ではない場合、方法1153は1218へ進む。

20

【0504】

1218では、方法1153は、現在の位置が最大フレーム長より大きいかどうかを決定する。この比較が真である場合には（これは、現在の検索中にいかなる同期シンボルにも遭遇しなかった場合にのみ起こるであろう）、方法1153は1220へ進み、そこで、ワードモードからビットストリームモードへ、または逆も同様に、仮定動作モードを切り替え、パラメータsync_{begin}=1を設定する（これは、ビットストリーム同期化方法によって使用されるカウンタを初期化するために使用される、同期化検索プロセスの開始を示す）。その後、方法1153は、1216へ進んで、現在の検索位置（L）をフレームの開始にリセットし、現在のフレームの長さ（L_{old}）および状態カウンタを0にリセットする。次いで、方法1153は、1222へ、次いで、方法1150の1154へ進む。

30

【0505】

ここで図60を再度参照すると、1154では、方法1150は、パラメータリセットが真（すなわち、1）に設定されているかどうかをチェックする。この条件が真である場合には、方法1150は1156へ進む。この条件が真ではない場合には、方法1150は1157へ進む。

40

【0506】

1156では、変数、カウンタ、およびパラメータが初期化される。例えば、この例示の実施形態では、変数L_{old}は、（以前のフレームからの情報に基づく）現在のフレーム長を表し、Lは、フレーム内の現在の位置を表し、stateは、（状態カウンタが最大状態値によって値を境界されるが）見出されている有効同期化パターンの数を表す。CRCは、動的同期部分を生成するために使用されるレジスタを表し、パラメータswe

50

`et_spot`は、同期化が取得されているかどうかを表し（すなわち、同期化が取得された後のフレームごとの1つのクロックサイクル中、`sweet_spot = 1`）、`bitstream_mode`は、検索がワードモード（すなわち、`bitstream_mode = 0`）またはビットストリームモード（すなわち、`bitstream_mode = 1`）の下で行われているかどうかを表し、`sync_begin`は、同期化パターンの検索が始めから再開される必要があるかどうかを表す（すなわち、`sync_begin = 1`）。いくつかのパラメータ、カウンタ、変数が、例証を簡単にするために1156で示されていないこともある。代替実施形態では、他のパラメータ、カウンタ、変数を使用することができる。初期化後、方法1150は1157へ進む。

【0507】

1157では、方法1150は、パラメータ`clock_gate`が真（すなわち、1）に設定されているかどうかを決定する。このパラメータは、ワードモードまたはビットストリームモードの下で動作するとき、同一の検索構造を使用することができるように使用される。クロックゲーティングの使用は、電力を節約し、同期化検索方法を実装するために使用される論理を単純化する方法である。クロックゲーティングの利益は、いくつかのスレーブデバイスがバス64と同期化しようとするかにかかわらず、スレーブデバイスにつき同一の電力消費が達成されることができることである。しかしながら、クロックゲーティングは、代替実施形態では、随意的であり得る。

【0508】

1157でのチェック後、方法1150は、1158へ進んで次のクロック遷移を待機する。別の実施形態では、クロックゲーティングは、クロックイネーブル信号によって置き換えられ得る。この場合、このチェックは、1164に入る前に1160に続くであろう。これは、クロックゲーティングを使用することと同一の利点、すなわち、同一の方法が、ワードモードおよびビットストリームモードの両方に使用され得るという利点を有するが、消費電力が、真のクロックゲーティングと比較してそれほど大きく削減されないという不利点を有する。それは、クロックゲーティングが利用可能ではない構成で使用されることができる。

【0509】

1158では、方法1150は、方法1150の残りのアクションがバス64のための新しいクロックサイクルで開始するために、以前に説明されたようにクロックと同期化する。

【0510】

1160では、方法1150は、`sync_begin`パラメータを0に設定する。これは、全クロックサイクル中にこのフラグを自動的にリセットするように行われるが、フレーム形式が変化するとき（図61bの作用1220を参照）、`sync_begin`は真に設定されるであろう。`sync_begin`パラメータは、ビットストリームモードで使用されるカウンタをリセットするためのフラグとして使用される。

【0511】

クロックゲーティングを用いると、`clock_gate`パラメータが1に設定されているときに、伝送されたデータが読み取られる。したがって、ワードモードでは、`clock_gate`パラメータは、通常、1に設定されるが、ビットストリームモードでは、`clock_gate`パラメータは、同期化パターンについて検索されている現在のビットストリームフレームチャンネルに対応する、伝送されたデータ中のこれらのビットのみについて1に設定される。これは、両方とも0で始まって参照される、現在検索されているビットストリームチャンネル（B）、および現在のチャンネル（全クロックサイクル中に変化するC）に依存する。ビットストリームフレーム形式について調査されるチャンネルの最大数が、`Amax`である一方で、現在の検索は、Aに等しい数のチャンネルを有するフレーム形式を調査する。パラメータDLは、短い形式が使用されることを除いて、Lに類似し、サブフレーム内の現在の位置を監視するために使用される。第1のビットストリームフレームチャンネル指数（`B = 0`）について、およびビットストリームフレームチャンネルの現在

10

20

30

40

50

仮定されている数 A (C の位置は 0 から A - 1 まで変化し得る) については、B の値は、伝送されたデータからデータを読み取るための開始位置を取得するため使用され、C の値は、現在検索されていないチャンネルをスキップするために使用される。本実施例では、クロックゲートおよびビットストリームモードを用いると、方法 1150 は、位置 0 で伝送されたデータを読み取り始め、次いで、A 回、タイムスロットを先にスキップして、検索されているビットストリームフレームチャンネルに対する次の伝送されたデータを読み取る。したがって、本実施例では、clock_gate パラメータは、0 番目、A 番目、2 * A 番目等のタイムスロットについては 1 に設定され、他のタイムスロットについては 0 に設定される。

【0512】

1164 では、パラメータ sweet_spot が 0 に設定され、現在の位置 L が 1 だけ増分させられる。次いで、方法 1150 は 1166 へ進む。

【0513】

1166 では、方法 1150 は、検索されている、現在の伝送されたデータ中の一定同期部分に対する合致があるかどうかを決定する。この作用は、方法 900 および 950 について説明された。この条件が真である場合、方法 1150 は 1168 へ進む。この条件が偽である場合、方法 1150 は、1169 へ進んで、欠落した同期シンボルがあるかどうかをチェックする。

【0514】

1168 では、一定同期部分の場所が特定されているので、方法 1150 は、状態カウンタが 0 に等しいかどうかを決定する。この条件が真である場合には、これは、一定同期部分の場所が特定されている最初の時であり、方法 1150 は 1170 へ進む。この条件が真ではない場合には、方法 1150 は、1172 へ進んで、場所が特定された一定同期部分が有効であるかどうかをチェックする。

【0515】

1169 では、方法 1150 は、状態が少なくとも 2 であるかどうか、および現在の位置がフレームの長さに合致することを決定する (方法が、2 の状態値を有することを意味する、少なくとも 2 つの一定同期部分を見出した場合、フレームの長さが取得されているであろう)。この場合、方法 1150 は、同期化シンボルを見出すことを期待するが、現在のフレームについては何も見出されていない。したがって、方法 1150 は 1174 へ進み、そこで、欠落した同期方法 1200 を呼び出す (図 61a 参照)。1169 での比較が真ではない場合、方法 1150 は、同期化パターンが期待される位置にない。したがって、それはエラーではなく、同期化パターンがない。したがって、方法 1150 は、メイン進入点 1152 に戻って、次の一定同期部分を検索し続ける。

【0516】

1170 では、方法 1150 は、状態カウンタを 1 に設定し (同期シンボルが初めて場所を特定されたことを意味する)、現在の位置 L を 0 にリセットし (これはフレームの開始を示す)、場所が特定される次の一定同期シンボルに関連付けられるべきである、次の動的同期シンボル (すなわち、次の期待動的同期シンボル) を計算する。次の動的同期シンボルは、同期定数に続いて、バス 64 からの読み取られた CRC 値に基づいて計算され、決定論的方法を使用して、次の値まで増分される。次いで、方法 1150 は、1152 へ戻って次の同期化パターンを検索する。したがって、リセットパラメータは真に設定されておらず、方法 1150 は 1157 へ進む。

【0517】

一定同期部分および一定同期シンボルという用語は、互に類似することに留意されたい。また、動的同期部分および動的同期シンボルという用語が、互に類似することにも留意されたい。

【0518】

1172 では、方法 1150 は、単一の同期シンボルのみの場所を特定しているかどうか、または現在の位置が期待フレーム長に合致するかどうかを決定する。この条件が真で

10

20

30

40

50

はない場合、現在の場所が特定されている一定同期シンボルは無視され、方法 1 1 5 0 は主要ループ 1 1 5 2 へ戻るであろう。1 1 7 2 でチェックされた条件が真である場合、方法 1 1 5 0 は 1 1 7 3 へ進む。

【 0 5 1 9 】

1 1 7 3 では、方法 1 1 5 0 は、現在の検索位置 (L) をゼロに設定する。次いで、方法 1 1 5 0 は 1 1 7 6 へ進む。

【 0 5 2 0 】

1 1 7 6 では、方法 1 1 5 0 は、現在場所が特定されている一定同期部分に関連付けられる動的同期部分が、計算された期待動的同期部分に等しいかどうかを決定する。条件が真である場合、方法 1 1 5 0 は 1 1 7 8 へ進む。条件が偽である場合には、少なくとも 1 つのエラーがあり、以前の一定同期部分または現在の一定同期部分のうちの少なくとも 1 つが無効である。次いで、方法 1 1 5 0 は、1 1 8 6 へ進み、以前の一定同期シンボルが見出された位置を選択して、現在の一定同期シンボルを無視するために、ランダム性を使用するか、または現在の一定同期シンボルが見出されている新しい位置を使用し、(それが合致しないため) 以前の位置を無視する。

【 0 5 2 1 】

1 1 7 8 では、方法 1 1 5 0 は、現在のフレーム長 L_old の値を現在の位置 L に設定し、現在の動的同期シンボルの値および決定論的方法に基づいて、次の動的同期シンボルを計算する。次いで、方法 1 1 5 0 は 1 1 8 0 へ進む。

【 0 5 2 2 】

1 1 8 0 では、方法 1 1 5 0 は、状態カウンタが最大状態値に等しいかどうかを決定する。例えば、最大状態値は、最大 CRC 指数が 1 5 であるときに 7 に設定されることができる。別の実施形態では、最大状態値は、1 5 または何らかの他の整数 N であり得る。7 の値を使用すること(または可能性として考えられる動的同期部分の数の約半分である値を使用すること)は、同期化を達成する時間と同期化方法のロバスト性との間の妥協である。例えば、1 5 個の値を伴う CRC カウンタに基づいて、9 つのバイナリビットが、一定同期部分に使用され得、誤った同期化条件(すなわち、誤った同期化パターンに基づいて同期化を達成する)の可能性は、ランダムなバストラフィックに基づき、同期化パターンを N 回検証して、 $(2^{-9N}) * (15^{-N})$ である。しかしながら、同期シンボルに合致するランダムな静的バストラフィックがある場合、この条件が起こる可能性は、 15^{-N} である。N = 7 の場合については、誤った同期化の可能性は、多くても $5.9 * 10^{-9}$ であるが、通常ははるかに低い。

【 0 5 2 3 】

1 1 8 0 での条件が真である場合には、同期化が達成されており、方法 1 1 5 0 は 1 1 8 2 へ進み、そこで、`sweet_spot` パラメータは、スレーブデバイス 5 4 がマスターデバイス 5 2 およびバス 6 4 と現在同期していることを示すように 1 に設定される。このフラグは、内部タイミングを追跡するために、スレーブデバイス 5 4 中の他の論理によって使用され得る。サブフレーム長を求めるために、フレーム長は、3 という因数で除算され得る。次いで、方法 1 1 5 0 は、1 1 5 2 へ進んで、同期化を維持するように将来の一定同期部分を検索し続ける。

【 0 5 2 4 】

1 1 8 0 での条件が真ではない場合には、同期化がまだ達成されておらず、方法 1 1 5 0 は 1 1 8 4 へ進み、そこで、状態カウンタが 1 だけ増分させられる。次いで、方法 1 1 5 0 は、1 1 5 2 へ進んで、次の一定同期部分を検索し、同期化を達成しようとし続ける。

【 0 5 2 5 】

1 1 8 6 では、動的同期部分エラーがあり、したがって、以前に場所が特定された一定同期部分または現在場所が特定されている一定同期部分のうちの少なくとも 1 つが無効である。次いで、方法 1 1 5 0 は、状態カウンタが 4 未満であり、かつパリティが 0 に等しいかどうかを決定する。状態カウンタが 4 以上である場合、方法 1 1 5 0 は、いくつかの

10

20

30

40

50

一定同期部分の場所を特定することにすでに成功しており、同期化中、単一のエラーは許容されるであろう。しかしながら、方法 1 1 5 0 は、状態値を減少させることによって、以前に場所が特定された同期位置を低信頼で扱う。1 1 8 6 での条件が真ではない場合、ランダム成分（例えば、限定されないがパリティビット）は、方法 1 1 5 0 が最後の同期位置（例えば、1 に等しいパリティ）を有効位置として維持すべきこと、または現在の同期位置（例えば、パリティが 0 に等しい）を有効位置として保つべきであることを示す。4 の値は、実際には、ほぼ、2 で除算された最大状態値である、中間状態値であることに留意されたい（本実施例では、最大状態値は 7 に設定される）。他の実施形態では、他の値を最大状態値および中間状態値に使用することができる。バス 6 4 に接続される物理的状態（すなわち、センサが測定データを提供すること、またはアナログ・デジタル変換器が電圧または電流測定値を提供すること）に何かがある限り、パリティの値がフレームによってランダムとなるように、デジタル化プロセスにより、バス 6 4 上で出現するビットに対してランダム成分が生じるであろう。揺動散逸定理が、散逸を伴う任意のシステムに対する揺動を保証するので、この揺動は、量子物理学によって保証される。したがって、全てのバストラフィックが決定論的状态マシンによって生成されない限り、パリティは真にランダムとなるであろう。さらに、他の実施形態では、フレーム中に伝送される 0 および 1 の数がランダムであるため、それが 1 に等しいかどうかを確認するためにパリティがチェックされることができる。1 1 8 6 での条件が真である場合、方法 1 1 5 0 は 1 1 8 8 へ進む。1 1 8 6 での条件が偽である場合、方法 1 1 5 0 は 1 1 9 0 へ進む。

【 0 5 2 6 】

1 1 8 8 では、方法 1 1 5 0 は、以前に場所が特定された一定同期部分を拒絶し、いくつかの変数およびカウンタの値をリセットする。ここで、方法 1 1 5 0 は、現在の位置をフレームの開始として使用するであろう。フレーム内の現在の位置（L）、計算されたフレーム長（L_{old}）、および状態カウンタが全てリセットされる。次の動的同期部分は、バス 6 4 から読み取られた最後の動的同期部分に基づいて、決定論的方法を使用して計算される。これは、フレームについての情報がないということと同等であり、フレームが現在の位置から始まると仮定される。方法 1 1 5 0 は、始めから同期化プロセスを再開することができるように、1 1 5 2 へ進む。

【 0 5 2 7 】

1 1 9 0 では、方法 1 1 5 0 は、以前の場所が特定された同期部分を有効として受け入れ、状態カウンタを減少させる。これは、数回の以前の成功したチェックの後に動的同期部分チェックにおけるエラーに遭遇したため、方法 1 1 5 0 が以前に示されたように現在の位置において低信頼を有することを示す。次いで、次の動的同期部分の値が、バス 6 4 からちょうど取得された動的同期部分ではなく、決定論的方法、および場所が特定された以前の動的同期部分に基づいて計算される。次いで、方法 1 1 5 0 は、1 1 5 2 へ進んで次の一定同期部分を見出す。

【 0 5 2 8 】

ここで図 6 1 a を参照すると、欠落した同期化パターンをチェックするために使用し、かつ方法 1 1 5 0 とともに使用されることができる、同期化方法 1 2 0 0 の例示的实施形態の略図が示されている。

【 0 5 2 9 】

方法 1 2 0 0 は、基本的に、エラーがあつて同期が達成されていない場合には、同期化プロセスが始めから再開される必要があると決定する。そうでなければ、エラーが発生する前に同期化が達成された場合には、方法 1 2 0 0 は、状態カウンタを 1 だけ減少させ、同期化プロセス全体を始めから再開するよりもむしろ、何も起こらなかったかのように次の同期シンボルを検索するであろう。

【 0 5 3 0 】

方法 1 2 0 0 が同期シンボルを期待したが、何も見出されなかったため、現在の位置は、1 2 0 4 でゼロにリセットされ、すなわち、方法 1 2 0 0 は、たとえ最後の場所が特定された一定同期部分に、例えば、バス 6 4 上の雑音による、エラーがあつたとしても、フ

10

20

30

40

50

レーンが現在の位置から始まることを仮定する。これは、クロック信号上の故障ではなく、バス64上の任意のデータエラーに有効となるであろう。

【0531】

1206では、方法1150は、スレーブデバイス54がマスタデバイス52とロックしているかどうかを決定する。それがロックしている(`state = max state`)場合、単一のエラーが許容され、状態値が1208で減少させられるであろう。この場合、次の動的同期部分が、間違っていると考えられるバス64上の現在のCRC値ではなく、バス64上の以前のCRC値に基づいて計算される。次いで、以前のCRC値が更新される。この値がバス64上の現在のCRC値に合致しない場合には、この更新されたCRCは、次の読み取られたCRC値と比較されるように、もう1度更新される。このスキームは、同期化を依然として維持しながら、単一のCRCエラーを許容する(一定同期部分のみにエラーがあると仮定される)。次いで、方法1150は、1211へ進み、メイン検索方法1150の1152に戻る。

10

【0532】

1210では、1つより多くのエラーがあり、またはスレーブデバイス54がまだマスタデバイス52と同期していない。したがって、これらの条件が有効同期化位置には起こらないであろうため、あるパラメータをリセットすることによって、再度、検索方法が再開される。次いで、方法1150は、1211へ進み、メイン検索方法1150の1152へ戻る。

20

【0533】

ここで図62を参照すると、方法1150とともに使用されることができ、ビットストリームモード同期化方法1250の例示的实施形態の略図が示されている。

【0534】

1252では、方法1250のインスタンス化が初期化される。方法1250への関数呼び出しが、方法1250のインスタンス化を引き起こす。

【0535】

1254では、方法1250は、パラメータリセットが真に(すなわち、1に)設定されているかどうかを決定する。この条件が真である場合には、これは、ビットストリームモードの仮定の下で同期化パターンを検索するために使用される、パラメータ、カウンタ、および変数の全てが、それらの初期値にリセットされ、よって、方法1250が1256へ進むことを意味する。そうでなければ、1254での条件が偽である場合、方法1250は1258へ進む。

30

【0536】

1256では、ビットストリームモードの仮定の下で同期化パターンを検索し、場所を特定し、検証するために使用される、パラメータ、状態、およびカウンタが初期化される。例えば、初期化は、ビットストリームフレームチャネルの現在の仮定された数(A)、チャネルが現在検索されていることを示すチャネルカウンタまたは指数(B)、検索されているチャネルに対するクロックゲーティングをアクティブにするための全てのチャネルを通じたカウンタ(C)、およびフレーム内の現在の位置(L)に類似するビットストリームチャネル内の現在の位置(DL)を定義することを含む。チャネルの最大許容数は、 A_{max} によって定義され、1つの実装では16に等しくあり得る。次いで、方法1250は1258へ進む。

40

【0537】

A 、 B 、および C 変数は、0の値から参照されるため、ビットストリームフレームチャネルの現在の数を8に設定することは、 $A = 7$ を意味する。これは、レジスタ空間を節約するように行われる。これらの変数がどのようにして使用されるかを理解するために、 $A_{max} = 2$ であるように、3つの最大数のビットストリームフレームチャネルがあると仮定されたい。次いで、カウンタは、 $A = 0$ 、 $B = 0$ 、 $C = 0$ 、および $DL = 1$ に設定される。これは、現在、1つのビットストリームフレームチャネルがあると仮定され($A = 0$)、同期化パターンについて検索されている現在のビットストリームフレームチャネルが

50

第1のビットストリームフレームチャンネルであり ($B = 0$)、全てのチャンネルに目を通すカウンタが第1の位置で始まり ($C = 0$)、伝送されたデータ中の開始位置が0である ($DL = 1$)ことを意味する。

【0538】

方法1250は、ビットストリームフレーム形式の全ての可能な構造を体系的に経過し、各仮定フレーム形式に対する許容チャンネルのうちの任意のもので同期化を取得しようとすることによって稼働する。方法1250は、どのビットストリームチャンネルがバス64上で現在伝送されているかを示す、チャンネル指数カウンタCを連続的に増分させるであろう。Cがフレーム形式に対して定義されるチャンネルの数(A)に等しいとき、それがリセットされ、一般検索方法1150は、単一のビットを捕捉し、すなわち、バス64上の多重化信号から単一のビットを取り出し、これが同期化を含むフレームチャンネルに属すると仮定するであろう。同時に、チャンネル内の現在のビット位置(DL)が、1だけ前進させられるであろう。同期化がこのチャンネル中で取得されない場合、同期化パターンについて検索される現在のチャンネル(B)が1だけ増分させられる。Bが、チャンネルの現在の数であるAに等しい場合、検索が、現在の仮定フレーム形式に対するチャンネルの現在の仮定された数について完了し、Aが1だけ増分させられる。AがAmaxに等しい場合、方法1250は、全ての形式および全てのチャンネルを検索しており、そうでなければ、同期化パターンについて別のチャンネルを検索するために、C、DL、B、およびAが、その特定の順番で増分させられるであろう。

【0539】

1258では、方法1250は、方法1250の残りのアクションがバス64のための新しいクロックサイクルで開始するために、以前に説明されたようにクロックと同期化する。同期化は、前述のように、立ち上がり、立ち下がり、または両方のクロックエッジのいずれかで行われることができる。

【0540】

1260では、パラメータre_syncが偽(すなわち、0の論理値)に設定される。これは、メイン検索方法1150による検索が途切れずに続行することを表す。このパラメータが真に設定されるとき、現在の検索位置Lおよび現在のフレーム長L_oldは、両方ともゼロにリセットされる。これは、検索されている現在のビットストリームチャンネル中でいかなる同期化シンボルも見出されなかった場合に起こる。次いで、方法1250は1262へ進む。

【0541】

1262では、方法1250は、パラメータbitstream_modeが真(すなわち、1の論理値)に設定されているかどうかを決定する。この条件が真である場合には、方法1250は1266へ進む。この条件が偽である場合には、何も起こらず、方法1250はビットストリーム進入点1252へ戻る。これは、全ての変数が明確に定義された状態にあり、これが電流消費も低下させるように行われる。これは、いくつかの実施形態では省略されることができる。1262で行われるチェックはまた、代替実施形態では方法1250において先に行われ得る。

【0542】

1266では、方法1250は、パラメータsync_beginが真(すなわち、1)に設定されているかどうかを決定する。この条件が真である場合には、これは、同期化パターンの検索がビットストリームモードの下で最初に始まっていることを意味する。方法1250は1268へ進み、そこで、カウンタおよびパラメータは、1つだけのビットストリームフレームチャンネルがあることを仮定するように、それらの初期値にリセットされ、その場合、同期化パターンの検索は、第1のビットストリームフレームチャンネル中の第1のビットから始まる。1266での条件が偽である場合には、同期化パターンに対する検索プロセスは、最初から始まっておらず、方法1250は1270へ進む。

【0543】

1270では、方法1250は、現在のアクティブなビットストリームチャンネル(C)

10

20

30

40

50

が、検索されているビットストリームチャンネルの現在の仮定された数 (A) に等しくないかどうかを決定する。条件が真である場合には、方法 1 2 5 0 は 1 2 7 2 へ進み、その時点で、現在のアクティブなビットストリームチャンネルは 1 だけ増加させられ (すなわち、次のチャンネルが検索目的でアクティブである)、方法 1 2 5 0 は 1 2 5 2 へ進み、その時点で、現在のアクティブなビットストリームチャンネルが、現在調査されているチャンネル、すなわち、チャンネル B に等しくなるまで待機することによって、同期化パターンの検索が実行される。

【 0 5 4 4 】

1 2 7 0 での条件が偽である場合には、これは、現在アクティブであるビットストリームフレームチャンネルが、検索されているチャンネルであることを意味する。この場合、メイン検索方法 1 1 5 0 にとってのクロックが有効にされ、方法 1 1 5 0 が続行することを可能にするように、論理有効フラグが設定されるであろう。したがって、方法 1 2 5 0 は 1 2 7 4 へ進み、その時点で、本方法が、アクティブなチャンネルが検索されている次の時間を決定することができるように、再度、アクティブなチャンネル (C) が新たにゼロから開始される。同時に、調査されているフレームチャンネル内の位置が、1 だけ前進させられる ($DL = DL + 1$)。これは、メイン検索方法 1 1 5 0 における現在の位置 L の増分に類似する。次いで、1 2 7 4 では、方法 1 2 5 0 は、メイン検索方法 1 1 5 0 によって、伝送されたデータの 1 ビットがチェックされることを可能にする。次いで、方法 1 2 5 0 は 1 2 7 6 へ進む。

【 0 5 4 5 】

1 2 7 6 では、同期化パターンが現在のアクティブなフレームチャンネル中で見出された場合には、方法 1 2 5 0 は 1 2 7 8 へ進み、その時点で、同期している可能性がある。次いで、1 2 7 8 では、現在の位置カウンタは、初期位置、例えば、1 2 8 に初期化されるが、また、何らかの他の値、例えば、0 でもあり得る。1 2 7 6 で行われる比較は、場所が特定された同期部分が有効同期部分であると決定するために、方法 9 0 0 および 9 5 0 について説明されたものと同じであり得る。パラメータ DL は、1 つの同期シンボルが見出された場合に、方法 1 2 5 0 がより長時間にわたって検索することを可能にするように、この時点で大きい数に設定され得る。次いで、方法 1 2 5 0 は、1 2 5 2 へ進んで、次のビット中で多重化データストリームから読み取る。しかしながら、1 2 7 6 での条件が真ではない場合には、方法 1 2 5 0 は、進入点 1 3 0 2 へ進んで、残りのビットストリームフレームチャンネルを循環するであろう方法と呼び出す。これは、図 6 3 に関して説明される方法 1 3 0 0 等の別の方法によって達成されることができる。

【 0 5 4 6 】

ここで図 6 3 を参照すると、ビットストリームモード同期化方法 1 2 5 0 とともに使用されることができる、ビットストリーム更新方法 1 3 0 0 の例示的实施形態の略図が示されている。

【 0 5 4 7 】

1 3 0 2 では、方法 1 3 0 0 のインスタンス化が初期化される。方法 1 3 0 0 への関数呼び出しが、方法 1 3 0 0 のインスタンス化を引き起こす。

【 0 5 4 8 】

1 3 0 4 では、方法 1 3 0 0 は、現在のビットストリーム開始位置 DL が最大位置 (最大フレーム長より大きいことと同等である) にあるかどうか、かつ `allow_change` パラメータが 1 (すなわち、論理真値) であるかどうかを決定する。この条件が真である場合、これは、検索方法が依然として同期化を達成する必要があることを意味し、方法 1 3 0 0 は 1 3 0 6 へ移動する。これは、現在のフレームチャンネル中で同期化が取得されなかったことを意味する。いくつかの実施形態では、最大位置は、1 2 7 の値であり得る。これは、DL の初期値が 1 2 8 に設定され、ラップアラウンドで 2 進計数が使用された場合に達成される最後の値であるが、他の実施形態では、他の最大値を使用することができる。1 3 0 4 での条件が偽である場合、これは、方法 1 2 5 0 が現在のフレームチャンネルをチェックすることを完了していないことを意味する。この場合、方法 1 3 0 0 は

10

20

30

40

50

、1318へ移動し、次いで、ビットストリーム検索方法1250に戻って、調査されている現在のフレームチャンネルからより多くのビットを読み込む。

【0549】

1306では、re_syncパラメータが1（すなわち、論理真値）に設定され、パラメータDLが0に設定されるが、それを何らかの他の初期値に設定することもできる。これは、同期化パターンが現在のビットストリームフレームチャンネル中で見出されず、次のビットストリームフレームチャンネルが試行されるべきであることを意味する。パラメータre_syncを1に設定することによって、パラメータLおよびL_loadがメイン検索方法1150においてリセットされ、それによって、新しい検索を開始するのである。次いで、方法1300は1308へ進む。

10

【0550】

1308では、方法1300は、調査されている現在のフレームチャンネル（B）がフレームチャンネルの仮定された数（A）に等しくないかどうかを決定する。この条件が真である場合、チャンネルの現在仮定されている数中の全てのチャンネルが調査されたわけではない。次いで、方法1300は1310へ進み、そこで、調査されている現在のフレームチャンネル（B）が1だけ増加させられる。次のフレームチャンネルは、1318で方法1250を呼び出すことによって、同期化パターンについて検索されるであろう。1308での条件が偽である場合、方法1300は、現在の仮定されたフレーム形式を伴う全てのチャンネル位置が経由されている。パラメータAは、フレームチャンネルの現在の仮定された数を示す。この場合、仮定チャンネルの数（A）は、最終的に1だけ増分させられるであろう。方法1300は1312へ進み、そこで、調査されている現在のフレームチャンネルが再初期化され、0に設定される。次いで、新しい仮定された数のチャンネル内の第1のビットストリームフレームチャンネルが検索されることができ。次いで、方法1300は1314へ進む。

20

【0551】

1314では、方法1300は、ビットストリームフレームチャンネルの現在仮定されている数（A）が、ビットストリームフレームチャンネルの最大許容数（A_max）に等しくないかどうかを決定する。条件が真である場合には、方法1300は1320へ進み、そこで、ビットストリームフレームチャンネルの現在仮定されている数が1だけ増加させられ、次いで、方法1300は、続いて、新たに仮定されたフレーム形式を伴う第1のフレームチャンネルを検索し、1318で方法1250を呼び出す。1314での条件が偽である場合、これは、全ての許容ビットストリームフレーム形式の位置の全てが検索されており、同期化パターンが見出されていないことを意味する。次いで、方法1300は1316へ進み、そこで、ビットストリームフレームチャンネルの現在仮定されている数が0に設定される。次いで、方法1300は、フレーム内の現在の位置（L）およびフレーム長（L_load）を初期化し、動作モードを反転させる（すなわち、ビットストリームモードからワードモードへ）。次いで、方法1300は、1318へ、次いで、検索方法1250の1252へ進む。ワードモードでは、方法1250が依然として作動するが、ビットストリームモードが偽であるため、より低い電力消費を伴う（1262での決定を参照）。ビットストリーム方法1250およびクロックゲートワードモード方法1150を並行して作動させられる（すなわち、遂行または実行される）ことができる。

30

40

【0552】

ここで図64を参照すると、迅速再同期化方法1350の例示的实施形態の略図が示されている。迅速再同期化方法1350は、同期化がすでに達成されており、同期化を維持するために同期化方法が依然として使用されているときに使用されることができ。例えば、方法1350は、方法1150および1250と並行して作動させられるか、または実行されることができ。

【0553】

1352では、方法1350のインスタンス化が初期化される。方法1350への関数呼び出しが、方法1350のインスタンス化を引き起こす。

50

【0554】

1354では、方法1350は、リセットパラメータが真に設定されているかどうかを決定する。この条件が真である場合、方法1350は1356へ進み、その時点で、パラメータ `allow_changes` が1（すなわち、真）に設定される。次いで、方法1350は1358へ進む。しかしながら、1354での条件が真ではない場合には、方法1350は1358へ進む。

【0555】

1358では、方法1350は、方法1350の残りのアクションがバス64のための新しいクロックサイクルで開始するために、以前に説明されたようにクロックと同期化する。

10

【0556】

1360では、方法1350は、状態カウンタの値が最大状態値または最大状態値 - 1（例えば、6および7に設定されることができ）に等しいかどうか、および同期化パターンの検索がビットストリームモードの下で動作しているかどうかを決定する。これは、同期化において単一のエラーを許容するように現在の状態値を2つの異なる状態値と比較する。これが真である場合には、方法1350は1362へ進み、そこで、パラメータ `allow_changes` の値は、同期化が達成されており、たとえいくつかのエラーが発生しても現在の状態が維持されるべきであることを意味する、0（すなわち、偽）に設定される。これは、たとえエラーがビットストリームモードで発生しても、同期化方法が即時にロックから外れないであろうことを意味する。この条件は、`allow_changes` パラメータによって対象とされ、それに従って作用する（換言すると、同期化方法は、単一のエラーに基づいて、ビットストリームモードであると仮定することから、ワードモードであると仮定することに戻って即時に変化することを可能にされない）。次いで、方法1350は、1352へ進んで、状態値の任意の変化を監視し続ける。しかしながら、1360での条件が偽である場合には、方法1350は1364へ進む。

20

【0557】

1364では、方法1350は、検索が現在のフレーム長を越えているかどうか、または同期化パターンの検索がワードモードの下で動作しているかどうかを決定する。したがって、1つより多くのエラーが発生している状況を見出すために、または同期化検索方法がワードモードの下で動作していることを見出すために、1364での条件を使用することができる。1364での条件が偽である場合には、方法1350は1352へ進む。1364での条件が真である場合には、方法1350は1366へ進み、その時点で、`allow_changes` パラメータが1に設定されるため、いかなる同期化シンボルも見出されない場合、（方法1300の作用1304の後に）BおよびAの現在の値が変更され得る。

30

【0558】

ここで図65を参照すると、本明細書で説明される方法1150、1250、および1350と並行して使用されることができ、クロックゲーティング方法1400の例示的实施形態の略図が示されている。クロックゲーティングまたはクロック有効は、ワードモードで常にバスアクティビティを捕捉するように、および以前に説明されたように、監視されている現在のフレームチャネルがビットストリームモードでアクティブである時間中のみバスアクティビティを捕捉するように、メイン方法1150を制御するために使用される。

40

【0559】

1402では、方法1400のインスタンス化が初期化される。方法1400への関数呼び出しが、方法1400のインスタンス化を引き起こす。

【0560】

1404では、方法1400は、リセットパラメータが真に設定されているかどうかを決定する。リセットパラメータは、電源オンリセット信号によって、またはバスコントローラによって開始される制御されたリセットイベント等の外部イベントによってのいずれ

50

かで、設定される。この条件が真である場合には、方法1400は1406へ進み、その時点で、clock_gateパラメータは、1に設定され、それは、デフォルト条件であり、方法1400を使用している検索方法においてクロックが有効にされるべきであることを意味する。次いで、方法1400は1408へ進む。1404での条件が偽である場合には、方法1400は1408へ直接進む。

【0561】

1408では、方法1400は、方法1400の残りのアクションがバス64のための新しいクロックサイクルで開始するために、以前に説明されたようにクロックと同期化する。

【0562】

1410では、方法1400は、多重化ビットストリームチャンネル(C)内の現在の位置が、監視されている現在のチャンネルに等しいかどうか、または本方法がワードモードで動作しているかどうかを決定する。両方の場合において、クロックは、方法1150のために有効にされるべきである。1410での条件が偽である場合、方法1400は1414へ進み、その時点で、パラメータclock_gateは、クロックが無効にされていることを意味する0に設定され、方法1150は、次のクロック周期中にアイドル状態であり、バス上の現在のデータを無視する。作用1412または1414の完了時に、方法1400は、1402へ戻って、動作モード(すなわち、ビットストリームモードまたはワードモード)またはCおよびBの値が変化したかどうかをチェックする。これは、CおよびBが全クロックサイクルで値を変化させることができるため、全クロックサイクルで行われる。

【0563】

また、無線リンクを経由して、本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態を使用することも可能である。この場合、無線リンクまたは無線インターフェースをバスと見なすことができる。それは、(例えば、RF-ID様リンクまたは無線ドッキングステーション等の高速短距離リンクについて)遅延が低い場合、または通信速度が速すぎず、より長い伝搬距離によって制限されている場合の構成に特に適している。例えば、統一バス通信プロトコルの少なくとも1つの実施形態は、低電力磁気リンクシステムで使用されることができ、マスタデバイスは、変調された搬送波信号を伝送し、単一ワイヤバス実施形態について説明されるものと同じタイムスロットシステムを利用する。スレーブデバイスは、RF信号を整流し、図66に示されるように、クロックおよびデータ信号を導出するために、この整流された信号のエンベロープを「バス信号」として使用するであろう。スレーブデバイスがマスタデバイスに通信し返すために、それは、長距離リンク用の別個の伝送機を使用し得るが、単純な磁気リンクについては、伝送される搬送波信号を単純にロードし得る。単一ワイヤバス事例と同様に、伝送される搬送波信号のローディングは、同一の利点(すなわち、受信機が、伝送された搬送波信号から直接、クロックおよびデータを取り出すことができる)を伴ってマスタデバイスに通信し返すための機構として使用されることができ、さらに、プロトコルおよびハードウェアは、実装することが比較的単純である。搬送波信号の伝送およびローディングは、例えば、マスタデバイスおよびスレーブデバイス中のPCBの一部の上のいくつかの円形巻きとして実装される、2つの小型コイルの使用によって達成されることができ、これらのPCBコイルの間隔および巻き数は、特定の動作周波数へのこれらのコイルの同調のために、巻線の間隔の固有寄生静電容量を利用するように改良され得る。代替実施形態では、2本の連結された伝送線が、高い動作周波数および高速データ転送のために使用され得る。伝送される搬送波信号は、典型的には、データの帯域幅よりはるかに高い周波数を有し、例えば、433MHzまたは2.4GHz搬送波信号が、搬送波として使用されることができ、3.072MHzまたは19.20MHzの周波数が、高品質ステレオオーディオの転送速度として使用されることができ、いくつかの実施形態では、特に、例えば、高解像度ビデオ等の広帯域データが転送され得る場合に、24または60GHz等のはるかに高い搬送波周波数が使用され得る。これらの信号は、密接に連結された伝送線を通して伝送

10

20

30

40

50

されることができる。さらに、受信機の所要電力が低い場合、RF-IDデバイスと比較して類似機能を有効にするが、より高い帯域幅およびさらなる機能性の可能性を伴って、(マスタデバイスから送信された)伝送された搬送波信号からの電力によって、このデバイスに電力供給することが可能である。これは、識別情報の転送、または大量の情報、例えば、ファイル、オーディオ、写真、または動画の転送を可能にする。いくつかの実施形態では、データのより高い帯域幅を可能にするために、所与のクロック遷移後、および次のクロック遷移前に、1つより多くのデータビットが転送され得る。データを暗号化することによって、データの安全な取扱を得ることができる。

【0564】

本説明の次の節は、1つ以上の利益を達成するように修正が行われている、統一バス通信プロトコルの他の例示的实施形態について議論する。

【0565】

ここで図67を参照すると、別様にSワードとして知られている同期化ワードに使用されることができる、フィールドおよびビット割付の別の例示的实施形態の略図が示されている。このSワードについては、IRQSビット(S15)は、第1のビット(または第1のタイムスロット)であり、パリティビット(S14)は、ここでは第2のビット(または第2のタイムスロット)であり、ACKビット(S0)は、ここでは最後のビット(またはSワードがバス64上で伝送される最後のタイムスロット)である。このSワード形式の残りの部分は、図7に示されるものに類似し、議論されない。

【0566】

例示的实施形態では、バス64の電源投入時のSワードのデフォルト値は、“0010.1100.0111.1110” = 0x2C7Eであり得る。これは、「1111」の値を伴う疑似ランダムシーケンスを開始することと同等である。

【0567】

Sワードに対するこの異なる構成を用いると、図40から44に示されるもの等の本明細書で説明される例示的フレーム形式の多くが、図67のSワード中のビットの異なる順序付けにより、変化するであろうことに留意されたい。

【0568】

IRQSビットの目的は、スレーブデバイスが送信する重要な状態メッセージを有する場合、READ、WRITE、またはFUNCTION動作を延期することである。この状態メッセージは、IRQSビットをアクティブにすることによって送信されることができる。いかなるスレーブデバイスもIRQSビットをアクティブにしていない(本実施例では「1」の論理レベルがアクティブにすることを意味する)場合、またはIRQS割り込み機能がS15 DELAY MASKによって無効にされる場合、READ、WRITE、またはFUNCTION動作が進行するであろう。割り込みは、スレーブデバイスが「10」または「11」に等しい状態レベルを有する場合にアクティブにされる。この割り込みを生成するために、スレーブデバイスは、IRQSビットがバス上で伝送されているタイムスロット中に、その状態レジスタのMSBのコンテンツをバス64上にコピーすることができる。前述のように、表7は、いくつかのコマンド動作(すなわち、PING、FUNCTION、READ、およびWRITE)ならびにIRQSおよびS0 DELAYビットの値の種々の組み合わせについて起こる、フレームアクションを示す。

【0569】

少なくとも1つの実施形態では、ここでは、割り込みが、図5のSワードの場合のように、Sワードの終わりよりもむしろSワードの始めに起こり得るため、IRQSビットはまた、超低電力モードになるために使用され得る。したがって、図67のSワードを用いると、バス64にアタッチされたデバイスは、フレームの始めに起こる、まさにSワードの始めにスリープモードになるように構成されることができる。スリープモードになるとき、クロック信号は、無効にされ、クロックラインの値は、静的論理0値または静的論理1値であろう。

【0570】

10

20

30

40

50

少なくとも1つの実施形態では、IRQSビットはまた、低電力シャットダウン中にバス64を起動させるために使用され得る。この場合、クロック信号は、高または低論理値状態にしておかれ、データラインの任意の変化は、クロック信号の動作を再開するであろう(すなわち、クロック信号がもはや静的値で保持されなくなるであろう)。データラインは、クロック信号が状態を変化させてアクティブになるまで駆動されることができる。他の実施形態では、スレーブデバイス54は、クロックライン自体をアクティブにすることによってバス64を起動させ得、この場合、マスタデバイス54は、バスホルダを使用してクロックラインを駆動することによって行うことができる、電力低下モード中、クロックラインを弱駆動された静的状態にしておく。

【0571】

PARビットは、図7のSワードについて説明されたように計算され、ここでは議論されない。

【0572】

確認ビットACKは、以前に説明されたように、すなわち、最後のフレーム中のバストラフィックに基づいて計算される。しかしながら、ここでの違いは、ACKビットが、PARビット後のはるかに後の時間で、バス64上で伝送されることである。これは、より遅いスレーブデバイスに、そのパリティ計算およびPARビットとの比較に基づいて、ACKビットを計算するためのより多くの時間を許容するため、有利である。この実施例は、ワードモードでの図67のSワードへのデータ動作を示す、別の例示的タイミング図である、図68に示されている。マスタデバイス52は、ビットS14でパリティ値を書き込み、次いで、スレーブデバイス54は、パリティ値を計算し、マスタデバイス52によって計算されるパリティ値に等しい、または等しくない場合に、それぞれ、後にACKまたはNACK14タイムスロットで応答する。

【0573】

代替実施形態では、SELECT GROUP ADDRESS関数を使用して同時にいくつかのデバイスをプログラムするために、ADDRESSフィールドのアドレス12-14が使用され得るように、Xワードのフィールドが修正されることができる。次いで、このグループアドレスに割り当てられる全てのデバイスは、単一のWRITE動作を使用して書き込まれることができ、それは、時間効率を増加させ、バス64の電力効率を増加させる。グループアドレスは、グループアドレス(GROUP)の値を、この例示的実施形態でデバイスがサポートする、バス64にアタッチされた全てのデバイスへのブロードキャストと同様である15に設定することによって、特定のデバイスに対して無効にされ得る。同様に、デフォルトGROUPアドレスは、デバイスが電源投入時に特定のデバイスサブグループに属さないであろうことを示す、15であり得る。

【0574】

少なくともいくつかの実施形態では、XワードのADDRESSフィールドに行われることができる別の修正は、アドレスフィールドが、バス64上の全ての他のデータと比較して逆順に符号化されることである。この場合、X7がADDRESSビット0であり、X0がADDRESSビット7(図9aに示されるXワードの逆数である)である。この逆順序付けは、ビット7が使用されない場合に、スレーブデバイスがより長い内部アクセス時間を有することを可能にする。この場合、最初の128ワードのみが8ビットアドレスモード(16ビットデータ)でプログラムされることができる。

【0575】

同様に、逆順序付けがXワード中のADDRESSビットに使用される、これらの実施形態では、逆順序付けはまた、Yワード中の8ビットADDRESSフィールドが、16ビットアドレスを形成するためにXワードのADDRESSフィールドと一緒に使用され得る、Yワード中のADDRESSビットに対しても使用されることができる。逆順序付けについては、Y15がビット8であり、Y8がビット15である。ADDRESSフィールド中のこの逆順序付けは、アドレスビット15が使用されない場合に、スレーブデバイスがより長い内部アクセス時間を有することを可能にする。換言すると、アドレスにお

10

20

30

40

50

ける値が、READ動作において等、スレーブデバイスから要求されるとき、なぜなら、要求されたアドレスが、16ビットの代わりに15ビットによって表すことができるアドレス値にある場合、スレーブデバイスは、より多くの応答するための時間を有するであろう。なぜなら、デバイスが、XおよびYワード中のADDRESSフィールドに使用される逆順序付けにより、最初にLSBを受信するからである。次いで、MSBは、ゼロであると仮定される。この場合、スレーブデバイスは、応答するためのもう1つのクロックサイクルを有するであろう。本実施例では、レジスタマップは、16ビットアドレスモード(8ビットデータ)で32kバイトに限定される。

【0576】

ここで図69aを参照すると、統一バス通信プロトコルの例示的实施形態に対するマスタデバイスまたはスレーブデバイスに使用されることができる、レジスタの定義の別の実施例が示されている。これらのレジスタは、図16aに示されるレジスタレイアウトの代替案である。この例示的实施形態では、第2および第3のレジスタが逆転させられていることが分かる。この配列の利点は、レジスタ用のプログラミングのシーケンスが、16ビットデータおよび8ビットデータを使用する両方の場合に順次的であることであり、すなわち、わずかに不規則なプログラミングシーケンス、すなわち、`update register = { 3, 2, 4, 5, 6 etc }`を使用する代わりに、最初にレジスタ2がプログラムされ、次いで、レジスタ3がプログラムされ、次いで、レジスタ4がプログラムされる等である。

【0577】

別の変化は、図16aのIRQF MASKフィールドの代わりに、DIRフィールドが使用され得ることである。DIRフィールドがゼロであるとき、ポート方向は入力である(すなわち、バス64からのデータがデバイスに伝送されるであろう)。デバイスからのポートが入力をサポートしない場合、いかなるデータもバス64から得られず、デバイスが出力を可能にしないであろう。DIRフィールドが1であるとき、ポート方向は出力である(すなわち、デバイスからのデータがバス64に転送されるであろう)。ポートが出力をサポートしない場合、いかなるデータもバス64に書き込まれないであろう。双方向ポートが使用される場合、それは、同時に両方ではなく、入力または出力のいずれかとして使用される。このビットを含むことの利点は、これが、単一のポートを使用して、双方向接続に対するサポートを可能にすることである。いくつかの実施形態では、DIRビットは、一方向ポートによって無視され得る。

【0578】

加えて、この例示的实施形態では、`reg . 0x05`が、SUBGROUPおよびREPEATフィールドに使用される一方で、`reg . 0x06`は、SYNCHRONIZEおよびCHANNEL LENGTHフィールドに使用される。ここで、`reg . 0x07`は、FUTURE EXPANSIONフィールドおよびSKIPフィールドに使用される。ここで、`reg . 0x08`は、FRAME STRUCTUREフィールド(図29aおよび29bのCOMMAND SEPARATIONフィールドと同一である)に使用される。最終的に、`reg . 0x09`は、PCLKDフィールドおよびTDMビット列フィールドに使用される。図16a、16b、29a、および29bのものに類似する名称を有するフィールドは、同一の機能を有し、さらに議論されない。図69aの例示的レジスタ定義における追加のレジスタビットの包含の利点は、レジスタのプログラミングがより単純であり、REPEATおよびSUBGROUPのより多くの組み合わせが可能にされることである。

【0579】

この例示的实施形態では、FUTURE EXPANSIONフィールドは、将来の機能のために保留されている。

【0580】

この例示的实施形態では、ここで、SUBGROUPフィールドは、データチャネルと一緒にグループ化することのより多くの組み合わせを可能にする、線形符号化を使用して

10

20

30

40

50

符号化される。

【0581】

TDMビット列フィールドは、列(0 - 15) TDMデータが転送され始めることを示すために、ビットストリームモードで使用される。TDMデータの開始は、STARTフィールドによって定義され、垂直方向に与えられる。STARTフィールドは、4行の数で与えられた、データを開始する前にスキップする行数を定義する。コマンドワードの終了は、制御ワードを含まない列においてさえも、依然として基準として使用される。コマンドワードを含まない列に対する任意の位置に書き込むことも可能にされる。例えば、列0(コマンドワードを含む列)において、この例示的实施形態では、コマンドワードS、X、およびYを含む位置に他のデータを書き込むことは、可能にされない。スレーブデバイス54は、プログラミングエラーによりバス64をクラッシュすることを回避するために、これらの位置に対する出力を自動的に無効にするべきである。

10

【0582】

この例示的实施形態では、ここで、FRAME STRUCTUREフィールドは、前のレジスタに含まれるため、図29aに示される以降のレジスタに含まれない。しかしながら、図29aに示されるレジスタ表の残りの部分は、この例示的实施形態で使用されることができる。

【0583】

ここで図69bを参照すると、いくつかのチャンネルがポートの各反復に対して転送されることができるかという定義が示されている。SUBGROUPパラメータを変化させることによって、2つ以上のポートからの出力をマージし、それによって、ポート間のデータ伝送の待ち時間を低減させることが可能である。実施例として、2つのポートは、1度に1つのチャンネルを伝送することによって、それらの出力をマージすることができる(SUBGROUP = 0)。このようにして、バス64上のサンプルは、全出力サンプルに対して2つのサポートの間で交互することができる。

20

【0584】

ここで図69cを参照すると、1つのデバイスグループアドレスに割り当てられたデバイスを有する、システム1500の例示的实施形態の略図が示されている。システム1500は、メモリ1504と、デジタル信号プロセッサ(DSP)1506と、マスタデバイス1508とを含むベースバンドプロセッサ1502を備える。システム1500はまた、4つのマイクロホン1510 - 1516も備える。本実施例では、4つのマイクロホン1510 - 1516は、アドレス12を割り当てられるデバイスグループ1に割り当てられる。

30

【0585】

ここで図69dを参照すると、3つのデバイスグループアドレスに割り当てられたデバイスを有する、システム1550の例示的实施形態の略図が示されている。システム1550は、統一バス通信プロトコルを使用する携帯電話システムの例示的实施形態である。システム1550は、例えば、携帯電話またはスマートフォンであり得る。システム1550は、ベースバンドプロセッサ1552と、CODEC1554と、FMラジオ1556と、右チャンネルクラスD増幅器1558と、左チャンネルクラスD増幅器1560と、Bluetooth(登録商標)モジュール1562と、IRセンサ1564と、容量センサ1566と、4つのマイクロホン1568から1574と、2つのバス1576および1578とを備える。バス1576は、要素1552から1566に連結され、バス1578は、要素1552、1554、および1568から1574に連結される。本実施例では、バス1576および1578は、以前に説明されたように、2ワイヤバス実施形態を使用して実装される。さらに、本実施例では、ベースバンドプロセッサ1552は、マスタデバイスとしての役割を果たし、他の要素1554から1574は、スレーブデバイスとしての役割を果たす。多くのデバイスが同一のバスに連結されるとき、容量損失の増加、およびより少ないエネルギー効率が生じるであろうため、システムをよりエネルギー効率的にするために、1つより多くのバスが使用される用途があり得る。加えて、異なる

40

50

バスは、より低いクロック速度がよりエネルギー効率的である、異なるクロック速度を有することができる。統一バス通信プロトコルの別の側面は、それが、低費用デジタル付属品をシステム1550に接続する、バス上で使用され得ることである。さらに、デバイスを別個のバスに接続することによって、クロック速度は、デバイスの全てによって使用される全ての帯域幅の合計に基づいて、もはや決定される必要がなくなり得る。これはまた、電力消費を低減させるはずである。

【0586】

システム1550については、バス1578に連結された4つのデジタルマイクロホン1568から1574に対するデバイスアドレスグループ1、およびバス1576に連結されたデバイスのうちのいくつかに対するデバイスアドレスグループ1および2がある。デバイスアドレスグループ1および2は、それぞれ、デバイスアドレス12および13に対応する。マイクロホン1568から1574、右チャンネルクラスD増幅器1558、および左チャンネルクラスD増幅器1560は、デバイスアドレスグループ1に割り当てられる。容量センサ1566およびIRセンサ1564は、第2のバスに対するデバイスグループ2に割り当てられる。前述のように、並行してデバイスをプログラムするように、コマンドを各デバイスアドレスグループ中のデバイスに並行して与えることができ、これは、コードの行を削減し、プログラミングに使用される実行時間をほぼ同量だけ減少させる。この設定は、マスタデバイス1552が、ステレオ再生状況を迅速に設定すること、複数のセンサを迅速に制御すること、または全てのマイクロホンの利得を同時に制御することを可能にする。

【0587】

ここで図70を参照すると、Xワードで設定され得る、関数および対応するビット設定の別の例示的リストが示されている。これらの関数の大部分は、図11aに示される関数に類似するが、これらの関数のうちのいくつかの順番が変更され得る。ここでは、類似関数については議論しないが、新しい関数、すなわち、SELECT GROUP ADDRESSおよびINVERT ACTIVE BANK関数について議論する。

【0588】

SELECT GROUP ADDRESS関数(X3:X0=0011)の実行は、代替的なアドレスを任意のデバイスに割り当てるために使用され得る。データフィールド(Y1:Y0)における最小の2つのビットは、実際のアドレスを選択し、12から15であり得る(例えば、Y1:Y0=10である場合には、グループアドレス14が選択される)。この例示的実施形態で同時に設定することができる、完全に個々のデバイスアドレスグループの最大数は、3{12、13、および14}であるが、これは、代替実施形態では変更されることができる。デフォルトグループアドレスは15である。

【0589】

SELECT GROUP ADDRESS関数は、最初に同一のグループアドレスを有するようにいくつかのデバイスを設定し、その後ポートレジスタを設定することによって、同時にいくつかのデバイスを迅速にプログラムするために使用されることができる。これらのデバイスは、REQUEST ID関数中にプログラムされるような各自の元のアドレスを使用して、依然としてアドレス指定されることができる。たとえデバイスがこの例示的実施形態で異なるグループアドレスを割り当てられていたとしても、ブロードキャストコマンド(すなわち、デバイスアドレス15)もまた、依然として機能するであろう。SELECT GROUP ADDRESS関数は、同時に同一のコマンドを受信することができる、同一または類似デバイスを一緒にグループ化するために使用され得る。例えば、データチャンネルをプログラムするとき、同一のタイムスロットを使用するように、受信機および伝送機チャンネルの両方をプログラムすることができる。代替的なDEVICE GROUP ADDRESSを、これらの受信機および伝送機チャンネルを伴うデバイスに割り当てることによって、これらのデバイスを同時にプログラムすることが可能である。これは、例えば、複数のスピーカまたは複数のマイクロホン等の複数のデバイスが同時に設定されることを可能にするであろうため、より効率的である。

【0590】

SELECT GROUP ADDRESS関数を用いると、デバイスの個々のDEVICE ADDRESSフィールドが変更される必要はないが、むしろ、デバイスアドレスのうちの一つかが、複数のデバイス（すなわち、デバイスの1つ以上のグループ）に割り付けられる。現在のGROUPアドレス（11がブロードキャストアドレスにデフォルト設定されるであろう）を与える2ビットのみが記憶され得、既存の論理の多くが再利用されることができると、これを実装するためのゲート総数オーバーヘッドが低減させられる。

【0591】

この例示の実施形態では、デバイスアドレス12 - 15は、現在、PING動作中に監視されていない。しかしながら、デバイスアドレス15が使用されるときにコマンドが全てのデバイスに送信されるため、デバイスアドレス15は、BROADCAST機能を有するようにすでに割り当てられている。これは、デバイスアドレス12から14を、デバイスグループアドレスとして使用されるようにしている。特定のデバイスに対してはデバイスグループアドレスを示すために、2ビットレジスタが使用され得る。デバイスグループアドレスが、特定のデバイスに割り当てられる必要はないが、アドレス15にデフォルト設定されるであろうことに留意されたい。

10

【0592】

ここで、デバイスグループアドレスをどのようにして使用するかという実施例が続く。マイクロホンおよびビットストリーム受信機は、両方とも第1のデバイスグループアドレスに割り当てられることができる。次いで、同一のコマンドを使用して、第1のデバイスグループアドレスに割り当てられたデバイスがプログラムされることができる。次いで、マイクロホンが有効にされることができる。次いで、ビットストリーム受信機が有効にされることができる。これは、これらのデバイスのより速い設定をもたらす。本実施例では、デバイスプログラミングは、コードの行数の約半分を用いて行われることができる。

20

【0593】

INVERT ACTIVE BANK関数（X3 : X0 = 0100）の実行は、CURRENT BANKビットを使用することなく、アクティブなバンクを変更するために使用され得る。これは、代替的なバンクにおいて現在の構成を有する、バス64にアタッチされた全てのデバイスをプログラムする必要なく、単一のデバイスをプログラムするために有用であり得る。LSB（Y0ビット）は、アクティブなバンクを決定するであろう。すなわち、Y0 = '0'である場合には、アクティブなバンクは、PINGコマンド中に発行されるCURRENT BANKビットによって決定され、Y0 = '1'である場合には、アクティブなバンクは、CURRENT BANKビットの逆値によって決定される。

30

【0594】

本明細書で説明される統一バス通信プロトコルの種々の実施形態はまた、汎用非同期受信機/伝送機（UART）コントローラとともに使用することもできる。UARTコントローラは、シリアル通信で使用される。例えば、第1のUARTコントローラは、順次的にデータのバイトを1度に1ビット伝送することができ、第2のUARTコントローラは、これらのビットを受信し、それらをデータの元のバイトに再構築することができる。このシリアル通信は、同期的または非同期的であり得る。

40

【0595】

UARTコントローラを、種々の用途に使用されることができ。例えば、UARTコントローラは、従来のシリアルインターフェースに、Bluetooth（登録商標）インターフェースまたはBluetooth（登録商標）モデムに、FM受信機に、WiFi構成要素（利用帯域幅が異なり得る）に、USB（可変帯域幅）通信に、タッチパッドセンサに、低データレートセンサに、および可変データ入力または出力を有する任意の構成要素に利用されることができ。UARTは、典型的には、リアルタイムシステムにおける有限待ち時間を補償するために、FIFO構造を含む。

50

【0596】

典型的なUARTコントローラは、クロック信号ピン、READコマンドピン、WRITEコマンドピン、データ(0-7)ピン、およびIRQ割り込みピンを含む、いくつかの入力または出力ピンを有する。場合によっては、データがシリアル形式で転送される場合、これは、4本のピン(CLOCK、TX DATA、RX DATA、およびIRQ)に削減されることができる。

【0597】

典型的には、UARTコントローラが低電力消費を有することが所望される。しかしながら、低電力消費を取得するために、多くのシステムは、アイドル期間中に、クロック信号を、論理低値または論理高値等の静的値に結び付ける。クロック信号上にアクティビティがないと、電力消費は非常に低くあり得る。実際の電力消費は、クロック信号がアクティブであるとき、デューティサイクルによって決定される。これは、使用されるデータ帯域幅によって直接決定される。

10

【0598】

実施例として、SLIMbusプロトコルは、クロック信号を連続的に作動させること、または未知の待ち時間を伴ってクロックを再開することを伴う。これは、連続クロック要件、したがって、いかなるデータも損失しないという要件により、UARTコントローラの高い電流消費をもたらす。したがって、SLIMbusプロトコルは、低電力消費およびUART使用に適していない。

【0599】

低電力消費を達成するために、別個のワイヤをUARTコントローラに使用することができる。これは、本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態を使用して、2つの端子の端子総額が達成されることができることを意味する。しかしながら、典型的なUART構成要素が、さらにいくつかの端子(すなわち、clk、data_in、data_out、IRQ等)を有するため、専用バスがUARTコントローラに割り付けられた場合でさえも、依然としていくつかの節約がある。

20

【0600】

さらに、低電力消費を達成するために、統一バス通信プロトコルの少なくとも1つの実施形態では、バス64は、クロックアイドルモードで設定され、UARTコントローラ中のバッファがある規定閾値に達すると、起動する。閾値は、典型的には、バッファサイズの0から100%の間である、特定の値にプログラムされることができる。これの後に、UARTバッファが低くなるか、または空になるまで、いくつかのサンプル、例えば、1~3つが、全フレーム中で転送される。

30

【0601】

いくつかの実施形態では、クロック信号は、等時性データ、および時にはUARTデータの両方をサポートするために、2つの周波数値の間で変更されることができ、他の実施形態では、クロック信号は、UART FIFOが新しいデータを有するまで、完全に停止させられるであろう。

【0602】

UARTコントローラは、データを折々にのみ伝送するために使用されることができ、これは、システムがアイドルモードまたはスリープモードで動作しているときに起こり得る。データが低頻度で伝送されるとき、クロック周波数を低減させることができ、これは、電力消費を減少させるであろう。さらに、クロック周波数が低減されるとき、帯域幅も低減させられる。しかしながら、より多くのデータが伝送される必要があり得るとき、次いで、クロック周波数を増加させることができ、本システムを低電力モードから引き出すことができる。

40

【0603】

このスキームはまた、自動帯域幅制御に使用され得る。この場合、伝送機は、全フレームにおいてCURRENT BANKビットを変化させることを可能にされる。CURRENT BANKビットが高に設定されるとき、マスタデバイス52は、低に設定される

50

ときより高いクロック周波数を使用することができ、変化が、レジスタの変化と同時に起こり得る。これは、使用され得る帯域幅に応じて、UARTコントローラがバス64の周波数を動的に制御することを可能にする。ビットストリームモードでは、クロック周波数が固定され得る。

【0604】

電力消費を削減するために使用され得る、種々の他の方法がある。例えば、外部物理層は、多くの場合、制限因子である。したがって、電力消費を削減する1つの効果的な方法は、複数のバスを使用することである。一実施形態では、ビットストリームマイクロホンは、共通DATAライン上でデータを多重化する代わりに、左右のマイクロホン用の別個のDATAラインを使用し得る。こうすることによって、外部電力消費をほぼ半減することができる。なぜなら、クロックラインが周波数の半分で作動することができ、各データラインが単一のマイクロホンからの信号によってのみ駆動されるので、容量損失がより低いからである。一実施形態では、このMONOモードへの変更は、右/左端子を第3の状態（例えば、浮動）に設定し、接続を決定するために図11eの回路を利用することによって、決定されることができ、この場合、入力端子が浮動していた場合、マイクロホンは、MONOマイクロホンとして接続されることができ（両方のクロック半周期で新しいデータを駆動する）。この構成は、全ての既存のシステムとの適合性を可能にする。

10

【0605】

いくつかの実施形態では、クロック信号は、3レベル変換器等の内部マルチレベルデルタシグマ変換器を駆動するために使用され得、次いで、このマルチビット信号は、選択に応じてSTEREOまたはMONOモードのいずれかで、バス上でデータを伝送する前に、2つのレベルに変換される。STEREOおよびMONOインターフェースモードの間の選択は、ピン、レジスタ、またはクロック周波数プログラミングによって決定され得る。マルチビット変換器を使用することの利点は、より低いクロック速度および低い電力消費であり得、それは、2レベル変換器を3レベル変換器と比較すると約2倍であり得、有意な電力削減節約をもたらす。デジタルデルタシグマ変換器または特殊論理が、複数のレベルから2レベルビットストリーム出力に変換するために使用され得る。

20

【0606】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態で使用されることができ、電力消費を削減する別の方法は、システムで何も起こっていないときにアイドル状態になるために、アイドル状態になることができるシステムを可能にすることである。

30

【0607】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態で使用されることができ、電力消費を削減する別の方法は、駆動電圧を低下させることであり、これは、多大な電力消費削減をもたらすことができる。また、全オーディオデータスロットにおいて、2ビットを転送するために、複数の信号電圧、例えば、{0、0.6、1.2、1.8V}を使用することも可能であり得る。

【0608】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態で使用されることができ、電力消費を削減する別の方法は、クロック信号をデータと統合することである。この方法は、あまり融通性がないクロッキング、特殊クロック回復回路の潜在的な使用、およびおそらく、バスにアタッチされた全てのデバイスに対する電力の一定消費をもたらし得る。しかしながら、物理層の物理的電力消費が優勢である場合、データを伴うクロック信号の包含（例えば、単一のワイヤ、ならびにクロックおよびデータの8/10bまたは類似符号化を使用して行われることができる）により、有意な電力削減が結果として生じ得る。この場合、電力消費は、1ビットにつき約0.5回の遷移（これは、物理的制約によって決定される）まで削減され得る。それは、8/10b符号化およびランダムデータについては、1ビットにつき0.625回の遷移である。いくつかの実施形態では、オーディオチャンネルのみが、8/10b符号化を使用して符号化され得る。

40

50

【0609】

本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態で使用されることができ、低電力消費を達成することが可能であり得る別の方法は、2つのモードでシステムを動作させることである。第1のモードでは、システムの特徴の全てが利用可能であり、第2のモードでは、特徴のうちのいくつかのみが利用可能であり、もはやデバイスをプログラムすることが可能ではない。クロック周波数は、必ずしもこれらの異なるモードの間で変更される必要はない。第2のモードは、第1のモードと比較して、より低い電力消費を有する。

【0610】

実施例として、現在、いくらかの帯域幅が、制御および同期化に割り付けられており、これは、任意のシナリオで割り付けられる。この情報をバス64にクロックアウトするために電力を費やす。しかしながら、この帯域幅がもはや使用されなくなる場合、特定のモードでバス64を設定することによって、より低い電力消費が得られ得る。しかしながら、これは、制御がもはや可能ではなくなり、スレーブデバイスおよびマスタデバイスが制御を再獲得するためにシーケンスを使用し得ることを意味する。制御データを含むシナリオから、制御データがないシナリオへ変更することが可能である。制御データが除去された後に同期化を確保することと、必要であれば、スレーブデバイスまたはマスタデバイスが制御シーケンスを開始できることを保証することに問題がある。制御が除去される場合、デバイスは同期から外れ得る。例えば、クロックライン上の故障によりエラーが発生する場合、同期化情報が除去されているため、これらのデバイスは同期に戻ることができない場合がある。この問題に対する最も単純な解決策は、バス64上のデバイスの数を2つのデバイスに限定し、クロック信号が高であるときに一方のデバイスにバス64を駆動させ、クロック信号が低であるときに他方のデバイスにバス64を駆動させることである。これは、バス64上のエラーの場合でさえも、同期化の問題を完全に解決する。

【0611】

制御データを送信しない実施例を依然として続けると、伝送スレーブデバイス（例えば、マイクロホン）は、統一通信プロトコルの一部として事前に決定されたあるシーケンスをマスタデバイス52に送信することによって、マスタデバイス52を起動させ得る。このシーケンスは、受信バッファを使用して、マスタデバイス52にこのシーケンスをフィルタにかけさせることによって、オーディオデータからマスクされるため、オーディオデータが可聴オーディオ信号に変換される状況がある場合に、このシーケンスは聞き取られない。マスタデバイス52は、強制的にスレーブデバイス54を制御モードに戻す必要があり得る。これは、例えば、特定の制御シーケンスを提出し、受信機バッファを使用して、受信機においてオーディオデータからこれを取り除くことによって、限定されないが、クラスD増幅器等の受信デバイスを用いて行われ得る。しかしながら、受信機が制御を再獲得することを希望するときに、強制的に制御に戻すことはより困難であり得る。マスタデバイス52が制御を再獲得することを希望する場合、クロック周波数を異なる値に変更し得るが、これは、いくらか時間がかかるであろう。したがって、このアプローチは、遅くあり得、受信機において特殊なハードウェアを使用し得る。

【0612】

代替として、別の実施形態では、2つのデバイスが同時にバス64上で信号伝達し、結果として生じるバス対立を動作モードの変更に至らせ、再度、バス64上で制御データを伝送し始めるように、物理層が修正され得る。これは、ここで説明されているバスシステムとともに使用され得る。例えば、論理0値が、バス64上に変化がないことによって信号伝達され、論理1値が、バス64上の値の変化によって信号伝達される。注意を欲する受信機が、伝送機と同時に伝送することによって、バス64上で変更を強制し、論理1値のみを伝送することができる。これは、伝送される全ての値が、その以前の値から反転させられることをもたらさずであろう。これは、バス対立をもたらさないであろうが、伝送機が複数の伝送エラーを受けるであろう。次いで、伝送機は、限定された少数のエラーの後に、バックオフして伝送を停止するであろう。マスタデバイス52は、バス64上の任意

10

20

30

40

50

の信号遷移の欠如によって、この条件を識別することができるであろう。次いで、全てのデバイスは、動作を初期プログラミングモードに戻して変更し、マスタデバイス52は、バス64の制御をもう1度再獲得するであろう。

【0613】

この例示的实施形態では、スレーブデバイスがバス64とのロックを失う場合、それは、ビットストリームモードで動作する場合、空の列または列全体、あるいは全ての列にさえも書き込み始めることもあり得る。スレーブデバイスが、現在使用されているフレーム形式に従って、空の列に書き込み始める場合、現在使用されているフレーム形式の空の列中に、なんらかのアクティビティがあるとは考えられないので、マスタデバイス52は、このアクティビティを検出するであろう。スレーブデバイスが、現在使用されているフレーム形式に従って、すでに利用されている列に書き込み始める場合には、リードバック値が書き込まれた値に対応しないであろうため、エラーが1つ以上のスレーブデバイスに対して結果として生じるであろう。これは、基本的に、全ての1を信号伝達することによって、このスレーブデバイスがモード変更を要求していることをマスタデバイスに伝え、それによって、このスレーブデバイスに異常があること、または注意を必要とすることをマスタデバイスに伝えている。このシナリオでは、通常条件下で、ある数の1を信号伝達することは、可能にされない(例えば、127個より多くの連続的な1を伝送すること(それによって、シグマデルタ変換器からの出力に制限を設定すること)、これは、ダイナミックレンジの非常にわずかな(例えば、 $< 0.07 \text{ dB}$)低減につながるであろう)。

【0614】

システムが、互からさらに遠く物理的に離間される物理的構成要素を有するとき、バス直径が増加する。遠く離間される構成要素の実施例は、大型画面TVの使用事例である。例えば、100インチ16/9HDTVは、87インチ(221cm)の辺長および49インチ(125cm)の高さを有する。大型画面TVは、スピーカ出力のビーム形成および多重マイクロホンサポートのためのビーム形成等のある機能性を提供することができる。これは、スレーブデバイス(すなわち、スピーカまたはマイクロホン)の間の大きい物理的距離により、より大きいバス直径をもたらす。アナログ信号伝達スキームを使用して、問題を完全に解決することが可能であるが、場合によっては、デジタルオーディオを遠く離間される構成要素に直接転送できることが有益であり得る。

【0615】

いくつかの従来の場合において、SLIMbus通信プロトコルが、いくつかの利点を提供するため使用され得る。例えば、SLIMbusプロトコルは、クロック周期の半分以内に安定化するようにデータに要求する。クロック周期の第2の半分は使用されず、したがって、異なるデバイス間に重複が発生しないように、クロックの周波数を低減させることが可能である。したがって、より低いクロック周波数を使用することにより、より長いバス半径が使用されることを可能にする。

【0616】

代替として、これまで説明されている統一バス通信プロトコルの種々の実施形態は、データ間の沈黙のクロック周期の半分を有さない。この解決策は、より低い電力消費およびより高い帯域幅を可能にするが、それは、また、高い帯域幅をサポートするために、クロック遷移後のある時間以内に(典型的には約5ナノ秒以内に)、データがバス64から除去されるべきことも意味する。これは、データを転送するために両方のクロックエッジを使用する場合、20MHzクロックについてバス半径を約30cm(設計要件による)に限定する。

【0617】

しかしながら、本明細書で説明される統一バス通信プロトコルの少なくとも1つの実施形態は、長いバスの半径を可能にするように、SLIMbus物理層を直接模倣することができる。例えば、偶数の列とともにビットストリームモードを使用し、1つおきの列のみを利用することによって、SLIMbus駆動モード(D-Z-D-Z等)を模倣することが可能である。これは、バス構成要素の物理層を変化させることなく、より長いライ

10

20

30

40

50

ンを駆動することができるという利点を有する。しかしながら、不利点は、電力消費が S L I M b u s プロトコルのものと同程度になることである。さらに、非常に長いバスラインを駆動するために、クロック速度が、伝送ライン影響を回避するように低減させられ、データが安定化するための長い時間があり得る。結果は、長いワイヤ長およびあまり効率的ではない信号伝達による、より低い帯域幅である。

【 0 6 1 8 】

いくつかの実施形態では、同一の I / O セルを使用して S L I M b u s および説明された統一プロトコルの両方をサポートするように、同一の物理層が実装され得る。

【 0 6 1 9 】

非常に短いワイヤを用いると、伝送ライン影響に関する問題がない。しかしながら、複数のデバイスを同一のバス 6 4 に接続するとき、別様に 2 地点間高速接続に使用され得る、整合伝送ラインを利用することが困難であり得る。

【 0 6 2 0 】

これらの問題への解決法は、デバイスの全てが同時に駆動されるわけではない、および / またはデバイスの全てが同一のバスにアタッチされるわけではないように、バス 6 4 に接続されるデバイスに代替のトポロジーを使用することによって、可能にされる。パイプまたはリングトポロジー / 構造でデバイスを接続することによって、たとえ個々のリンクが単一の伝送機および受信機のみを有するとしても、全てのデバイスがアクセスされることができる。個々のデバイスの間の距離が、はるかに短くあり得るため、伝送速度は、単一の伝送機および単一の受信機を伴うリンクにとって、はるかに高くあり得る。この構造は、合計して単一の大きい数になる代わりに、いくつかのデバイス間にタイミングのマージンを効果的に分配する。

【 0 6 2 1 】

ここで図 7 1 を参照すると、デバイスをバスに連結するためにリングまたはパイプトポロジーを使用するシステム 1 6 0 0 の例示的实施形態が示されており、本システムは、統一バス通信プロトコルを使用する。システム 1 6 0 0 の構造は、リングまたはパイプをマルチドロップトポロジーに追加することと同等である (マルチドロップという用語は、多くのデバイスが単一のワイヤにアタッチされることを意味する)。システム 1 6 0 0 は、デバイスの全てを接続するために使用される、2 つの異なるバス 1 6 0 2 および 1 6 0 4 を含む。バス 1 6 0 2 および 1 6 0 4 は、クロック信号を送信するために 1 本のワイヤが使用されることができ、データ、制御情報、および同期化情報を送信するために別のワイヤが使用されることができ、2 ワイヤ実施形態を使用して実装される。2 ワイヤバスを使用することの利点は、ピン数が削減され得、それによって、費用および空間要件も低減され得ることである。

【 0 6 2 2 】

システム 1 6 0 0 は、バス 1 6 0 2 を通してスレーブデバイス 1 6 0 8 から 1 6 1 4 に接続されるマスタデバイス 1 6 0 6 を含む。スレーブデバイス 1 6 1 4 はまた、マスタデバイス 1 6 0 6 をスレーブデバイス (スピーカ) 1 6 1 6 から 1 6 2 2 と接続するように、ハブデバイスの役割も果たす。スピーカ 1 6 1 6 から 1 6 2 2 のトポロジーは、スレーブデバイス 1 6 0 8 から 1 6 1 4 のトポロジーとは異なることに留意されたい。具体的には、スピーカ 1 6 1 6 から 1 6 2 2 は各々、クロックラインを受け取るが、データラインは、スピーカ 1 6 1 6 から 1 6 2 2 の各々の間で順次に接続される。したがって、動作時に、マスタデバイス 1 6 0 6 は、バス 1 6 0 2 を制御し、この特定の実施例については、スレーブハブデバイス 1 6 1 4 を通して、スピーカデータを伝送する。次いで、スピーカ 1 6 1 6 は、スレーブデバイス 1 6 1 4 からデータを受信する。次いで、スピーカ 1 6 1 8 は、スピーカ 1 6 1 6 からデータを受信する。次いで、スピーカ 1 6 2 0 は、スピーカ 1 6 1 8 からデータを受信し、次いで、スピーカ 1 6 2 2 は、スピーカ 1 6 2 0 からデータを受信する。最終的に、スピーカ 1 6 2 2 からのデータは、スレーブハブデバイス 1 6 1 4 に返送されることができる。したがって、バス 1 6 0 4 に接続される構造は、リング構造である (スピーカ 1 6 2 2 がスレーブハブデバイス 1 6 1 4 に接続されない場合には

10

20

30

40

50

、構造はパイプである)。いくつかの実施形態では、クロック信号は、最初に、ハブデバイスからデータを受信するデバイスに送信され、他の実施形態では、クロック信号は、最初に、データをハブデバイスに送信するデバイスに送信され得る。いずれにしても、スピーカ1616から1622の間の物理的距離、ならびにスピーカ1616とハブデバイス1614との間の物理的距離、ならびにスピーカ1622とスレーブハブデバイス1614との間の物理的距離は、データ伝送をデバイス1614から1622の各々の間で同期化されることができるよう、ほぼ同一に選択される。スピーカ1616から1622の間、ならびにスピーカ1616から1622とスレーブハブデバイス1614との間で送信することができるデータは、リングトポロジの場合には完全帰還路を使用して、またはパイプ構造の場合には部分帰還路を使用してのいずれかで、所与のスピーカの温度を示すインピーダンス情報または状態情報であり得る。スレーブハブデバイスから、パイプまたはリングにアタッチされたスレーブデバイスへ伝送されるデータは、典型的には、同期化、状態、およびプログラミング情報、ならびに例えば、オーディオ等のデータである。図71は、マスタデバイス1606からスレーブハブデバイス1614へ進むオーディオ情報を示すが、情報の流れは、逆方向でもあり得ることに留意されたい。いくつかの実施形態では、スレーブハブデバイス1614は、スピーカまたはマイクロホンと組み合わせられ得る。

10

【0623】

ここで図72を参照すると、デバイスをバス(図示せず)に連結するためにパイプトポロジを使用する、システム1650の例示的实施形態が示されており、システム1650は、統一バス通信プロトコルを使用する。システム1650は、増幅器1662、ならびに抵抗器1664および1666とともに、スレーブハブデバイス1652と、スレーブデバイス1654から1660とを備える。(デジタル入力)増幅器は、オーディオデータの受信機の実施例にすぎない。

20

【0624】

図72に示されるトポロジの利点は、ハブデバイス1652が単一のクロック信号を取り扱うことである。したがって、たとえ最も外側の構成要素の間の大きい距離により、システム1650にかなりの遅延があっても、これらの遅延は、全てのデバイスにわたって分散される。また、スレーブハブデバイス1652またはマスタデバイス(図示せず)が、データを伝送するために使用される同一のクロック信号にタイミングが従った情報を受信するので、伝送されるクロック信号と受信されるクロック信号との間に差異があるはずがない。

30

【0625】

ここで図73を参照すると、ハブデバイス(HUB)を通して多くのデバイスAMP1、AMP2、AMP3、およびAMP4をバスに連結するためにパイプ制御を使用する、システム1700の例示的实施形態が示されており、システム1700は、統一バス通信プロトコルを使用する。この場合、データを2つの異なる方向に転送されることができるよう、タイミングがより困難である。データの大部分は、左から右へ伝送されるが、右から左へ伝送される何らかのデータ、典型的には、状態または制御データがある。

40

【0626】

また、データが正しい時間にデバイスの全てに伝送されるために、隣接するデバイス間に遅延があり得ることに留意されたい。これは、レジスタ設定によって定義されることがある数のクロック周期、ハブデバイスからのビットストリームデータを遅延させることによって行われることができる。遅延は、データが、正しい時間に両方向へ(ハブデバイスからスレーブデバイスへ、およびスレーブデバイスからハブデバイスへ)進むことを可能にする。パイプトポロジでは、典型的には、例えば、状態情報が、スレーブデバイスからハブデバイスへ転送され得る。システム1700では、ハブデバイス(HUB)が、ビットストリームチャンネル6を1つのサンプルだけ遅延させ、第1のスレーブデバイス(AMP1)が、ビットストリームチャンネル6を1つのサンプルだけ遅延させ、第2のスレーブデバイス(AMP2)が、ビットストリームチャンネル4から6を1つのサン

50

プルだけ遅延させ、第3のデバイス（AMP2）が、ビットストリームチャンネル2から6を1つのサンプルだけ遅延させる。パイプの終点は、いかなるサンプルも遅延させない。この例示的实施形態では、ハブデバイスは、マスタデバイスまたはスレーブデバイスであり得る。

【0627】

ここで図74を参照すると、デバイスをバス（図示せず）に連結するために1次元交互リングトポロジーを使用する、システム1750の例示的实施形態が示されており、システム1750は、統一バス通信プロトコルを使用する。システム1750は、ハブデバイス1752と、スレーブデバイス1754から1760と、増幅器1762と、抵抗器1764および1766とを備える。抵抗器1764および1766は、クロック信号の直列または並列終端として使用される。増幅器1762は、デジタル入力オーディオ受信機の実施例である。

10

【0628】

システム1800の利点は、両方向へ（ハブデバイスからスレーブデバイスへ、またはスレーブデバイスからハブデバイスへ）の大量の情報の転送が、より単純であることであるが、ハブデバイスは、クロック信号の伝送部分とび受信部分との間のスキューを補償する必要があり得る。リングトポロジーが、隣接するスレーブデバイスの間でほぼ等しい距離を取得するような方法で使用される（これが行われない場合には、信号が完全に安定化させられないため問題があり得る）ため、このトポロジーは、サウンドバー等の大型パネル上で使用されることができる。代替として、システム1800は、このクロックスキューを調整するために、スレーブデバイス14においてクロック信号を利用し得る。

20

【0629】

図74に示される例示的实施形態では、物理的デバイスは、スレーブデバイスの間でほぼ等しい物理的距離を取得するように、異なる方法で番号付けされる。この場合、スレーブデバイス1754は、スレーブデバイス1760に物理的に隣接し、スレーブデバイス1760は、スレーブデバイス1756に物理的に隣接し、スレーブデバイス1756は、スレーブデバイス1758に物理的に隣接する。しかしながら、スレーブデバイス1754から1760は、それらの間に、ならびにスレーブデバイス1754とハブデバイス1752との間に、およびスレーブデバイス1760とハブデバイス1752との間に、ほぼ等しい電気的距離があるように、互に電気的に接続される。デバイスのこの物理的配置の利点は、デバイス間の電気的距離がほぼ同一であり、それによって、これらのデバイス間で等しくタイミングのマーヅンを分配することである。不利点は、より複雑な配線パターンである。デバイスのこの物理的配置はまた、リングトポロジーを利用する他のバス通信プロトコルとともに使用され得る。

30

【0630】

ここで図75を参照すると、デバイスをバス（図示せず）に連結するために2次元リングトポロジーを使用する、システム1800の例示的实施形態が示されており、システム1800は、統一バス通信プロトコルを使用する。この例示的实施形態では、ハブデバイスは、スレーブデバイスのリングと接続し、各デバイス間に約20~30cmがある（他の類似する短い小距離も使用することができる）。データが順次にスレーブデバイスの各々を通して進行するため、データは短い時間内に安定化すべきである。このトポロジーは、個々のスレーブデバイス間の物理的距離が全体的な物理的距離よりはるかに短い場合に、多くの大型構造をサポートすることを可能にし、たとえデバイスによって対象とされる合計の全体的な物理的距離が極めて大きくあり得たとしても、非常に高い速度を依然として達成することができる。クロック信号が、リング構造の始点または終点のいずれかに供給され得る一方で、クロック信号の終端は、リング構造の反対側の端で起こり得る。

40

【0631】

図75のトポロジーを使用する少なくともいくつかの実施形態では、可能な限り最高の伝送速度を取得するために整合伝送ラインを使用して、クロックおよび/またはデータラインが実装され得る。このアプローチの不利点は、整合目的で、これらのデバイスの

50

伝送機における慎重に調整された出力インピーダンスおよび受信機の入力インピーダンス、ならびにPCB上の比較的広い物理的空間である。さらに、帯域幅の使用量が非常に高くない限り、この解決策は、他の伝送の方法と比較して過剰な電力消費をもたらすであろう。しかしながら、この解決策は、伝送ライン影響が無視される技法よりもはるかに大きい帯域幅を提供する。

【0632】

ここで図76を参照すると、多くのデバイスをバスに連結するためにパイプ制御を使用する、システム1900の別の例示的实施形態が示されており、システム1900は、統一バス通信プロトコルを使用する。ここで、タイミングは、データが左から右へ転送されるのみであるため、(図73の構造と比較して)より単純である。このトポロジーは、パイプトポロジーより大きい帯域幅を有するが、伝送されるクロック信号と受信されるクロック信号との間にいくらかのクロックスキューがあり得るため、ハブデバイスまたはマスタデバイスの受信セクションで追加の処理を使用し得る。パイプリング構造の両方で全てのスレーブデバイスをプログラムし、データをスレーブデバイスの全てに転送することが可能である。両方の構造が、ビーム形成スピーカアレイを作成するために使用され得る。これらの構造はまた、マイクロホンアレイを作成するために使用され得る。この場合、パイプの始めの代わりに、パイプの終わりが、最大帯域幅のためにハブに接続されることができる。このトポロジーはまた、他の信号転送の方法とともに使用することもできる。

【0633】

通常シナリオでは、情報の転送がある時間を定義するために、バスクロックエッジが使用される。しかしながら、クロック信号の遷移の数を削減することが可能であり、これは、クロック受信機中の内部クロック乗算器を使用すること、またはクロックおよびデータ信号を組み合わせることを伴い得る。理論的な場合において、平均で全クロックエッジ中の時間の50%を変化させるデータ信号を制御する、連続クロック信号の場合と比較して、外部電力消費が3倍も削減され得る。しかしながら、入力クロック信号を受信した後にロックを獲得するために、PLLがいくらかの時間を要するであろうため、デバイスがバス上でオーディオデータの伝送を可能にするためにより長い時間がかかり得、1つのロック条件から別のロック条件へ変化するために、PLLがいくらかの有限時間を要するであろうため、1つのクロック周波数から別のクロック周波数へ変化するとき、いくつかの制限があり得る。さらに、オーバークロックをサポートしないデバイスを含む、任意のデバイスがバス64にロックすることを可能にするために、(マスタデバイス52は、クロックエッジの直前に、または代替として、クロックエッジ後のデータの最初の部分で、フレームのデータの最後の部分中で信号を送信し得る(両方のクロックエッジがこの目的で利用されることができるから)。このシナリオでは、マスタデバイス52は、いかなる他のデバイスもこのタイムスロットの他の部分を利用することを可能にしないことによって、タイムスロット全体を制御し得る。このタイミングは、バス64にロックされたデバイスが、ちょうど伝送されたフレーム中のS、X、およびY情報を読み取り、それに応答することを可能にする。このスキームは、スレーブデバイスとマスタデバイス52との間でデータを転送するために使用されるが、通常、このオーバークロックモードでのタイミング制約により、2つのスレーブデバイス間でデータを転送するためには使用されないであろう。マスタデバイス52およびスレーブデバイス54の両方は、ジッタを低減させるために、内部同期分割器を使用し得る。スレーブデバイス中の内部PLLは、レジスタマップ中のビットをリセットすることによって、オーバークロックが使用されない場合、無効にされ得る。さらに、オーバークロック動作モードでは、PLLジッタによる間違った読み取りを回避するために、バス衝突を回避するために、および電力消費の増加を回避するために、いくらかの空間がオーバークロックデータと固有のジッタとの間に追加される必要があり得るため、合計の可能帯域幅は、通常、オーバークロックの使用がない場合よりも低い。

【0634】

ここで図77aを参照すると、2Xオーバークロックデータスロットが使用される(す

10

20

30

40

50

なわち、2 というオーバークロッキング因数がある) ときの実施例のタイミング図が示されている。これは、半分のデータを送信するのみである、図 4 a に示されるタイミングと対照的である。図 7 7 a に示されるタイミングを用いると、クロック遷移があるとき、データの 1 ビットのみよりもむしろ 2 ビットが送信される。したがって、例えば、図 4 a に示されるスキームと同量のデータを送信するために、それほど多くないクロック遷移が使用されるため、節電がある。図 7 7 a は、データが立ち上がりクロックエッジおよび立ち下がりクロックエッジの両方の上で送信されるため、単一のクロックサイクルについて 4 つのデータサンプリングイベントがあることを示す。

【0635】

図 7 7 a はまた、マスタデバイス 5 2 によって使用されるクロックとスレーブデバイス 5 4 によって使用されるクロックとの間に、どのようにジッタがあり得るかも示す。このジッタは、有効データが示されるようにバス 6 4 上で伝送されない、ある期間に変換される、スレーブデバイスによって使用されるクロック信号の影付き領域として示されている。本実施例は、マスタデバイス 5 2 から書き込まれた 1 つだけのデータビットを示し、他の場合において、マスタデバイス 5 2 は、他のタイムスロットに書き込むことも可能であり得る。PLL の有限ロックイン範囲により、このスキームは、典型的には、このスキームが利用されないときよりもはるかに狭い周波数範囲で使用され得る。一般に、クロック信号を用いることなくロックを達成することが可能ではないため、このスキームは、連続クロック信号を使用する。

【0636】

ここで図 7 7 b を参照すると、オーバークロックデータスロットを利用するデバイスに使用され得る、タイミングパラメータ値の例示的实施形態を示す表が示されている。4 倍のオーバークロッキングは、半分のクロック周期につき (2 つの代わりに) 4 つのデータスロットを使用して行われ得、6 倍のオーバークロッキングは、半分のクロック周期ごとに 6 つのデータビットを伝送することによって行われ得る。5 ナノ秒より長い、出力を無効にする時間 (T_{DZ})、および 9 ナノ秒未満である、出力を有効にする時間 (T_{ZD}) を有することのみが可能である場合には、出力電流は、違反期間中に 1 mA 未満となるべきであることに留意されたい (図 7 7 b の * を参照)。データの任意の重複が、望ましくない電力消費の増加につながるであろう。また、PLL の VCO 範囲は、内部デバイスタイミングイベントによって必要とされる場合、整数の因数だけより高くあり得る (図 7 7 b の + を参照)。

【0637】

ここで、いくつかのオーバークロッキング実施例の実際のタイミングについて議論する。8.2 MHz クロックを使用する 2 X オーバークロッキングについては、設定時間は、方程式 1 によって与えられる。

$$T_{SETUP} = T_{CLKMIN} / 2 - T_{ZD} - T_{CLKSKEW(WRITE)} - T_{CLKSKEW(READ)} - T_{PCB} - T_{SLJITTER} - T_{MSJITTER} - T_{DLY} \quad (1)$$

= (70 / 2 - 16 - 2 - 2 - 2 - 4 - 1 - 4) ナノ秒 = 4 ナノ秒、マージン 2 ナノ秒
ホールド時間は、方程式 2 によって与えられる。

【0638】

$$T_{HOLD} = T_{ZD} - T_{CLKSKEW} + T_{PCB} - T_{SLJITTER} - T_{MSJITTER} + T_{DLY} \quad (2)$$

= (9 - 2 - 4 - 1 + 2) ナノ秒 = 4 ナノ秒、マージン 3 ナノ秒
バス対立時間は、方程式 3 によって与えられる。

【0639】

$$T_{BUSCONFLICT} = T_{ZD} + T_{SLJITTER(MIN)} - T_{DZ(MAX)} - T_{SLJITTER(MAX)} \quad (3)$$

= (9 - 2 - 5 - 2) ナノ秒 = 0 ナノ秒 (0)

8.2 MHz クロックを使用する 4 X オーバークロッキングについては、設定時間は、

10

20

30

40

50

方程式 4 によって与えられる。

$$T_{SETUP} = T_{CLKMIN} / 4 - T_{ZD} - T_{CLKSKEW(WRITE)} - T_{CLKSKEW(READ)} - T_{PCB} - T_{SLJITTER} - T_{MSJITTER} - T_{DLY} \quad (4)$$

= (140 / 4 - 16 - 2 - 2 - 2 - 4 - 1 - 4) ナノ秒 = 4 ナノ秒、マージン 2 ナノ秒

ホールド時間は、方程式 5 によって与えられる。

【0640】

$$T_{HOLD} = T_{ZD} - T_{CLKSKEW} + T_{PCB} - T_{SLJITTER} - T_{MSJITTER} + T_{DLY} \quad (5)$$

= (9 - 2 - 4 - 1 + 2) ナノ秒 = 4 ナノ秒、マージン 3 ナノ秒

10

8.2 MHz クロックを使用する 6 X オーバークロッキングについては、設定時間は、方程式 6 によって与えられる。

$$T_{SETUP} = T_{CLKMIN} / 6 - T_{ZD} - T_{CLKSKEW(WRITE)} - T_{CLKSKEW(READ)} - T_{PCB} - T_{SLJITTER} - T_{MSJITTER} - T_{DLY} \quad (6)$$

= (204 / 6 - 16 - 2 - 2 - 2 - 4 - 1 - 4) ナノ秒 = 4 ナノ秒、マージン 2 ナノ秒

ホールド時間は、方程式 7 によって与えられる。

【0641】

$$T_{HOLD} = T_{ZD} - T_{CLKSKEW} + T_{PCB} - T_{SLJITTER} - T_{MSJITTER} + T_{DLY} \quad (7)$$

= (9 - 2 - 4 - 1 + 2) ナノ秒 = 4 ナノ秒、マージン 3 ナノ秒

20

前述のように、バス 64 にアタッチされるデバイスのうちのいくつかは、オーバークロッキングをサポートし、バス 64 にアタッチされるデバイスのうちのいくつかはサポートしない場合には、バス衝突のリスクがある。オーバークロッキングをサポートしないデバイスは、データスロット全体（バスクロック周期の半分）を占有し得る。他のデバイスがこのスロットの一部をすでに占有している場合、バス衝突が 1 つ以上のデバイスによって報告されるであろう。マスタデバイス 52 が、クロックエッジの前の最初または最後のスロットを利用しているであろうため、オーバークロッキングをサポートしないデバイスは、多くの状況下でバス 64 とロックすることができる。したがって、マスタデバイス 52 からの全ての READ 動作が正しいはずである。これは、マスタデータスロットがバス 64 上のクロックエッジの前の最後のスロットである、2 倍のオーバークロッキングの実施例について、図 77c で見ることができる。

30

【0642】

ここで図 77d を参照すると、オーバークロッキングを実装するために使用されることができる、レジスタ値の実施例が示されている。また、各オーバークロッキングレートを用いて達成され得る、推定節電も示されている。

【0643】

ここで図 78a を参照すると、1 度にデータの複数のビットを送信するためのタイミングおよび電圧レベルスキームが示されている。本実施例では、2 つの電圧レベルの代わりに、使用される 4 つの電圧レベルがある。4 つの電圧レベルは、タイムスロット（すなわち、データスロット）につき 2 つのデータビットを送信するために使用されることができ、これは、より多くのデータが所与の期間で送信されるため、より高い帯域幅を可能にする。この多重電圧レベルデータスキームはまた、同量のデータを送信するためにより低いクロックレートを使用することができるため、より低い電力消費も可能にする。しかしながら、多重電圧レベルデータスキームは、2 電圧レベルデータスキームよりも電磁干渉（EMI）に敏感である。

40

【0644】

例えば、1.8 ボルト電力供給があると仮定して、2 電圧レベルデータスキームでは、

50

データを伝送するために0 Vおよび1.8 Vの電圧を使用することができ、これは、0.9 Vの雑音マージンがあることを意味する。4電圧レベルデータスキームを用いると、0 V、0.6 V、1.2 V、および1.8 Vの電圧が異なる電圧レベルに使用されることができ、これは、0.3 Vの雑音マージンがあることを意味する。4電圧レベルの場合、0 Vは「00」を表すことができ、0.6 Vは「01」を表すことができ、1.2 Vは「10」を表すことができ、1.8 Vは「11」を表すことができる。マルチレベル信号伝達が使用されるとき、単一のデバイスのみが1つより多くのビットを決定することができるという点で、軽微な制限があることに留意されたい。したがって、1つより多くのデバイスが書き込むことを望み得る、任意のビット（例えば、確認または割り込みビット）について、このデータビットは、マルチレベルではなく2電圧レベル信号伝達を使用する必要がある。

10

【0645】

ここで図78bを参照すると、オーバークロッキングモードを使用して、デバイスがデータビット中で複数のパックされたビットを送信する（すなわち、マルチレベル信号伝達）ときに使用されることができ、例示的符号化の表が示されている。この場合、受信デバイスは、それに従ってビットを解釈するように構成されるであろう。この符号化は、バス64上での伝送の数を最小化し、それによって、電力消費を最小化するであろう。実施例として（データビットにつき6ビットを仮定して）、グループ中のバイナリワードの一部がB17 - B12 = "111001"を有する場合には、図78bに示される符号化スキームを使用して、"100101"が伝送されるであろう。別の実施例として、バイナリストリームの一部が"101011"である場合には、図78bに示される符号化スキームを使用して、"100001"が伝送されるであろう。符号化されたデータ（例えば、マイクロホンからのビットストリーム）が、そのようなシステム（例えば、通常の範囲内で動作するデルタシグマ変換器）からの典型的な出力に一致する限り、このスキームは、遷移の数を削減し、それによって、電力消費を削減する。

20

【0646】

システムは、異なるレベルの電力消費制御を達成するために、以下の動作モードまたは動作の変化の種々の組み合わせを有するように構成されることができる。

【0647】

通常の動作モードは、デフォルト動作モードとして設定され得る。このモードでは、マスタデバイス52は、バス64が有効にされて作動しているように、ある値をプログラムし得る。これを達成するために、マスタデバイス52は、「00000」に等しくない値をMCLKDフィールドにプログラムし得る。ここで、バス64が有効にされ、作動する。

30

【0648】

少なくとも1つの実施形態では、マスタデバイス52は、「00000」等のある値を、例えば、MCLKDフィールドにプログラムすることによって、バス64を無効にし得る。逆に、マスタデバイス52はまた、（「00000」がバス64を無効にするために使用されるときに）「00000」以外の何か等のある値を、例えば、MCLKDフィールドにプログラムすることによって、バス64を起動させ得る。

40

【0649】

電力消費を改変するために、バスクロック構成に行われる変更があり得る。例えば、少なくとも1つの実施形態では、マスタクロック分割フィールドMCLKDに記憶された値を変更することによって、バス64の動作周波数を切り替えることが可能である。場合によっては、個々のデバイスポート用のクロック分割器（PCLKDフィールドを参照）もまた、同時に変更される必要がある。両方の動作が、非アクティブフィールドBANK中の情報を変更するべきである。全てのデバイスが設定されているとき、動作モードを変更するために、同一の更新されたCURRENT BANKフィールドを伴う2つのPING動作が使用され得る。

【0650】

50

電力消費を改変するために、バスモード構成に行われる変更があり得る。例えば、BANKフィールドの値の変更中に、フレーム中の列数を変更することが可能であり得る(FRAME STRUCTUREフィールドを参照)。これは、帯域幅の動的制御を可能にする。次いで、COLUMN WIDTHフィールドが、非アクティブBANKフィールド中で、バス64にアタッチされた全てのデバイスにおいて変更され得る。この動作が完了したとき、以前の非アクティブBANKフィールドに対応する新しいBANK数を伴う2つのPINGフレーム(すなわち、動作)を提出することによって、CURRENT BANKフィールドが変更され得る。

【0651】

別の動作モードは、(PING動作中に)REQUEST BANK CHANGEフィールドをアクティブにすることによって、デバイスがより多くの帯域幅を要求し得る、デバイスBANDWIDTH CONTROLモードである。マスタデバイス52が、すでにENABLE MODE CHANGESフィールドを1に設定している場合、BANKフィールドが次のPING動作中に変化するであろう。本実施形態では、これは、(PING動作中の)REQUEST BANK CHANGEフィールドおよびENABLE MODE CHANGESフィールドが両方とも1に設定されている限り、継続するであろう。マスタデバイス52は、元のバンク選択より多くの帯域幅を可能にするように第2のバンクを構成することによって、より多くの帯域幅を可能にし得る。他の実施形態では、デバイスが、電力消費を節約するために、より少ない帯域幅を要求し得る。

【0652】

別の動作モードは、DEVICE ACTIVATED WAKE UP FROM BUS SLEEPモードであり、このモードにおいて、マスタデバイス52がバス64へのクロックをオフにしたか、または無効にした後、任意のデバイスが、DATAラインのレベルを変更することにより割り込みを開始することによって、バス64を起動させることができる。この場合、バス64が無効にされているとき、それは、3状態モードで保たれ得るが、スレーブデバイスのうちの少なくとも1つがバス起動を開始することができるように、弱い駆動でバスホルダによって駆動され得る。スレーブデバイス54がデバイス起動を開始するとき、割り込みがマスタデバイス52に送信され、次いで、マスタデバイス52は、バス64に対するクロックを再度アクティブにし始めるであろう。この場合、バス64は、マスタデバイス52への入力の役割を果たす。加えて、この場合、バス64にアタッチされたデバイスの全てが、他の手段によって低電力モードにプログラムされていない限り、起動するであろう。マスタデバイス52のクロックの電源が切られているか、または単に無効にされているかどうかは不明であるため、この特定の動作のために低遅延を達成することが可能ではない場合がある。具体的には、マスタデバイス52の電源が切られている場合、発振器で使用される水晶のQ値に応じて、マスタデバイス52の主要発振器が再度通常レベルで作動する前に、数千のクロック周期を要し得る。

【0653】

別の動作モードは、MASTER INITIATED DEVICE WAKE UP FROM SLEEPモードである。非常に低い電力消費のために、デバイスの全てではないにしてもほとんどが、無効にされたそれらのポートを有し得、バス64が停止させられ得る。しかしながら、それは、通常、バス64にアタッチされる残りのデバイスと依然として通信することができる一方で、バス64にアタッチされたいくつかのデバイスに対する電力消費を最小限まで選択的に削減できることは、便利である。これを達成するために、特別な状態(その場合、スレーブデバイスは、フレームを折々のみ監視し、バス64上の他のトラフィックの全てを破棄するであろう)にスレーブデバイスを設定するために、FUNCTION SET POWER(「0001」および「0010」、部分シャットダウン)によって開始される、SLEEPモードが使用され得る(図79参照)。無視されるであろうフレームの数は、15個または255個のフレームのいずれかであり得、その後、1つのフレームが監視されるであろう。これらの数は、起こった16回または256回のクロックサイクルを検出することが容易であるため、および両方とも1

10

20

30

40

50

5で割り切れるため、選択されている。したがって、待機期間は、任意の数の $M = N * L$ フレームであり得、 N は、正の整数であり、 L は、使用される同期化コードの異なる数に基づいて選択され、 L は、この例示的实施形態では15である。この技法では、スリープ状態であるデバイスの内側の疑似ランダムカウンタは、たとえ多数のフレームがスキップされていたとしても、依然として次のフレームに合致するであろう。例えば、依然としてマスタデバイスと同期したままである一方で、電力消費を削減するために15 * N フレームを無視するスレーブデバイスを示す、図79を参照されたい。正しく実装されると、このモードが、1MHzのバスクロック速度で1uA未満の内部電力消費をもたらし得る一方で、(容量性ローディングによる)外部電力消費は、1桁大きくあり得る。電力消費をさらに低減させるために、バスクロック周波数は、100kHz以下に設定され得る。この動作モードで、マスタデバイス52は、スリープ状態であるデバイスに通信し返す方法として、PING動作を活発に使用し得る。DEVICE STATUSフィールドに書き込むことによって(デフォルト値はゼロであり得る)、マスタデバイス52は、1(すなわち、高論理値)に等しい最上位ビットを設定することができる。最上位ビットが1に等しい場合、デバイスが再度、起動するべきである一方で、それがゼロである場合、デバイスは、この例示的实施形態では低電力モードを維持するべきである。デバイスがスリープ状態であり、 $N * 15$ フレームごとにバス64上のデータをチェックしているのみであるため、次いで、マスタデバイス52は、デバイスが実際に起動するまで、 $N * 15$ フレームのためのデバイス起動コマンドを繰り返し送信する必要があり得る。

【0654】

別の動作モードは、DEVICE INITIATED WAKEUP FROM SLEEPモードである。このモードは、電源を切った後にデータ転送のための低遅延を必要とするデバイスに使用され得る。典型的なシナリオの実施例は、データを待っており、突然、その内部バッファが、データによって迅速に満たされているUARTデバイスである。したがって、UARTデバイスは、送信すべき多くのデータを有し得るが、マスタデバイス52が、電力消費を削減するようにバス64を非アクティブに設定していることがある。水晶発振器が、 Q / f_0 にほぼ比例する非常に長い起動時間を有するので(Q は、水晶の品質係数であり、 f_0 は、動作周波数(例えば、10MHz XTAL水晶については約1ミリ秒、および32.768kHz水晶については1,000ミリ秒))、水晶発振器を利用することによって、必ずしも低遅延を達成することが可能ではない場合がある。したがって、低遅延を必要とするデバイスは、代わりに、低い既知の起動時間を伴う内部緩和(RC)発振器を使用し得る(図80b参照)。マスタデバイス52は、FUNCTION SET POWERを設定することによって、UARTモードのためにデバイスをプログラムし得る。これは、UARTが、使用される帯域幅に応じてバス64を動的に制御すること、およびデータをマスタデバイス52に送信することを可能にする。次いで、マスタデバイス52は、その水晶発振器が安定したときに引き継いでもよい。代替実施形態では、LC発振器またはリング発振器が、RC発振器の代わりに使用され得る。これらの異なる発振器は、1つまたはいくつかのクロック周期で再起動することができ、これは、有利であるが、マスタデバイス52のクロック信号より多くのジッタをもたらし得る。

【0655】

UARTデバイスが引き継ぐ前にクロック対立を回避するように、マスタデバイス52がクロックを制御したい場合(例えば、UARTデバイスからの応答がない)において、マスタデバイス52は、クロックラインを変更することなく、ある遅延を伴って、異なる値をDATAラインに1回以上書き込むことができる。これは、マスタデバイス52が、再度、クロック信号を再アクティブ化したいことをUARTに信号伝達し、それによって、バスクロックライン上の潜在的な衝突を回避する。

【0656】

電力消費を削減するための多くのオプションがあるが、制限因子は、実装で 사용되는物理層であり得ることに留意されたい。したがって、システムのデバイスの全てが、同一

10

20

30

40

50

のバス上で実行されなくてもよい。いくつかのデバイスは、他のデバイスよりはるかに高い帯域幅を必要とし得、異なるバス上でこれらのデバイスを作動させることによって、電力が節約され得る。さらに、システムクロック速度は、バス64にアタッチされるデバイスの数とともに増加し、これは、容量損失を増加させ、したがって、電力消費をさらに増加させる。これに加えて、より低い信号伝達電圧、例えば、1.80Vの代わりに1.20Vを使用することによって、電力損失を有意に最小化することが可能である。

【0657】

ここで図80aを参照すると、統一バス通信プロトコルを使用してポートまたはデバイスのために設定されることができ、種々の電力消費モードまたは電力管理モードの代替実施形態の実施例が示されている。FUNCTION SET POWERは、図80a 10
に示されるモードのうち少なくとも1つに従って、ポートまたはデバイスの動作モードを設定するために使用されることができ。

【0658】

この場合、電力消費スケールは、関数によって設定され、正確であると見なされるべきではないが、むしろ典型的な値と見なされるべきである。したがって、図80aに示される電力レベルは、異なるスレーブデバイスに対して変化し得る。FUNCTION SET POWERを使用して変化することができるモードは、依然として、完全制御レジスタマップへのアクセスを有し得るが、アナログ回路の性能に影響を及ぼし得る可能性が高い。デバイスは、電力管理についての以前の節で説明されるように、特別な機構を使用して、シャットダウンモード0000から0011に戻り得る。少なくとも1つの実施 20
形態では、デバイスが低電力状態から戻されるとき、シャットダウンコマンドが受信される前と同一のモードで動作を再開し得る。

【0659】

シャットダウン電力モード(0000)では、スレーブデバイスは、完全にシャットダウンされ、バス64上のアクティビティは、もはやスレーブデバイスによって監視されなくなるであろう。シャットダウン電力モードにされると、マスタデバイス52がスレーブデバイスを起動させることは可能ではない。スレーブデバイスの起動は、スレーブデバイス自体によって、または完全にシャットダウンしたスレーブデバイスを起動させるために使用される制御信号用の追加の端子を使用することによって、行われる必要があり得る。

【0660】

ゲートクロックシャットダウン電力モード(0001および0010)では、非常に低い電力消費があり、内部回路のクロックゲーティングは、これらのモードのいずれかに置かれたスレーブデバイスに対して使用される。さらに、全てのポートおよび全ての内部関数の電源が切られる。バス64がもはや監視されなくなるが、M番目ごとのフレーム中に、スレーブデバイスは、その論理をアクティブにし、1つのフレームに対して起動するであろう(実施例については図79参照)。スレーブデバイスは、依然として同期しているかどうかを依然としてチェックし、XおよびYワードをチェックし、それに従って応答するであろう。マスタデバイス52が、このモードでスレーブデバイスを起動させたい場合、スレーブデバイスがメッセージを受信して応答する前に、最大でM回、このスレーブデバイスに書き込む必要があり得る。スレーブデバイスが(例えば、スリープ状態であった 40
間のモード変更による)同期化エラーを受ける場合、例えば、2つ等のある数の同期エラー後、再度、同期に戻ろうとし、プロセスにおいてその内部論理の電源を入れるであろう。同期に戻った後、スレーブデバイスは、再度、その内部論理を非アクティブにするであろう。この例示的实施形態に対して、 $M = 16 = 15 + 1$ の値については、疑似ランダム同期カウンタの完全サイクル(すなわち、15個のフレーム)がスキップされる一方で、 $M = 256 = 15 * 17 + 1$ については、すなわち、疑似ランダム同期カウンタの17回のサイクル(すなわち、272個のフレーム)がスキップされ、次のフレームが監視されることに留意されたい。

【0661】

図80aを再度参照すると、電力消費モード「0011」が、UARTクロック停止モ 50

ードに対して使用され得、UARTは、クロックが停止した後、それ自体でバス64を起動させることができる。電力消費モード「1000」は、制御のみの用途に使用され得、全ての内部回路は、統一バスプロトコルに関連付けられる論理を除いてシャットダウンされる。電力消費モード「1001」から「1111」は、通常動作モードに使用され得、種々の電流が内部回路に対して増減し、それによって、デバイスの実際の性能を決定する。これらのモードの間で選択することは、電流消費とスレーブデバイス性能との間の妥協になるであろう。電力消費モード「0100」から「0111」は、現在、定義されておらず、スレーブデバイスにおいて第1の利用可能な低電力モードにマップされ得る。

【0662】

統一バス通信プロトコルの実施形態を使用してデータ通信を増進する別の方法は、クロック信号がバス64上で伝送される方法を変更することである。例えば、冗長性をデータ信号に追加し、それによって、データ信号中のデータと多重化されるクロック情報を伝えるために、通常はデータ伝送に使用される1つ以上のタイムスロットを使用することが可能である。したがって、データ自体が、クロックエッジを伝えるために使用される。これは、ここで説明されるように、データ転送のための効率の増加を可能にする。これはまた、より少ない電力消費ももたらし、クロック情報を伝えるために利用される帯域幅を低減させることができる。この技法は、ビットストリームデータが伝送されている場合に、またはワードデータが単一のワイヤを経由して、あるいは単一の伝送チャネルを使用して伝送されている場合に、使用されることができる。この技法はまた、クロック信号を生成する技法が異なるチャネルからのデータを組み合わせないため、各データチャネルが互から独立して制御されることも可能にする。この技法はまた、複数のソースとともに使用することもできる。この技法は、Dビットごとにクロック遷移が生じることを保証する。この技法はまた、別個のクロックおよびデータライン用の第2のワイヤを使用することなく、同期化が維持されることも可能にする。

【0663】

ここで図81aを参照すると、単一のワイヤがクロックおよびデータを転送するために使用される、統一データ形式の例示的实施形態が示されている。クロックが、立ち上がりエッジによって定義される一方で、データは、立ち下がりエッジによって定義される。これは、「Single wire bus system」と題された公開された米国特許出願第2012/0144078号で、さらに説明されている。各データビットは、3つのタイムスロットを使用して伝送され、1つのタイムスロットは、クロック低信号を送信するために使用され、1つのタイムスロットは、クロック高信号を送信するために使用され、1つのタイムスロットは、実際のデータに使用される。したがって、本実施例では、立ち上がりエッジが、クロック情報を伝えるために使用される一方で、立ち下がりエッジは、データ情報を伝えるために使用される。この場合、データ帯域幅は、1/3であり、電力消費は、ビットにつき2回の遷移に関係する（低・高について1回の遷移、および高・低について1回の遷移）。

【0664】

ここで図81bを参照すると、6つのデータスロットまたはタイムスロットを使用して、4つのデータビットが伝送される、データ転送の第2の実施例が示されている。再度、クロック情報を伝えるために、2つのタイムスロットが利用される（第1のデータスロットが低クロック信号を信号伝達し、第2のデータスロットが高クロック信号を信号伝達する）。この場合、データ帯域幅は、2/3であり、電力消費は、ビットにつき0.875回の遷移に関係する。したがって、図81aと比較して、データ転送に利用可能な帯域幅がより高いとともに、消費電力がより低い一方で、同期化のための同一の低・高遷移が使用される。

【0665】

ここで図81cを参照すると、クロックおよびデータ情報をより効率的に伝送するためにビット反転を使用する、データ転送の実施例が示されている。この場合、データビット群のうちの1つのデータビットは、1回繰り返されるが、2回目には別の極性で伝送され

10

20

30

40

50

(すなわち、ビット反転される)、それによって、全データシンボル群に対するクロック遷移を生成する。換言すると、データ列の第1のビットは、クロック信号のエッジを伝えるために、2回目に反対極性で繰り返される。換言すると、データビットのうちの1つは、全データ群に対するクロック遷移を提供し、バスにロックするために使用する遷移をデバイスに提供するように、1回は反転して、1回は通常で(または逆も同様に)、2回伝送される。したがって、 D_0 および

【0666】

【数1】

$\overline{D_0}$

10

は、クロックパルスを提供するとともに、データの1ビット(すなわち、 D_0)を伝送する。代替実施形態では、これらの2ビットを反対順で伝送することができる。データビット D_0 は、オーディオ情報、制御情報、または他の情報等の情報を伝送する。したがって、データビット D_0 は、バス64上のアクティビティにかかわらず、任意の値を有することができる。したがって、ここでは、6つのタイムスロット中に、伝送されるデータの5ビットがある。

【0667】

図81cの実施例では、データ帯域幅は、5/6(6つのデータスロットにつき5つの伝送されたビット)であり、電力消費は、ビットにつき0.7回の遷移に関する。この決定を行うために、データビットが相関していないことが仮定される。したがって、

20

【0668】

【数2】

$\overline{D_0} \rightarrow D1$

、 $D1 \rightarrow D2$ 、 $D2 \rightarrow D3$ 、および $D4 \rightarrow D0$ に進んで、遷移の1/2という確率がある(次のビットが同一の値である場合、いかなる遷移もない)。 $D0$ から

【0669】

【数3】

$\overline{D_0}$

30

への遷移は、遷移の1という確率がある。したがって、遷移の総数は、 $5 * 0.5 + 1 = 3.5$ である。伝送される合計5ビットがある。したがって、ビットあたりの総電力消費は、ビットにつき0.7回の遷移である(すなわち、 $3.5 / 5$)。したがって、クロックおよびデータ情報を伝えるためのデータビット反転のこの新しい方法は、図81aおよび81bの実施例と比較して、より大きい帯域幅およびより低い電力消費をもたらす。加えて、この技法では、全てのデータスロットを依然として個々に駆動できることに留意されたい。

【0670】

図81bおよび81cのスキームは、高い帯域幅を使用し、低い電力消費を有し、また、情報を伝送するために単一のワイヤまたは媒体を使用する、バスシステムに適している。他の実施形態では、伝送媒体は、光学または無線であり得る。PLLが、典型的には、データのストリームからクロック情報を抽出するためにデバイスの受信機で使用されるであろうことに留意されたい。さらに、III型またはIV型位相検出器等の単一のクロックエッジ(例えば、低・高または高・低)遷移を利用するPLLは、全データ群に対する遷移を見出すために、XORゲートとともに小遅延ラインを利用し得る。この用途では、遅延は、便宜上、1つのデータスロットにほぼ等しくあり得、直接および遅延信号は、XORゲートに供給され得、XORゲートからの出力は、位相検出器に提供され得る。

40

【0671】

また、ソフトウェアを介して実装される、本明細書で説明される統一バス通信プロトコルの方法および関連同期化技法のうちの少なくとも1つを行うために使用される、要素の

50

うちの少なくともいくつかは、オブジェクト指向プログラミング等の高レベルプロシージャ言語で書かれ得ることに留意されたい。したがって、プログラムコードは、C、C⁺、または任意の他の好適なプログラミング言語で書かれ得、オブジェクト指向プログラミングの当業者に公知であるように、モジュールまたはクラスを備え得る。代替として、またはそれに加えて、ソフトウェアを介して実装されるこれらの要素のうちの少なくともいくつかは、必要に応じて、アセンブリ言語、機械言語、またはファームウェアで書かれ得る。いずれにしても、プログラムコードは、記憶媒体上に、またはプロセッサ、オペレーティングシステム、および本明細書で説明される統一バス通信プロトコルの方法のうちの少なくとも1つの機能性を実装するために必要である関連ハードウェアおよびソフトウェアを有する、汎用または専用コンピュータデバイスによって読み取り可能である、コンピュータ読み取り可能な媒体上に記憶することができる。プログラムコードは、プロセッサによって読み取られるとき、統一バス通信プロトコルについて本明細書で説明される方法のうちの少なくとも1つを行うために、新しい特定かつ所定の様式で動作するように、プロセッサを構成する。

10

【0672】

さらに、本明細書で説明される方法のうちの少なくともいくつかは、1つ以上のプロセッサのためのコンピュータ使用可能命令を持つ、コンピュータ読み取り可能な媒体を備える、コンピュータプログラム製品の中で分散されることが可能である。少なくともいくつかの場合において、コンピュータ読み取り可能な媒体は、非一過性であり得る。媒体は、限定されないが、1つ以上のディスク、コンパクトディスク、テープ、チップ、USBキー、外部ハードドライブ、有線伝送、衛星伝送、インターネット伝送またはダウンロード、磁気および電子記憶媒体、デジタルおよびアナログ信号等の種々の形態で提供され得る。コンピュータ使用可能命令はまた、コンパイラ型および非コンパイラ型コードを含む、種々の形態であり得る。

20

【0673】

本明細書で説明される出願者の教示は、例証目的で種々の実施形態と関連するが、出願者の教示がそのような実施形態に限定されることは意図されない。反対に、本明細書で説明および例証される出願者の教示は、その一般的範囲が添付の請求項で定義される、実施形態から逸脱することなく、種々の代替案、修正等を包含する。

【図1】

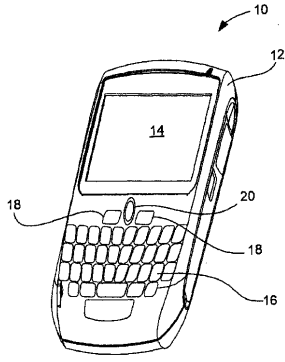


Figure 1

【図2】

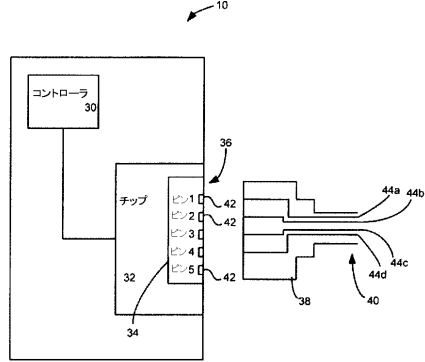


Figure 2

【図3a】

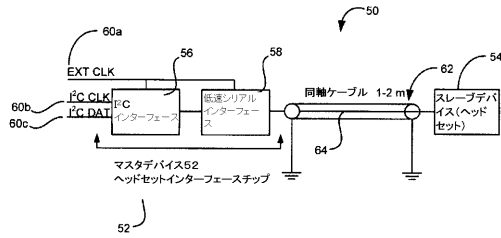


Figure 3a

【図3b】

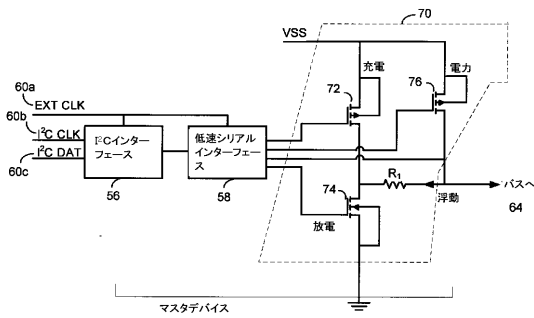


Figure 3b

【図3c】

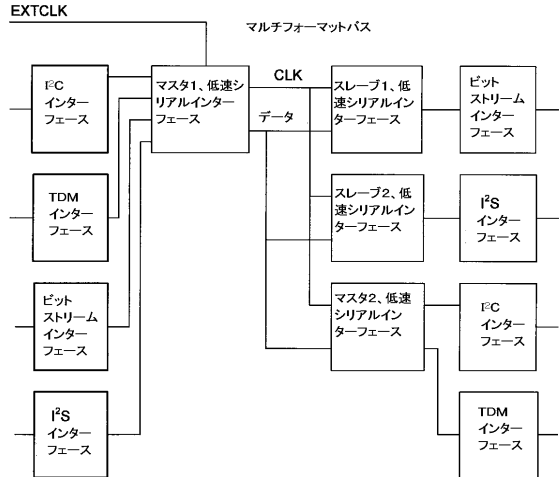


Figure 3c

【図4a】

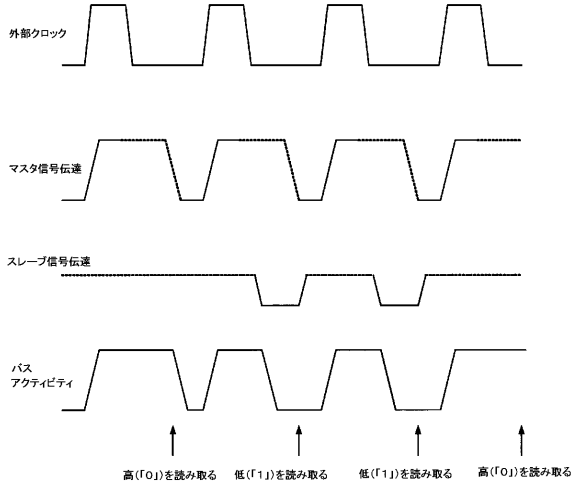


Figure 4a

【図4b】

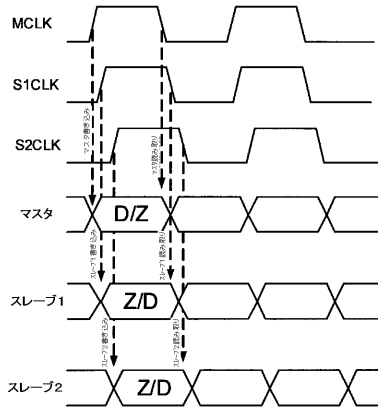


Figure 4b

【図5】

| Sコマンドワード | | | | | | | | | | | | | | | | |
|----------|---------|---------|--------------------------|-----|-----|-----|----|----|----|----|----|-------------------------|----|----|----------|----|
| ビット | S15 | S14 | S13 | S12 | S11 | S10 | S9 | S8 | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |
| フィールド | PAR (w) | ACK (r) | CONSTANT SYNC SYMBOL (w) | | | | | | | | | DYNAMIC SYNC SYMBOL (w) | | | IRGS (w) | |

Figure 5

【図6】

| ビット | 0-15 | 16-31 | 32-47 | 48-63 | 64-79 | 80-95 | 96-111 |
|-------|------|--------|-------|--------|-------|--------|--------|
| コンテンツ | S | オーディオ1 | X | オーディオ1 | Y | オーディオ1 | S |

Figure 6

マスタが書き込む: S15
 スleepが応答する: S14

【図7】

| 最後のフレーム | 応答すべきデバイス | | ACK値へのマスタアクション | | ACK値へのスleepアクション | |
|------------------|--|---|---|---|---|--------------------------------|
| | パリティ | ACK | ACK=1 | ACK=0 | ACK=1 | ACK=0 |
| PING | | バスにアタッチされた全てのデバイスが、NACKでパリティ不具合に反応する。 | このフレームにおいてバスエラーがあった。慎重にデータを扱い、バスエラーを信号伝達する。 | 全てOK、有効フレーム | デバイス自体が連続して2つのスleep信号伝達している場合、状態レベルを「10」に設定し、そこでなければ無視する。 | 全てOK、有効フレーム |
| FUNCTION (データなし) | マスタが、最後のパリティビットを含む最後のフレーム中のバス上の全てのビットに基づいて、パリティビットを書き込む。 | | | | | |
| FUNCTION (データあり) | | スleepアドレスがアドレス指定されたデバイスに合致する場合、パリティエラーであればNACKで返答する | | IO ERRORを信号伝達する。バスコントローラが、再度、IO動作を行うことによって、回復しようとするべきである。 | | 動作に応じて、READおよびWRITEを参照 |
| READ | | | 全てOK、有効データ | | 全てOK、有効データ | スleepは条件を無視するべきである。 |
| WRITE | | | | | | アドレス指定されたデバイスは、書き込みを続けるべきではない。 |

Figure 7

【図8】

| IRGS (Sワード) | SO DELAY (Reg. 0x11, B4) | フレームアクション | |
|-------------|--------------------------|----------------------|--|
| | | マスタコントローラがPINGを要求する | マスタコントローラが読み取り/書き込み動作を要求する |
| 0 | 0 | | |
| 0 | 1 | PINGを行う X15:13="000" | 読み取り/書き込みシーケンスを行う X15:X13≠'000' |
| 1 | 0 | | |
| 1 | 1 | | ピンツを行、元の読み取り/書き込みアクションが復元せられ、再度、次のフレーム中で再実行される。 X15:13="000" |

Figure 8

【図9a】

| Xコマンドワード | | | | | | | | | | | | | | | | | | | |
|----------|-------|-------|-------|------|---------------------------|-----|---------|--------------|----------|----------|---------|----|----|----|----|-------------------|--|--|--|
| タイプ | X15 | X14 | X13 | X12 | X11 | X10 | X9 | X8 | X7 | X6 | X5 | X4 | X3 | X2 | X1 | X0 | | | |
| PING | 0 (w) | 0 (w) | 0 (w) | | DS0 | DS1 | REQ BUS | REL BUS | ST11 (r) | ST10 (r) | ST9 (r) | | | | | ST8 (r) | | | |
| FUNC | 0 (w) | 0 (w) | 1 (w) | | DEV3-0 (w) | | | FUNCTION (w) | | | | | | | | | | | |
| その他 | 0 (w) | 1 (w) | 0 (w) | | 将来拡張値はスleepによって無視されるべきである | | | | | | | | | | | | | | |
| WRB | 1 (w) | 0 (w) | 0 (w) | BANK | DEV3-0 (w) | | | | | | | | | | | ADDRESS (7-0) (w) | | | |
| RDB | 1 (w) | 0 (w) | 1 (w) | | DEV3-0 (w) | | | | | | | | | | | ADDRESS (7-0) (w) | | | |
| WR16 | 1 (w) | 1 (w) | 0 (w) | | DEV3-0 (w) | | | | | | | | | | | ADDRESS (7-0) (w) | | | |
| RD16 | 1 (w) | 1 (w) | 1 (w) | | DEV3-0 (w) | | | | | | | | | | | ADDRESS (7-0) (w) | | | |

Figure 9a

【図9b】

| Xコマンドワード | | | | | | | | | | | | | | | | | | | |
|----------|-------|-------|-------|-----|---------------------------|-----|-----|--------------|----------|----------|---------|----|----|----|----|-------------------|--|--|--|
| タイプ | X15 | X14 | X13 | X12 | X11 | X10 | X9 | X8 | X7 | X6 | X5 | X4 | X3 | X2 | X1 | X0 | | | |
| PING | 0 (w) | 0 (w) | 0 (w) | | DS1 | DS2 | DS3 | BANK | ST11 (r) | ST10 (r) | ST9 (r) | | | | | ST8 (r) | | | |
| FUNC | 0 (w) | 0 (w) | 1 (w) | | DEV3-0 (w) | | | FUNCTION (w) | | | | | | | | | | | |
| その他 | 0 (w) | 1 (w) | 0 (w) | | 将来拡張値はスleepによって無視されるべきである | | | | | | | | | | | | | | |
| WRB | 1 (w) | 0 (w) | 0 (w) | DS0 | DEV3-0 (w) | | | | | | | | | | | ADDRESS (7-0) (w) | | | |
| RDB | 1 (w) | 0 (w) | 1 (w) | | DEV3-0 (w) | | | | | | | | | | | ADDRESS (7-0) (w) | | | |
| WR16 | 1 (w) | 1 (w) | 0 (w) | | DEV3-0 (w) | | | | | | | | | | | ADDRESS (7-0) (w) | | | |
| RD16 | 1 (w) | 1 (w) | 1 (w) | | DEV3-0 (w) | | | | | | | | | | | ADDRESS (7-0) (w) | | | |

Figure 9b

【図10】

| Yコマンドワード | | | | | | | | | | | | | | | | |
|----------|---------------------------|---------|---------|---------|---------|---------|------------|---------|----|----|----|----|----|----|----|----|
| タイプ | Y15 | Y14 | Y13 | Y12 | Y11 | Y10 | Y9 | Y8 | Y7 | Y6 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |
| PING | ST7 (r) | ST6 (r) | ST5 (r) | ST4 (r) | ST3 (r) | ST2 (r) | ST1 (r) | ST0 (r) | | | | | | | | |
| FUNC | FUNCTION DATA (r/w) | | | | | | | | | | | | | | | |
| その他 | 将来拡張 値はスレーブによって無視されるべきである | | | | | | | | | | | | | | | |
| WRS | ADDR 7-0 (w) | | | | | | WD 7-0 (w) | | | | | | | | | |
| RD8 | | | | | | | RD 7-0 (r) | | | | | | | | | |
| WR16 | WD15-0 (w) | | | | | | | | | | | | | | | |
| RD16 | RD15-0 (r) | | | | | | | | | | | | | | | |

Figure 10

【図11b】

| データ | SEL_IN(N) | SEL_OUT(N) | 位置 | 関数 |
|----------|-----------|------------|----|-------------|
| ビットストリーム | 0 | 0 | 左 | 2チャンネルバス |
| ビットストリーム | 0 | 1 | 右 | 2チャンネルバス |
| バス | 0 | 0 | 0 | マルチフォーマットバス |
| バス | 0 | 1 | 1 | マルチフォーマットバス |
| バス | 1 | 0 | 2 | マルチフォーマットバス |
| バス | 1 | 1 | 3 | マルチフォーマットバス |
| バス | 浮動 | 0 | 4 | マルチフォーマットバス |
| バス | 浮動 | 1 | 5 | マルチフォーマットバス |
| バス | 0 | 浮動 | 6 | マルチフォーマットバス |
| バス | 1 | 浮動 | 7 | マルチフォーマットバス |
| バス | 浮動 | 浮動 | 8 | マルチフォーマットバス |

Figure 11b

【図11a】

| X3 | X2 | X1 | X0 | 関数 | プロトキャストをサポートする |
|----|----|----|----|------------------|----------------|
| 0 | 0 | 0 | 0 | デバイスIDを要求する | はい |
| 0 | 0 | 0 | 1 | ハードリセット | はい |
| 0 | 0 | 1 | 0 | ソフトリセット | はい |
| 0 | 0 | 1 | 1 | IRQソースを読み取る | いいえ |
| 0 | 1 | 0 | 0 | デバイス自己試験を行う | はい |
| 0 | 1 | 0 | 1 | 能力情報を要求する | いいえ |
| 0 | 1 | 1 | 0 | 相対校正データを要求する | いいえ |
| 0 | 1 | 1 | 1 | 絶対校正データを要求する | いいえ |
| 1 | 0 | 0 | 0 | アドレスを読み取って増分する | いいえ |
| 1 | 0 | 0 | 1 | アドレスを書き込んで増分する | いいえ |
| 1 | 0 | 1 | 0 | アドレスで関数を実行する | はい |
| 1 | 0 | 1 | 1 | 間接的にアドレスで関数を実行する | はい |
| 1 | 1 | 0 | 0 | 利得を設定する | はい |
| 1 | 1 | 0 | 1 | 遅延を設定する | はい |
| 1 | 1 | 1 | 0 | 電力を設定する | はい |
| 1 | 1 | 1 | 1 | 将来拡張のために保留されている | |

Figure 11a

【図11c】

| データ | SEL_IN(N) | SEL_OUT(N) | 位置 | 関数 |
|----------|--------------|--------------|-----------|-----------------|
| ビットストリーム | 0 | 0 | 左 | 2チャンネルバス |
| ビットストリーム | 0 | 1 | 右 | 2チャンネルバス |
| バス | 0 | 0 | 0 | マルチフォーマットバス |
| バス | 0 | 1 | 1 | マルチフォーマットバス |
| バス | 1 | 0 | 2 | マルチフォーマットバス |
| バス | 1 | 1 | 3 | マルチフォーマットバス |
| バス | 0 | SEL_IN(1) | 0 (第1の要素) | マルチフォーマットバス/リング |
| バス | SEL_OUT(N-1) | SEL_IN (N+1) | N (中間要素) | マルチフォーマットバス/リング |
| バス | SEL_OUT(N-1) | 0 | N (最後の要素) | マルチフォーマットバス/リング |
| X | Z | 1 | | 試験 |

Figure 11c

【図11d】

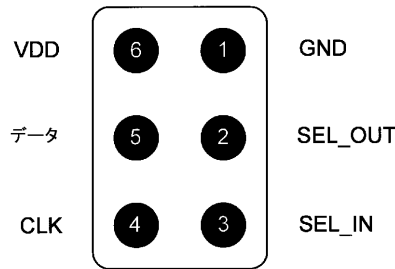


Figure 11d

【図11e】

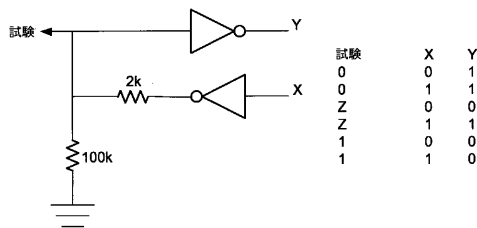


Figure 11e

【図15】

| B3 | B2 | B1 | B0 | IRQソース | エラー情報 |
|------------|----|----|----|---------------------|----------------------------|
| 0 | 0 | 0 | 0 | エラー無し | 全てゼロ |
| 0 | 0 | 0 | 1 | スレーブIRQ高レベル | なし(スレーブ状態レジスタを読み取る) |
| 0 | 0 | 1 | 0 | スレーブIRQ高レベル | なし(スレーブ状態レジスタを読み取る) |
| 0 | 0 | 1 | 1 | マルチマスタ要求 | なし(全てゼロ) |
| 0 | 1 | 0 | 0 | デバイス自己試験 | デバイス依存性(ゼロ=エラーなし) |
| 0 | 1 | 0 | 1 | 7/9/10エラー | 検出以降のフレームの数 |
| 0 | 1 | 1 | 0 | バス衝突 | フレームの内側の破壊ビット |
| 0 | 1 | 1 | 1 | スレーブ同期損失 | 同期の損失前のスレーブ数 |
| 1 | 0 | 0 | 0 | 関数 | サポートされていない関数 (reg. 0x0001) |
| 1 | 0 | 0 | 1 | 過熱 | 摂氏単位の温度(随機的) |
| 任意の位の組み合わせ | | | | 将来拡張(ソースおよび情報を集積する) | |

Figure 15

【図12】

| レジスタ | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|--------|-----|---------|----|----|-------|----|----|----|
| 0x0000 | タイプ | チャンネル情報 | | | データ形式 | | | |

Figure 12

【図16a】

| レジスタ | 目的 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|--------|--------|----------------------|---------|------------|-----------|-----------|--------|----|----|
| 0x0000 | データポート | DATA (7:0) | | | | | | | |
| 0x0001 | | DATA (15:8) | | | | | | | |
| 0x0002 | 設定ポート | CH-CHANNEL SELECTION | | | | | | | |
| 0x0003 | | ACTIVATE | PREPARE | IRQ23 MASK | IRQF MASK | PORT | | | |
| 0x0004 | | START | | | | | | | |
| 0x0005 | | SYNCHRONIZE | | SUBGROUP | | | REPEAT | | |
| 0x0006 | | SKIP(1:0) | | | | LENGTH | | | |
| 0x0007 | | PCLKD | | | | SKIP(5:2) | | | |

Figure 16a

【図13】

| B3 | B2 | B1 | B0 | データ形式 |
|----|------------|----|----|---------------------------------|
| 0 | 0 | 0 | 0 | この値はポート情報の終了を示すために保留されている |
| 0 | 全ての他の組み合わせ | | | 将来拡張 |
| 1 | 0 | 0 | 0 | バイナリ、2の補数 |
| 1 | 0 | 0 | 1 | ビットストリーム |
| 1 | 0 | 1 | 0 | 文字列 |
| 1 | 0 | 1 | 1 | ファイル |
| 1 | 1 | 0 | 0 | 浮動小数点、IEEE 754-2008単精度バイナリ |
| 1 | 1 | 0 | 1 | 浮動小数点、IEEE 754-2008倍精度バイナリ |
| 1 | 1 | 1 | 0 | 浮動小数点、IEEE 754-2008 decimal32形式 |
| 1 | 1 | 1 | 1 | 浮動小数点、IEEE 754-2008 decimal64形式 |

Figure 13

【図14】

| レジスタ | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|--------|-------------|----|----|--------|----|----|----|----|
| 0x0000 | エラー情報(3:0) | | | エラーコード | | | | |
| 0x0001 | エラー情報(11:0) | | | | | | | |

Figure 14

【図16b】

| レジスタ | 目的 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|--------|--------|--------------------|-------------|--------|----|--------|----|----|----|
| 0x0000 | データポート | DATA (7:0) | | | | | | | |
| 0x0001 | | DATA (15:8) | | | | | | | |
| 0x0002 | 設定ポート | ACTIVATE | POWER LEVEL | | | PORT | | | |
| 0x0003 | | CHANNEL SELECTION | | | | | | | |
| 0x0004 | | START | | | | | | | |
| 0x0005 | | BANK MODE | BANK SELECT | LENGTH | | | | | |
| 0x0006 | | IRQ23 MASK | IRQF MASK | SKIP | | | | | |
| 0x0007 | | SYNCHRONIZE | SUBGROUP | | | REPEAT | | | |
| 0x0008 | | COMMAND SEPARATION | | | | | | | |

Figure 16b

【図17】

| B2 | B1 | B0 | 電力[%] | 機能性 |
|----|----|----|-------|---|
| 0 | 0 | 0 | <0.1 | 次のスレーム境界でのシャットダウン、クロック遷移によるS001へのモード変更 |
| 0 | 0 | 1 | 1 | 次のフレーム境界でのシャットダウン、SoundWireコマンドによるモード変更 |
| 0 | 1 | 0 | 5 | SLEEPモード、限定された機能性 |
| 0 | 1 | 1 | 50 | 完全機能性を伴う最低電力 |
| 1 | 0 | 0 | 75 | 低電力モード |
| 1 | 0 | 1 | 100 | 通常モード |
| 1 | 1 | 0 | 150 | 高性能モード |
| 1 | 1 | 1 | 200 | 最高性能モード |

Figure 17

【図18】

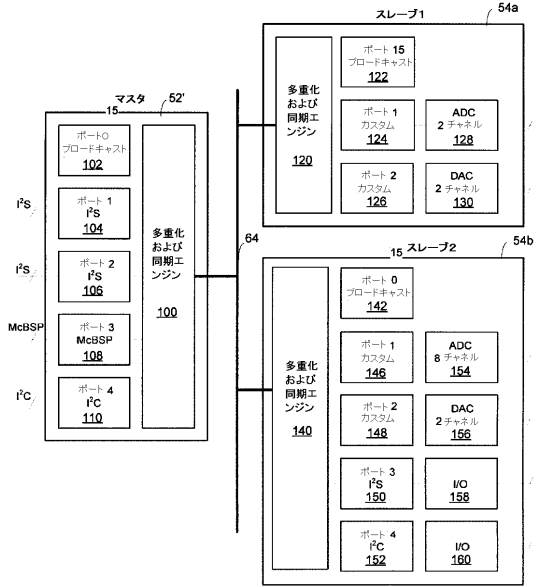


Figure 18

【図19】

| ビット | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-------|----|----|----|----|----|----|----|----|
| チャンネル | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Figure 19

【図20】

| BANK MODE | BANK SELECT | 読み取り/書き込み中に選択されるバンク | 読み取り/書き込みアドレス |
|-----------|-------------|---------------------|--|
| 0 | 0 | 1/O動作のMSBによって決定される | MSBはBANK選択によって与えられる。1/O動作中に与えられるマスタからのMSBは基調される。 |
| 0 | 1 | 0 | |
| 1 | 0 | 0 | 1/O動作中にマスタによって与えられる (8ビットまたは16ビットアドレス) |
| 1 | 1 | 1 | |

Figure 20

【図21】

| ビット | 0-15 | 16-31 | 32-47 | 48-63 | 64-79 | 80-95 |
|-------|-------------|--------|--------|-------|--------|--------|
| コンテンツ | S, X, または Y | オーディオ0 | オーディオ1 | 空 | オーディオ0 | オーディオ1 |

START=00000000
 REPEAT=010=2X1=2
 SUBGROUP=111 (無視)
 SKIP=000100
 CHANNEL SELECTION= 00000011

Figure 21

【図22a】

| 構成 | B7 | B6 | 同期化のソース | | | 機能 |
|----|----|----|--------------|--------------|----------|-------|
| | | | D50 | D51 | ポートプレアンプ | |
| 1 | 0 | 0 | | | | 等時性 |
| 2 | 0 | 1 | | | TX/RX | 非同期 |
| 3 | 1 | 0 | 位相加算器をリセットする | | | 分数同期化 |
| 4 | 1 | 1 | | 位相加算器をリセットする | | |

Figure 22a

【図22b】

| B3 | B2 | B1 | B0 | ポートクロック分割係数 |
|----|----|----|----|------------------|
| 0 | 0 | 0 | 0 | クロックが無効にされている(低) |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 2.5 |
| 0 | 1 | 0 | 0 | 3 |
| 0 | 1 | 0 | 1 | 4 |
| 0 | 1 | 1 | 0 | 5 |
| 0 | 1 | 1 | 1 | 6 |
| 1 | 0 | 0 | 0 | 6.5 |
| 1 | 0 | 0 | 1 | 8 |
| 1 | 0 | 1 | 0 | 10 |
| 1 | 0 | 1 | 1 | 12 |
| 1 | 1 | 0 | 0 | 12.5 |
| 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 1 | 0 | 16 |
| 1 | 1 | 1 | 1 | 24 |

Figure 22b

【図22c】

| 構成 | B7 | B6 | 同期化のソース | | | | 機能 |
|----|----|----|--------------|-----|----------------|-----|-------|
| | | | D50 | D51 | D52 | D53 | |
| 1 | 0 | 0 | | | | | 等時性 |
| 2 | 0 | 1 | | | | ポート | 非同期 |
| 3 | 1 | 0 | TXサンプルが提供される | | RXサンプルが受け入れられる | | 分数同期化 |
| 4 | 1 | 1 | マルチフレーム開始 | | | | |

Figure 22c

【 図 2 3 】

| B3 | グループ |
|----|------|
| 0 | X1 |
| 1 | X3 |

| B5 | B4 | グループ |
|----|----|------|
| 0 | 0 | X1 |
| 0 | 1 | X2 |
| 1 | 0 | X4 |
| 1 | 1 | X8 |

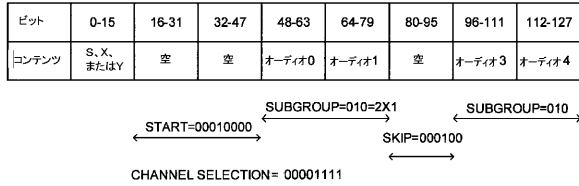


Figure 23

【 図 2 4 】

| B3 | チャンネルアクティブ化 |
|----|-------------|
| 0 | X1 |
| 1 | X3 |

| B2 | B1 | チャンネルアクティブ化 |
|----|----|-------------|
| 0 | 0 | X1 |
| 0 | 1 | X2 |
| 1 | 0 | X4 |
| 1 | 1 | X8 |

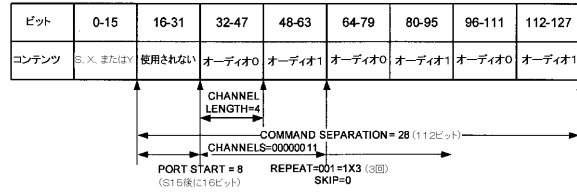


Figure 24

【 図 2 5 a 】

| COMMAND SEPARATION | | | | | | | | フレームタイプ |
|--------------------|------------------|----|----|----|------------|----|----|----------|
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | |
| 0 | ニブル単位のサブフレーム間の距離 | | | | | | | ワード |
| 1 | サブフレーム距離 図25b参照 | | | | ビットストリームの数 | | | ビットストリーム |

Figure 25a

【 図 2 5 b 】

| B6 | B5 | B4 | サブフレーム分離(ビット) | サブフレーム長(ビット) |
|----|----|----|---------------|--------------|
| 0 | 0 | 0 | 0 | 16 |
| 0 | 0 | 1 | 34 | 50 |
| 0 | 1 | 0 | 48 | 64 |
| 0 | 1 | 1 | 80 | 96 |
| 1 | 0 | 0 | 84 | 100 |
| 1 | 0 | 1 | 112 | 128 |
| 1 | 1 | 0 | 114 | 130 |
| 1 | 1 | 1 | 176 | 192 |

Figure 25b

【 図 2 5 c 】

| COMMAND SEPARATION | フィールドの解釈 | フレームタイプ |
|--------------------|------------------------------------|-------------|
| 0-239 | ニブル単位のサブフレーム間の距離 | ワードモード |
| 240-255 | ビット単位のサブフレーム間の距離(4つの下位ビットのみが使用される) | ビットストリームモード |

Figure 25c

【 図 2 6 a 】

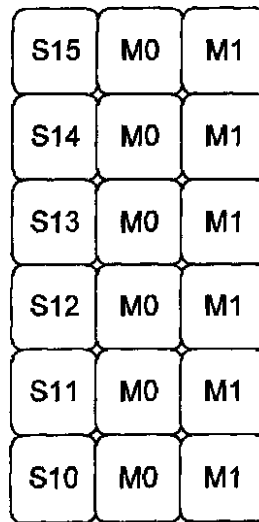


Figure 26a

【図26b】

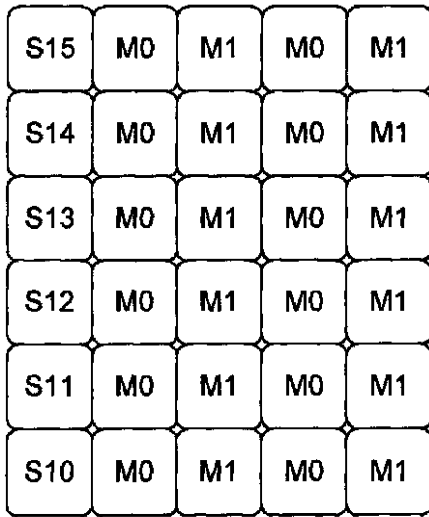


Figure 26b

【図28a】

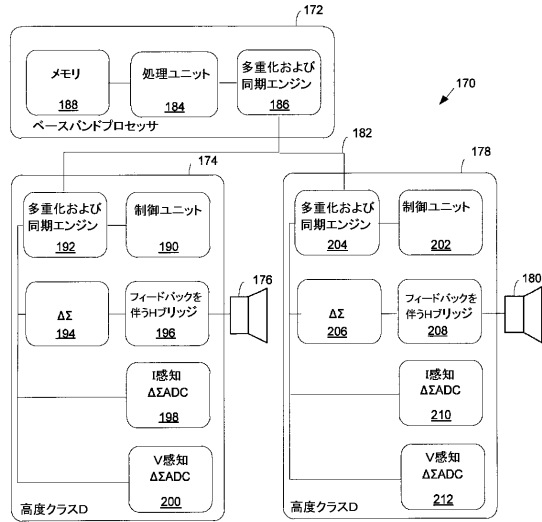


Figure 28a

【図27】

| | | | | | |
|---------|----|----|---------|----|----|
| B5 | B4 | B3 | B2 | B1 | B0 |
| VREPEAT | | | HREPEAT | | |

| LSB | 乗数 |
|-----|----|
| 0 | 1 |
| 1 | 3 |

| MSB | LSB+1 | 乗数 |
|-----|-------|----|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 4 |
| 1 | 1 | 8 |

Figure 27

【図28b】

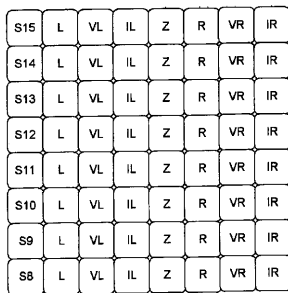


Figure 28b

【図28c】

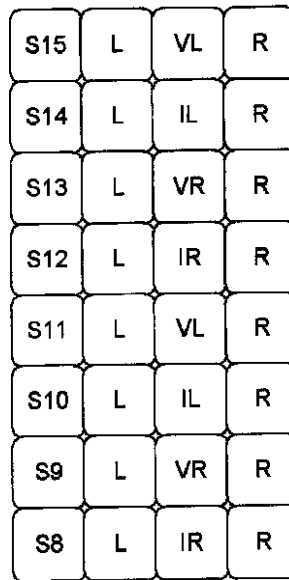


Figure 28c

【図28d】

| | | | | | |
|-----|---|---|----|---|---|
| S15 | L | R | VL | L | R |
| S14 | L | R | IL | L | R |
| S13 | L | R | VR | L | R |
| S12 | L | R | IR | L | R |
| S11 | L | R | VL | L | R |
| S10 | L | R | IL | L | R |
| S9 | L | R | VR | L | R |
| S8 | L | R | IR | L | R |

Figure 28d

【図29a】

| レジスタ | 目的 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | |
|--------|---------|------------------------------|---------------|----------------|-----------------------|-------------------|-----------------|----------------------|-----------------|--|
| 0x0008 | 設定ポート | COMMAND SEPARATION | | | | | | | | |
| 0x0009 | | MULTI FRAME | FIRST STAGE | OPTIONAL STAGE | 2 nd STAGE | THIRD STAGE | | | | |
| 0x000A | | FRACTIONAL (LOW) 6/9ビットをチェック | | | | | | | | |
| 0x000B | | FRACTIONAL (HIGH) | | | | | | | | |
| 0x000C | アドレスポート | ADDRESS (7:0) | | | | | | | | |
| 0x000D | | ADDRESS (15:8) | | | | | | | | |
| 0x000E | コマンドポート | FUNCTION | | | | | | | | |
| 0x000F | | FRAME CONTROL | CURRENT BANK | DEVICE ADDRESS | | | | | | |
| 0x0010 | | MCLKD | | | CLOCK SOURCE | ENABLE MULTI MAST | MASTER IRQ MASK | REQUEST MULTI MASTER | | |
| 0x0011 | | BUS ERROR MASK | IO ERROR MASK | ATTENTION MASK | SO DELAY ENABLE | ATTACHMENT MASK | IOC MASK | COMMAND MASK | FRAME DONE MASK | |
| 0x0012 | | BUS ERROR | IO ERROR | STATUS1 | STATUS0 | ATTACHMENT | IOC | COMMAND ERROR | FRAME DONE | |
| 0x0013 | | SLAVE STATUS | | | | | | | | |
| 0x0014 | | SLAVE STATUS | | | | | | | | |
| 0x0015 | | SLAVE STATUS | | | | | | | | |

Figure 29a

【図29b】

| レジスタ | 目的 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | |
|--------|----------------|---------------------|---------------|----------------|-----------------|-----------------|----------|---------------|-----------------|--|
| 0x009 | アドレスポート | SYS FREQ/FRAME RATE | | | | | | | | |
| 0x00A | | OVERSAMPLE (X) | | | | | | | | |
| 0x00B | | OVERSAMPLE (Y) | | | | | | | | |
| 0x00C | | ADDRESS (7:0) | | | | | | | | |
| 0x00D | ADDRESS (15:8) | | | | | | | | | |
| 0x00E | コマンドポート | FUNCTION | | | | | | | | |
| 0x00F | | FRAME CONTROL | MULTI MASTER | DEVICE ADDRESS | | | | | | |
| 0x0010 | | ENABLE MULTI | CURR BANK | POWER | CLOCK SOURCE | CLOCK DIV | | | | |
| 0x0011 | | BUS ERROR MASK | IO ERROR MASK | ATTENTION MASK | SO DELAY ENABLE | ATTACHMENT MASK | IOC MASK | COMMAND MASK | FRAME DONE MASK | |
| 0x0012 | | BUS ERROR | IO ERROR | STATUS1 | STATUS0 | ATTACHMENT | IOC | COMMAND ERROR | FRAME DONE | |
| 0x0013 | | SLAVE STATUS | | | | | | | | |
| 0x0014 | | SLAVE STATUS | | | | | | | | |
| 0x0015 | | SLAVE STATUS | | | | | | | | |

Figure 29b

【図29d】

| B7 | B6 | 比率 |
|----|----|------|
| 0 | 0 | 1 |
| 0 | 1 | 5 |
| 1 | 0 | 13 |
| 1 | 1 | 将来拡張 |

Figure 29d

【図29e】

| B5 | 比率 |
|----|-----|
| 0 | 1 |
| 1 | 1/3 |

Figure 29e

【図29c】

| B7 | B6 | 比率 |
|----|----|----|
| 0 | 0 | 13 |
| 0 | 1 | 16 |
| 1 | 0 | 24 |
| 1 | 1 | 25 |

Figure 29c

【図 29 f】

| B4 | B3 | 比率 |
|----|----|----|
| 0 | 0 | 2 |
| 0 | 1 | 4 |
| 1 | 0 | 8 |
| 1 | 1 | 16 |

Figure 29f

【図 29 g】

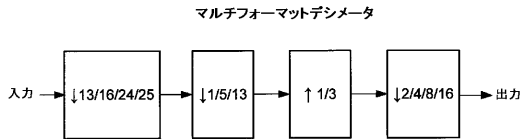


Figure 29g

【図 29 h】

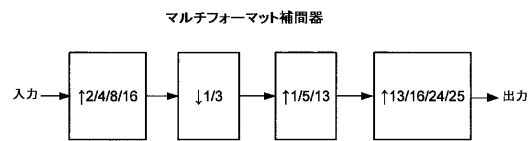


Figure 29h

【図 33 a】

| B3 | B2 | B1 | B0 | ポートクロック分割係数 |
|----|----|----|----|------------------|
| 0 | 0 | 0 | 0 | クロックが無効にされている(低) |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 2.5 |
| 0 | 1 | 0 | 0 | 3 |
| 0 | 1 | 0 | 1 | 4 |
| 0 | 1 | 1 | 0 | 5 |
| 0 | 1 | 1 | 1 | 6 |
| 1 | 0 | 0 | 0 | 6.5 |
| 1 | 0 | 0 | 1 | 8 |
| 1 | 0 | 1 | 0 | 10 |
| 1 | 0 | 1 | 1 | 12 |
| 1 | 1 | 0 | 0 | 12.5 |
| 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 1 | 0 | 16 |
| 1 | 1 | 1 | 1 | 24 |

Figure 33a

【図 33 b】

| 分割率 | | | | EXTCLK/単一ワイヤバスクロック | EXTCLK/2ワイヤバスクロック |
|-----------|----|----|----|--------------------|-------------------|
| B3 | B2 | B1 | B0 | クロック出力が無効にされている | |
| 0 | 0 | 0 | 0 | クロック出力が無効にされている | |
| 0 | 0 | 0 | 1 | 2 | 1 |
| 0 | 0 | 1 | 0 | 4 | 2 |
| 0 | 0 | 1 | 1 | 6 | 3 |
| 0100-1110 | | | | 8-28 | 4-14 |
| 1 | 1 | 1 | 1 | 30 | 15 |

Figure 33b

【図 34 a】

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|------------------------|-----------------------|------------------------|-------------------------|-------------------------|------------------|----------------------|-------------------------|
| BUS ERROR MASK (読取/書込) | IO ERROR MASK (読取/書込) | ATTENTION MASK (読取/書込) | SO DELAY ENABLE (読取/書込) | ATTACHMENT MASK (読取/書込) | IOC MASK (読取/書込) | COMMAND MASK (読取/書込) | FRAME DONE MASK (読取/書込) |

Figure 34a

【図 29 i】

| B7 | B6 | B5 | フレーム動作 |
|----|----|----|--------------|
| 0 | 0 | 0 | PING |
| 0 | 0 | 1 | FUNCTION |
| 0 | 1 | 0 | 将来拡張 |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | WRITE 8 ビット |
| 1 | 0 | 1 | READ 8 ビット |
| 1 | 1 | 0 | WRITE 16 ビット |
| 1 | 1 | 1 | READ 16 ビット |

Figure 29i

【図 30】

| 値 | WRITE CONTROLへのマスタ書き込み | B7からのマスタ読み取り |
|----|------------------------|------------------------|
| 00 | 動作無し | 書き込み動作が進行中ではない |
| 10 | 書き込み動作を開始する | 書き込み動作がアクティブであり完了していない |

Figure 30

【図 31】

| B7 | B6 | マスタ書き込みの結果 | マスタ読み取りの結果 |
|----|----|-------------|--------------------|
| 0 | 0 | 動作が開始されていない | 全ての動作が完了している |
| 0 | 1 | 読み取り動作を開始する | 読み取りが依然としてアクティブである |
| 1 | 0 | 書き込み動作を開始する | 書き込みが依然としてアクティブである |
| 1 | 1 | 関数を開始する | 関数が依然としてアクティブである |

Figure 31

【図 32 a】

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|--------|----|----|----|--------------|---------------------|-----------------|----------------------|
| MCLKID | | | | CLOCK SOURCE | ENABLE MULTI MASTER | MASTER IRQ MASK | REQUEST MULTI MASTER |

Figure 32a

【図 32 b】

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|--------------|-----------|-------|--------------|-------|-------|-------|-------|
| ENABLE MULTI | CURR BANK | POWER | CLOCK SOURCE | DIV 3 | DIV 2 | DIV 1 | DIV 0 |

Figure 32b

【図 34 b】

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|------------------------|-----------------------|------------------------|-------------------------|-------------------------|------------------|------------------|-------------------------|
| BUS ERROR MASK (読取/書込) | IO ERROR MASK (読取/書込) | ATTENTION MASK (読取/書込) | SO DELAY ENABLE (読取/書込) | ATTACHMENT MASK (読取/書込) | WRC MASK (読取/書込) | RDC MASK (読取/書込) | FRAME DONE MASK (読取/書込) |

Figure 34b

【図 35】

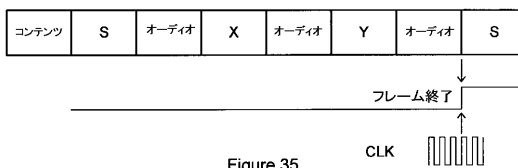


Figure 35

【図 36 a】

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|------------------|-----------------|----------------|----------------|-------------------|------------|----------------------|-------------------|
| BUS ERROR (読み取り) | IO ERROR (読み取り) | STATUS1 (読み取り) | STATUS0 (読み取り) | ATTACHMENT (読み取り) | IOC (読み取り) | COMMAND ERROR (読み取り) | FRAME DONE (読み取り) |

Figure 36a

【図 36 b】

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|------------------|-----------------|----------------|----------------|-------------------|------------|------------|-------------------|
| BUS ERROR (読み取り) | IO ERROR (読み取り) | STATUS1 (読み取り) | STATUS0 (読み取り) | ATTACHMENT (読み取り) | WRC (読み取り) | RDC (読み取り) | FRAME DONE (読み取り) |

Figure 36b

【図 37】

| Reg | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|--------|-------|-------|-------|-------|------|------|------|------|
| 0x0013 | ST3H | ST3L | ST2H | ST2L | ST1H | ST1L | ST0H | ST0L |
| 0x0014 | ST7H | ST7L | ST6H | ST6L | ST5H | ST5L | ST4H | ST4L |
| 0x0015 | ST11H | ST11L | ST10H | ST10L | ST9H | ST9L | ST8H | ST8L |

Figure 37

【図38a】

| フィールドタイプ | 符号化タイプ | 位置 | 製造業者名 (別表を参照) | | 部品番号 |
|--------------|--------|--------------|-------------------|-------------------|----------------|
| ビット番号 | B31 | B30-27 | B26-B22 | B21-B17 | B16-B0 |
| ビットの長さ | 1 | 4 | 5 | 5 | 17 |
| 符号化された 情報 | 0=圧縮 | 0=なし | 第1の 頭字語の 文字 | 第2の 頭字語の 文字 | 2進数で符号化された部品番号 |
| | 1=一般 | 1-15 = 位置 | 図38b参照 | | |

Figure 38a

【図38b】

| フィールドタイプ | 符号化タイプ | 位置 | 等級 | 改訂 | 部品番号 |
|----------|---------|-------------------|--------------------------|-------------|--------------|
| バイト番号 | | 第1 | | 第2 | 以降 |
| ビット番号 | B7 | B6-B3 | B2-B0 | B7-B4 | B3-B0 |
| ビットの長さ | 1 | 4 | 3 | 4 | 4 |
| 符号化 | 0=圧縮符号化 | 図38a参照 | | | |
| | 1=一般符号化 | 0=なし 1-15 = 位置 | 000-011 等級A-D 将来拡張 | バス バージョン | チップ バージョン |

Figure 38b

【図39】

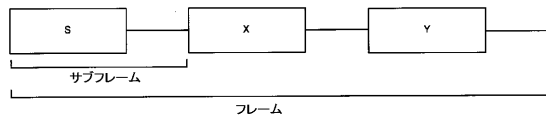


Figure 39

【図42】

| ビット | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----|-----|----|----|----|-----|----|----|----|-----|----|----|----|-----|----|----|----|
| 000 | PAR | M0 | M1 | M2 | ACK | M0 | M1 | M2 | 1 | M0 | M1 | M2 | 0 | M0 | M1 | M2 |
| 010 | 1 | M0 | M1 | M2 | 1 | M0 | M1 | M2 | 0 | M0 | M1 | M2 | 0 | M0 | M1 | M2 |
| 020 | 0 | M0 | M1 | M2 | 0 | M0 | M1 | M2 | 1 | M0 | M1 | M2 | CR3 | M0 | M1 | M2 |
| 030 | CR2 | M0 | M1 | M2 | CR1 | M0 | M1 | M2 | CR0 | M0 | M1 | M2 | IRQ | M0 | M1 | M2 |
| 040 | X15 | M0 | M1 | M2 | X14 | M0 | M1 | M2 | X13 | M0 | M1 | M2 | X12 | M0 | M1 | M2 |
| 050 | X11 | M0 | M1 | M2 | X10 | M0 | M1 | M2 | X9 | M0 | M1 | M2 | X8 | M0 | M1 | M2 |
| 060 | X7 | M0 | M1 | M2 | X6 | M0 | M1 | M2 | X5 | M0 | M1 | M2 | X4 | M0 | M1 | M2 |
| 070 | X3 | M0 | M1 | M2 | X2 | M0 | M1 | M2 | X1 | M0 | M1 | M2 | X0 | M0 | M1 | M2 |
| 080 | Y15 | M0 | M1 | M2 | Y14 | M0 | M1 | M2 | Y13 | M0 | M1 | M2 | Y12 | M0 | M1 | M2 |
| 090 | Y11 | M0 | M1 | M2 | Y10 | M0 | M1 | M2 | Y9 | M0 | M1 | M2 | Y8 | M0 | M1 | M2 |
| 0A0 | Y7 | M0 | M1 | M2 | Y6 | M0 | M1 | M2 | Y5 | M0 | M1 | M2 | Y4 | M0 | M1 | M2 |
| 0B0 | Y3 | M0 | M1 | M2 | Y2 | M0 | M1 | M2 | Y1 | M0 | M1 | M2 | Y0 | M0 | M1 | M2 |

Figure 42

【図40】

| ビット | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|-----|-----|-----|-----|-----|
| 000 | PAR | ACK | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | CR3 | CR2 | CR1 | CR0 | IRQ |
| 010 | X15 | X14 | X13 | X12 | X11 | X10 | X9 | X8 | X7 | X6 | X5 | X4 | X3 | X2 | X1 | X0 |
| 020 | Y15 | Y14 | Y13 | Y12 | Y11 | Y10 | Y9 | Y8 | Y7 | Y6 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |

Figure 40

【図41】

オーディオチャネルワード
オーディオチャネルワード

| ビット | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|-----|-----|-----|-----|-----|
| 000 | PAR | ACK | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | CR3 | CR2 | CR1 | CR0 | IRQ |
| 010 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| 020 | B15 | B14 | B13 | B12 | B11 | B10 | B9 | B8 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| 030 | C15 | C14 | C13 | C12 | C11 | C10 | C9 | C8 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
| 040 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 050 | E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| 060 | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z |
| 070 | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z |
| 080 | X15 | X14 | X13 | X12 | X11 | X10 | X9 | X8 | X7 | X6 | X5 | X4 | X3 | X2 | X1 | X0 |
| 090 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| 0A0 | B15 | B14 | B13 | B12 | B11 | B10 | B9 | B8 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| 0B0 | C15 | C14 | C13 | C12 | C11 | C10 | C9 | C8 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
| 0C0 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 0D0 | E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| 0E0 | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z |
| 0F0 | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z |
| 100 | Y15 | Y14 | Y13 | Y12 | Y11 | Y10 | Y9 | Y8 | Y7 | Y6 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |
| 110 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| 120 | B15 | B14 | B13 | B12 | B11 | B10 | B9 | B8 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| 130 | C15 | C14 | C13 | C12 | C11 | C10 | C9 | C8 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
| 140 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 150 | E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| 160 | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z |
| 170 | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z |

Figure 41

【図43】

| ビット | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----|-----|----|----|----|----|----|-----|-----|-----|----|----|----|----|----|-----|-----|
| 000 | PAR | M0 | M1 | M2 | M3 | M4 | A15 | B15 | ACK | M0 | M1 | M2 | M3 | M4 | A14 | B14 |
| 010 | 1 | M0 | M1 | M2 | M3 | M4 | A13 | B13 | 0 | M0 | M1 | M2 | M3 | M4 | A12 | B12 |
| 020 | 1 | M0 | M1 | M2 | M3 | M4 | A11 | B11 | 1 | M0 | M1 | M2 | M3 | M4 | A10 | B10 |
| 030 | 0 | M0 | M1 | M2 | M3 | M4 | A9 | B9 | 0 | M0 | M1 | M2 | M3 | M4 | A8 | B8 |
| 040 | 0 | M0 | M1 | M2 | M3 | M4 | A7 | B7 | 0 | M0 | M1 | M2 | M3 | M4 | A6 | B6 |
| 050 | 1 | M0 | M1 | M2 | M3 | M4 | A5 | B5 | CR3 | M0 | M1 | M2 | M3 | M4 | A4 | B4 |
| 060 | CR2 | M0 | M1 | M2 | M3 | M4 | A3 | B3 | CR1 | M0 | M1 | M2 | M3 | M4 | A2 | B2 |
| 070 | CR0 | M0 | M1 | M2 | M3 | M4 | A1 | B1 | IRQ | M0 | M1 | M2 | M3 | M4 | A0 | B0 |
| 080 | X15 | M0 | M1 | M2 | M3 | M4 | A15 | B15 | X14 | M0 | M1 | M2 | M3 | M4 | A14 | B14 |
| 090 | X13 | M0 | M1 | M2 | M3 | M4 | A13 | B13 | X12 | M0 | M1 | M2 | M3 | M4 | A12 | B12 |
| 0A0 | X11 | M0 | M1 | M2 | M3 | M4 | A11 | B11 | X10 | M0 | M1 | M2 | M3 | M4 | A10 | B10 |
| 0B0 | X9 | M0 | M1 | M2 | M3 | M4 | A9 | B9 | X8 | M0 | M1 | M2 | M3 | M4 | A8 | B8 |
| 0C0 | X7 | M0 | M1 | M2 | M3 | M4 | A7 | B7 | X6 | M0 | M1 | M2 | M3 | M4 | A6 | B6 |
| 0D0 | X5 | M0 | M1 | M2 | M3 | M4 | A5 | B5 | X4 | M0 | M1 | M2 | M3 | M4 | A4 | B4 |
| 0E0 | X3 | M0 | M1 | M2 | M3 | M4 | A3 | B3 | X2 | M0 | M1 | M2 | M3 | M4 | A2 | B2 |
| 0F0 | X1 | M0 | M1 | M2 | M3 | M4 | A1 | B1 | X0 | M0 | M1 | M2 | M3 | M4 | A0 | B0 |
| 100 | Y15 | M0 | M1 | M2 | M3 | M4 | A15 | B15 | Y14 | M0 | M1 | M2 | M3 | M4 | A14 | B14 |
| 110 | Y13 | M0 | M1 | M2 | M3 | M4 | A13 | B13 | Y12 | M0 | M1 | M2 | M3 | M4 | A12 | B12 |
| 120 | Y11 | M0 | M1 | M2 | M3 | M4 | A11 | B11 | Y10 | M0 | M1 | M2 | M3 | M4 | A10 | B10 |
| 130 | Y9 | M0 | M1 | M2 | M3 | M4 | A9 | B9 | Y8 | M0 | M1 | M2 | M3 | M4 | A8 | B8 |
| 140 | Y7 | M0 | M1 | M2 | M3 | M4 | A7 | B7 | Y6 | M0 | M1 | M2 | M3 | M4 | A6 | B6 |
| 150 | Y5 | M0 | M1 | M2 | M3 | M4 | A5 | B5 | Y4 | M0 | M1 | M2 | M3 | M4 | A4 | B4 |
| 160 | Y3 | M0 | M1 | M2 | M3 | M4 | A3 | B3 | Y2 | M0 | M1 | M2 | M3 | M4 | A2 | B2 |
| 170 | Y1 | M0 | M1 | M2 | M3 | M4 | A1 | B1 | Y0 | M0 | M1 | M2 | M3 | M4 | A0 | B0 |

Figure 43

【図44】

| ビット | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 000 | PAR | M0 | M1 | M2 | M3 | M4 | C15 | D15 | ACK | M0 | M1 | M2 | M3 | M4 | C14 | D14 |
| 010 | 1 | M0 | M1 | M2 | M3 | M4 | C13 | D13 | 0 | M0 | M1 | M2 | M3 | M4 | C12 | D12 |
| 020 | 1 | M0 | M1 | M2 | M3 | M4 | C11 | D11 | 1 | M0 | M1 | M2 | M3 | M4 | C10 | D10 |
| 030 | 0 | M0 | M1 | M2 | M3 | M4 | C9 | D9 | 0 | M0 | M1 | M2 | M3 | M4 | C8 | D8 |
| 040 | 0 | M0 | M1 | M2 | M3 | M4 | C7 | D7 | 0 | M0 | M1 | M2 | M3 | M4 | C6 | D6 |
| 050 | 1 | M0 | M1 | M2 | M3 | M4 | C5 | D5 | CR3 | M0 | M1 | M2 | M3 | M4 | C4 | D4 |
| 060 | CR2 | M0 | M1 | M2 | M3 | M4 | C3 | D3 | CR1 | M0 | M1 | M2 | M3 | M4 | C2 | D2 |
| 070 | CR0 | M0 | M1 | M2 | M3 | M4 | C1 | D1 | IRQ | M0 | M1 | M2 | M3 | M4 | C0 | D0 |
| 080 | X15 | M0 | M1 | M2 | M3 | M4 | A15 | B15 | X14 | M0 | M1 | M2 | M3 | M4 | A14 | B14 |
| 090 | X13 | M0 | M1 | M2 | M3 | M4 | A13 | B13 | X12 | M0 | M1 | M2 | M3 | M4 | A12 | B12 |
| 0A0 | DS0=1 | M0 | M1 | M2 | M3 | M4 | A11 | B11 | X10 | M0 | M1 | M2 | M3 | M4 | A10 | B10 |
| 0B0 | X9 | M0 | M1 | M2 | M3 | M4 | A9 | B9 | X8 | M0 | M1 | M2 | M3 | M4 | A8 | B8 |
| 0C0 | X7 | M0 | M1 | M2 | M3 | M4 | A7 | B7 | X6 | M0 | M1 | M2 | M3 | M4 | A6 | B6 |
| 0D0 | X5 | M0 | M1 | M2 | M3 | M4 | A5 | B5 | X4 | M0 | M1 | M2 | M3 | M4 | A4 | B4 |
| 0E0 | X3 | M0 | M1 | M2 | M3 | M4 | A3 | B3 | X2 | M0 | M1 | M2 | M3 | M4 | A2 | B2 |
| 0F0 | X1 | M0 | M1 | M2 | M3 | M4 | A1 | B1 | X0 | M0 | M1 | M2 | M3 | M4 | A0 | B0 |
| 100 | 1 | M0 | M1 | M2 | M3 | M4 | C15 | D15 | 0 | M0 | M1 | M2 | M3 | M4 | C14 | D14 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1C0 | 0 | M0 | M1 | M2 | M3 | M4 | A3 | B3 | 0 | M0 | M1 | M2 | M3 | M4 | A2 | B2 |
| 1D0 | 1 | M0 | M1 | M2 | M3 | M4 | A1 | B1 | CR3 | M0 | M1 | M2 | M3 | M4 | A0 | B0 |
| 1E0 | CR2 | M0 | M1 | M2 | M3 | M4 | C15 | D15 | CR1 | M0 | M1 | M2 | M3 | M4 | C14 | D14 |
| 1F0 | CR0 | M0 | M1 | M2 | M3 | M4 | C13 | D13 | IRQ | M0 | M1 | M2 | M3 | M4 | C12 | D12 |
| 200 | X15 | M0 | M1 | M2 | M3 | M4 | C11 | D11 | X14 | M0 | M1 | M2 | M3 | M4 | C10 | D10 |
| 210 | X13 | M0 | M1 | M2 | M3 | M4 | C9 | D9 | X12 | M0 | M1 | M2 | M3 | M4 | C8 | D8 |
| 220 | DS0=0 | M0 | M1 | M2 | M3 | M4 | C8 | D8 | X10 | M0 | M1 | M2 | M3 | M4 | C6 | D6 |

Figure 44

【図45a】

| EXTCLK=12.288 MHz | | | | | | | | |
|-------------------|-----------------|---------------|------------|----|-----------|---------|-----------|-----------------|
| オーディオチャンネルの数 | 出力サンプル周波数 [kHz] | サブフレーム長 [ビット] | WORD [ビット] | 回復 | 制御動作 | | EXTCLK 分割 | バスクロック周波数 [kHz] |
| | | | | | 周波数 [kHz] | 帯域幅 [%] | | |
| 1 | 48 | 64 | 16 | 3 | 5.3 | 25 | 12 | 1024 |
| 1 | 48 | 32 | 16 | 1 | 16 | 50 | 8 | 1536 |
| 2 | 48 | 128 | 28 | 2 | 8 | 12.5 | 4 | 3072 |
| 3 | 48 | 64 | 16 | 1 | 16 | 25 | 4 | 3072 |
| 3 | 48 | 256 | 20 | 4 | 4 | 6.3 | 4 | 3072 |
| 5 | 48 | 256 | 24 | 2 | 8 | 6.3 | 2 | 6144 |
| 6 | 48 | 256 | 20 | 2 | 8 | 6.3 | 2 | 6144 |
| 12 | 48 | 256 | 20 | 1 | 16 | 6.3 | 1 | 12.288 |

Figure 45a

【図46b】

| EXTCLK=19.200 MHz | | | | | | | | |
|-------------------|-----------------|-----------------|---------------|-----------|---------|-----------|-----------------|--|
| ビットストリームの総数 | オーディオビットストリームの数 | 出力サンプル周波数 [kHz] | オーバーサンプリングレート | 制御動作 | | EXTCLK 分割 | バスクロック周波数 [kHz] | |
| | | | | 周波数 [kHz] | 帯域幅 [%] | | | |
| 2 | 1 | 16 | 50 | 16.7 | 50 | 12 | 1600 | |
| 3 | 2 | 16 | 200/3 | 22.2 | 33.3 | 6 | 3200 | |
| 4 | 3 | 16 | 50 | 16.7 | 25 | 6 | 3200 | |
| 6 | 5 | 16 | 50 | 16.7 | 16.7 | 4 | 4800 | |
| 6 | 5 | 16 | 200/3 | 22.2 | 16.7 | 3 | 6400 | |
| 12 | 11 | 16 | 50 | 16.7 | 8.3 | 2 | 9600 | |

Figure 46b

【図47】

| EXTCLK=12.288 MHz | | | | | | | | | | |
|-------------------|-----------------|---------------|------------|----|--------------|-----------------|------------|-----|---------------|-----------------|
| オーディオチャンネルの数 | 出力サンプル周波数 [kHz] | サブフレーム長 [ビット] | WORD [ビット] | 回復 | オーディオチャンネルの数 | 出力サンプル周波数 [kHz] | WORD [ビット] | 回復 | 制御動作周波数 [kHz] | バスクロック周波数 [kHz] |
| | | | | | | | | | | |
| 2 | 48 | 128 | 16 | 3 | 2 | 8 | 16 | 1/2 | 5.3 | 2048 |
| 2 | 48 | 256 | 16 | 6 | 3 | 8 | 16 | 1 | 2.7 | 2048 |

Figure 47

【図48a】

| EXTCLK=12.000 MHz | | | | | | | | | |
|-------------------|------------|-----------------|------------|--------------|-----------------|------------|-----------|---------|-----------------|
| ビットストリームの総数 | ビットストリームの数 | 出力サンプル周波数 [kHz] | WORD [ビット] | オーディオチャンネルの数 | 出力サンプル周波数 [kHz] | WORD [ビット] | 制御動作 | | バスクロック周波数 [kHz] |
| | | | | | | | 周波数 [kHz] | 帯域幅 [%] | |
| 10 | 5 | 24 | 50 | 5 | 48 | 20 | 25 | 10 | 12000 |

Figure 48a

【図45b】

| EXTCLK=19.200 MHz | | | | | | | | |
|-------------------|-----------------|---------------|------------|----|-----------|---------|-----------|-----------------|
| オーディオチャンネルの数 | 出力サンプル周波数 [kHz] | サブフレーム長 [ビット] | WORD [ビット] | 回復 | 制御動作 | | EXTCLK 分割 | バスクロック周波数 [kHz] |
| | | | | | 周波数 [kHz] | 帯域幅 [%] | | |
| 4 | 48 | 400 | 24 | 4 | 4 | 4 | 4 | 4800 |
| 6 | 48 | 400 | 16 | 4 | 4 | 4 | 4 | 4800 |
| 8 | 48 | 400 | 24 | 2 | 8 | 4 | 2 | 9600 |
| 12 | 48 | 400 | 16 | 2 | 8 | 4 | 2 | 9600 |

Figure 45b

【図46a】

| EXTCLK=12.288 MHz | | | | | | | |
|-------------------|-----------------|-----------------|---------------|-----------|---------|-----------|-----------------|
| ビットストリームの総数 | オーディオビットストリームの数 | 出力サンプル周波数 [kHz] | オーバーサンプリングレート | 制御動作 | | EXTCLK 分割 | バスクロック周波数 [kHz] |
| | | | | 周波数 [kHz] | 帯域幅 [%] | | |
| 2 | 1 | 16 | 48 | 16 | 50 | 8 | 1536 |
| 2 | 1 | 16 | 64 | 21.3 | 50 | 6 | 2048 |
| 4 | 3 | 16 | 48 | 16 | 25 | 4 | 3072 |
| 4 | 3 | 16 | 64 | 21.3 | 25 | 3 | 4096 |
| 6 | 5 | 16 | 64 | 21.3 | 16.7 | 2 | 6144 |
| 8 | 7 | 16 | 48 | 16 | 12.5 | 2 | 6144 |
| 12 | 11 | 16 | 64 | 21.3 | 8.3 | 1 | 12.288 |
| 16 | 15 | 16 | 48 | 16 | 6.3 | 1 | 12.288 |

Figure 46a

【図48b】

| EXTCLK=12.288 MHz | | | | | | | | | |
|-------------------|-----------------|-----------------|-----|--------------|-----------------|------------|-----------|---------|-----------------|
| 総ビットストリームの数 | オーディオビットストリームの数 | 出力サンプル周波数 [kHz] | OSR | オーディオチャンネルの数 | 出力サンプル周波数 [kHz] | WORD [ビット] | 制御動作 | | バスクロック周波数 [kHz] |
| | | | | | | | 周波数 [kHz] | 帯域幅 [%] | |
| 4 | 1 | 16 | 48 | 2 | 48 | 16 | 48 | 25 | 3072 |
| 8 | 3 | 16 | 48 | 4 | 48 | 16 | 16 | 12.5 | 6144 |
| 8 | 3 | 16 | 48 | 2 | 48 | 32 | 48 | 12.5 | 6144 |
| 8 | 4 | 16 | 48 | 3 | 48 | 16 | 16 | 12.5 | 6144 |
| 8 | 4 | 16 | 48 | 3 | 32 | 24 | 16 | 12.5 | 6144 |
| 8 | 5 | 16 | 48 | 2 | 48 | 16 | 16 | 12.5 | 6144 |
| 8 | 6 | 16 | 48 | 1 | 48 | 16 | 16 | 12.5 | 6144 |

Figure 48b

【図48c】

| EXTCLK=19.200 MHz | | | | | | | | | |
|-------------------|-----------------|-----------------|-----|--------------|-----------------|------------|-----------|---------|-----------------|
| 総ビットストリームの数 | オーディオビットストリームの数 | 出力サンプル周波数 [kHz] | OSR | オーディオチャンネルの数 | 出力サンプル周波数 [kHz] | WORD [ビット] | 制御動作 | | バスクロック周波数 [kHz] |
| | | | | | | | 周波数 [kHz] | 帯域幅 [%] | |
| 8 | 3 | 24 | 50 | 5 | 48 | 20 | 25 | 12.5 | 9600 |
| 8 | 3 | 48 | 50 | 10 | 48 | 20 | 50 | 12.5 | 19200 |
| 8 | 5 | 48 | 50 | 5 | 48 | 20 | 50 | 12.5 | 19200 |

Figure 48c

【図49a】

| ビット | B _{N+1} | B _{N+2} | B _{N+3} | ... | B ₂ | B ₁ | B ₀ |
|-----|------------------|------------------|------------------|-----|-------------------|-------------------|-------------------|
| 値 | -1 | +2 ¹ | +2 ² | ... | +2 ^{N+3} | +2 ^{N+2} | +2 ^{N+1} |

Figure 49a

【図49b】

| 元のコンテンツ [kHz] | オーバーサンプリングされた周波数 [kHz] | オーバーサンプリング率 | 典型的なバスクロック周波数 [kHz] | 典型的なシステムクロック周波数 [kHz] |
|---------------|------------------------|-------------|---------------------|-----------------------|
| 48 | 3072 | 64 | 3072 or 6144 | 12288 |
| 48 | 2400 | 50 | 4800 | 19200 |
| 48 | 4800 | 100 | 4800 | 19200 |
| 48 | 3200 | 66.67 | 3.200 | 19200 |
| 44.1 | 2822.4 | 64 | 11289.6 | PLL needed |
| 48 | 7056 | 147 | 7056 | PLL needed |
| 44.1 | 7056 | 160 | 7056 | PLL needed |

Figure 49b

【図49c】

| プリアンブル | バイト1 | バイト2 | バイト3 | バイト4 | バイト5 | バイト6 | バイト7 | バイト8 | バイト9 |
|----------|--|------|------|------|------|------|----------|------|--------|
| 0 (NULL) | 0=空 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 (CR) | 文字列(不明な長さ)、各文字≠0 | | | | | | 0=文字列の終了 | 0 | 0 |
| 32 (SP) | 無制限(等時性)の文字列 | | | | | | | | |
| 35 (#) | 文字列またはファイル長:最上位バイト、MSB-1、...、LSB。max2 ⁶⁴ -1バイト長さ文字列 | | | | | | | | 第1のバイト |
| 64 (@) | 将来拡張のために保留されている | | | | | | | | |

Figure 49c

【図50b】

| ステップ | ACC=ACC-X | MSB(ACC)からの繰り上げ | ACC=ACC-Y | SYNC |
|------|-------------|-----------------|-------------|------|
| 初期化 | 96 | 0 | 96-0=96 | 0 |
| 1 | 96+147=243 | 0 | 243-0=243 | 0 |
| 2 | 243+147=390 | 1 | 390-160=230 | 1 |
| 3 | 230+147=377 | 1 | 377-160=217 | 1 |
| 4 | 217+147=364 | 1 | 364-160=204 | 1 |
| 5 | 204+147=351 | 1 | 351-160=191 | 1 |
| 6 | 191+147=338 | 1 | 338-160=178 | 1 |
| 7 | 178+147=325 | 1 | 325-160=165 | 1 |
| 8 | 165+147=312 | 1 | 312-160=152 | 1 |
| 9 | 152+147=299 | 1 | 299-160=139 | 1 |
| 10 | 139+147=286 | 1 | 286-160=126 | 1 |
| 11 | 126+147=273 | 1 | 273-160=113 | 1 |
| 12 | 113+147=260 | 1 | 260-160=100 | 1 |
| 13 | 100+147=247 | 0 | 247-160=87 | 0 |
| 14 | 247+147=394 | 1 | 394-254=140 | 1 |

Figure 50b

【図50c】

| Fサンプルバインド [kHz] | Fサンプルオーディオ [kHz] | X | Y | コメント |
|-----------------|------------------|-----|------|--------|
| 48 | 44.1 | 147 | 160 | |
| 3.25=13/4 MHz | 8 | 4 | 1625 | |
| 48 | 8 | 1 | 6 | |
| 96 | 8 | 1 | 12 | |
| 16 kHz | 48 kHz | 3 | 1 | 分数ではない |

Figure 50c

【図50a】

```

Initialize accumulator ACC = 0
Loop: ACC=ACC+reg_X
if (overflow carry from MSB(ACC)) then ACC=ACC-reg_Y and D50=1
Else D50=0
Continue loop

```

Figure 50a

【図51a】

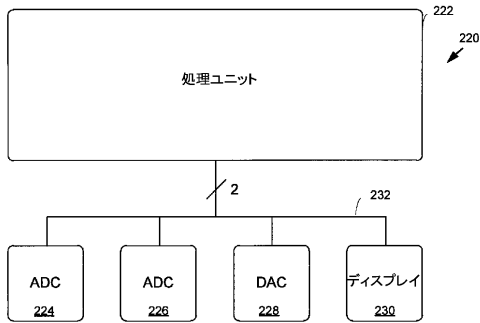


Figure 51a

【図51b】

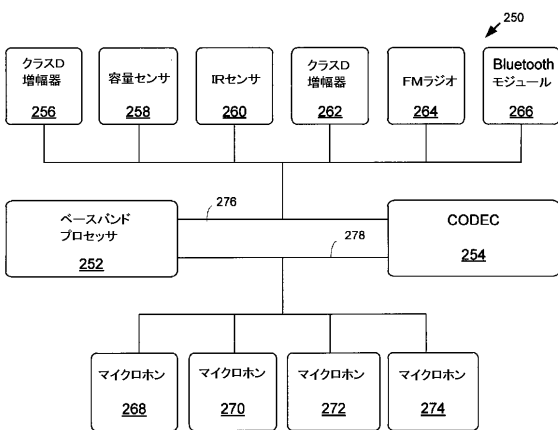


Figure 51b

【図52a】

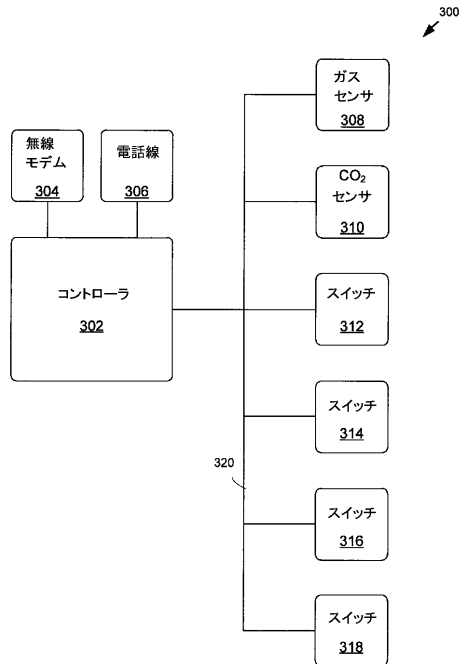


Figure 52a

【図52b】

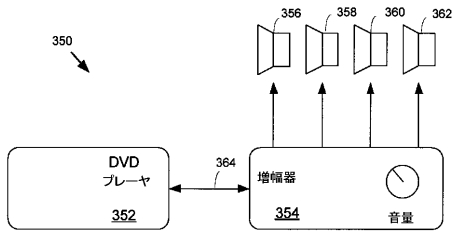


Figure 52b

【図52c】

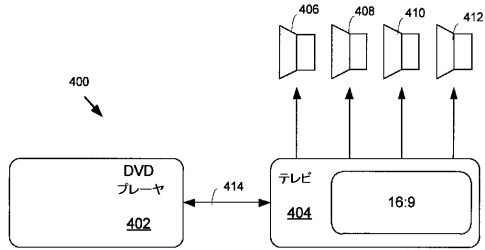


Figure 52c

【図52d】

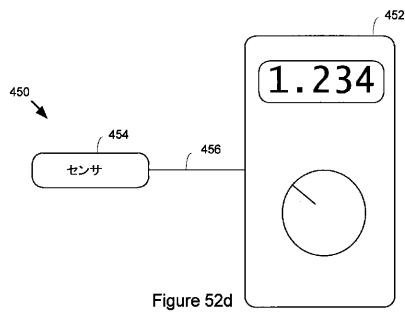


Figure 52d

【図52e】

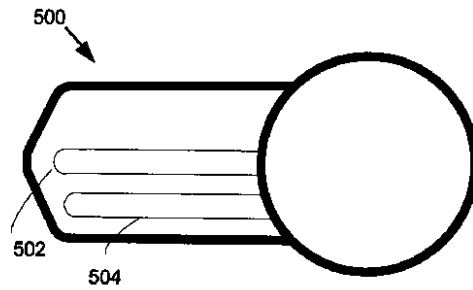


Figure 52e

【図52f】

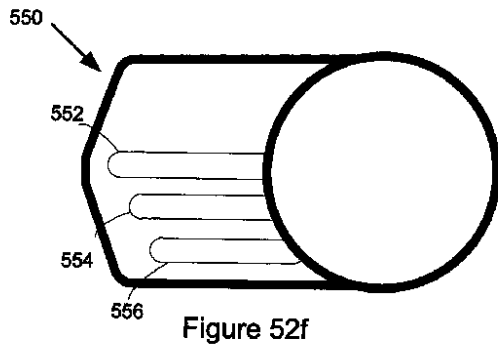


Figure 52f

【図52h】

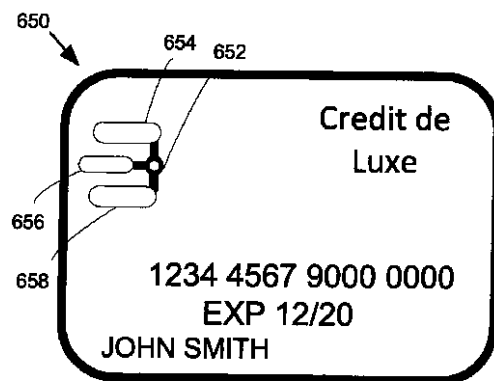


Figure 52h

【図52g】

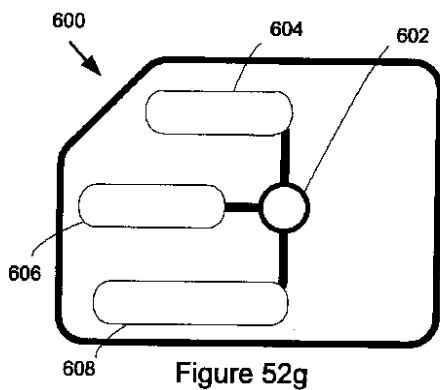


Figure 52g

【図52i】

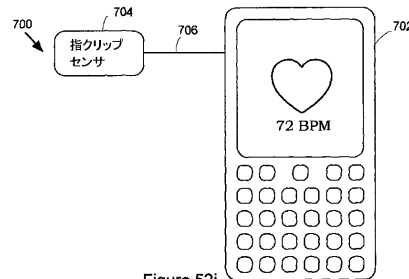


Figure 52i

【図53】

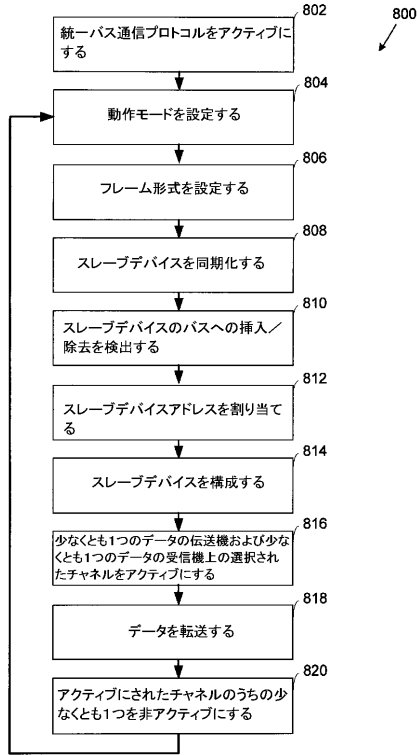


Figure 53

【図54】

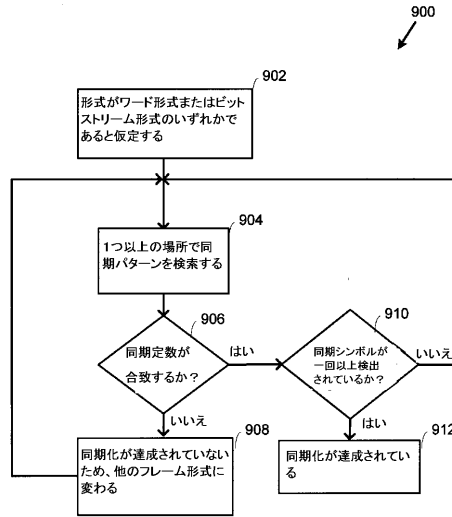


Figure 54

【図55a】

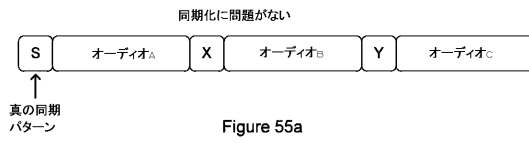


Figure 55a

【図55b】

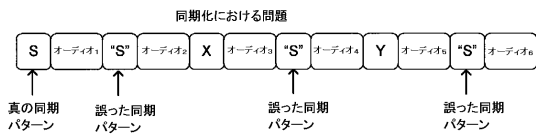


Figure 55b

【図56】

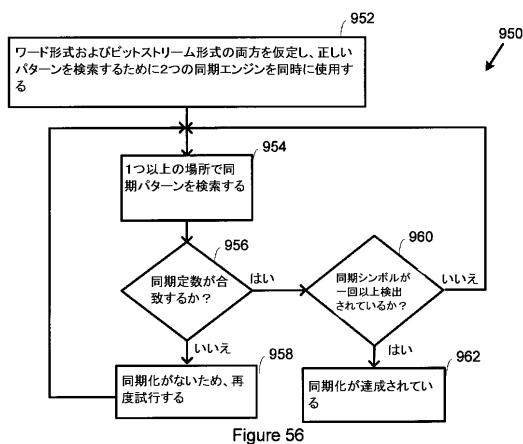


Figure 56

【図57】

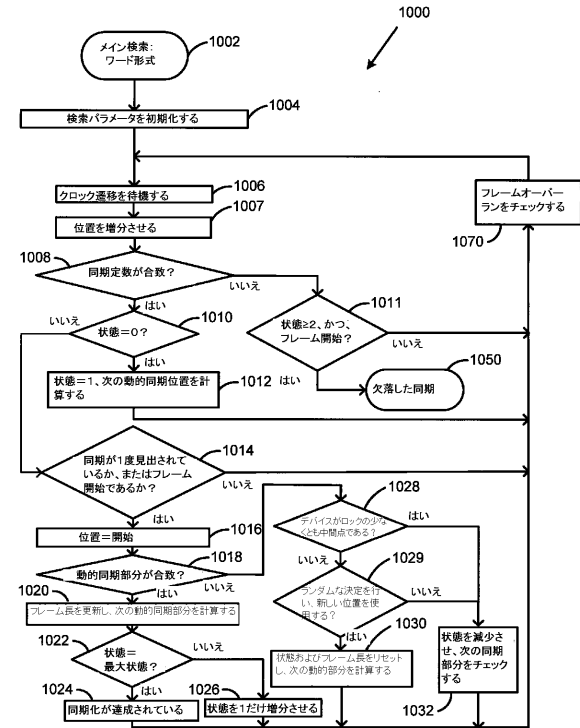


Figure 57

【図58a】

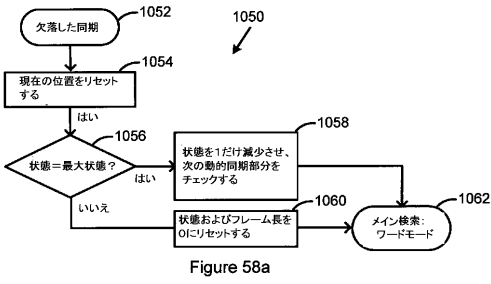


Figure 58a

【図58b】

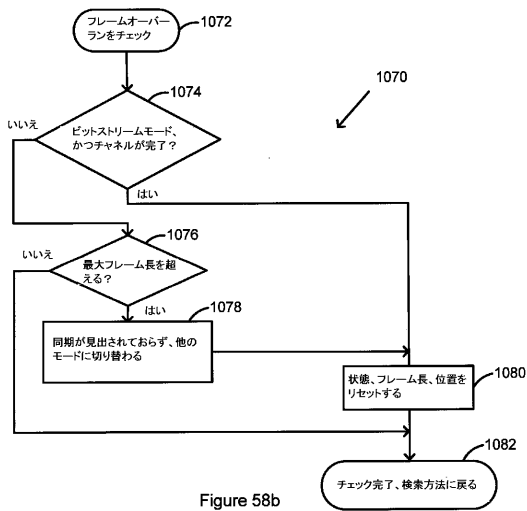


Figure 58b

【図59】

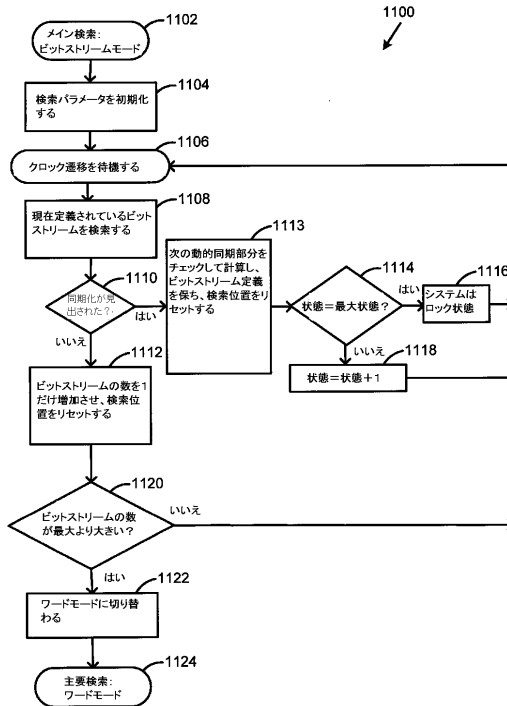


Figure 59

【図60】

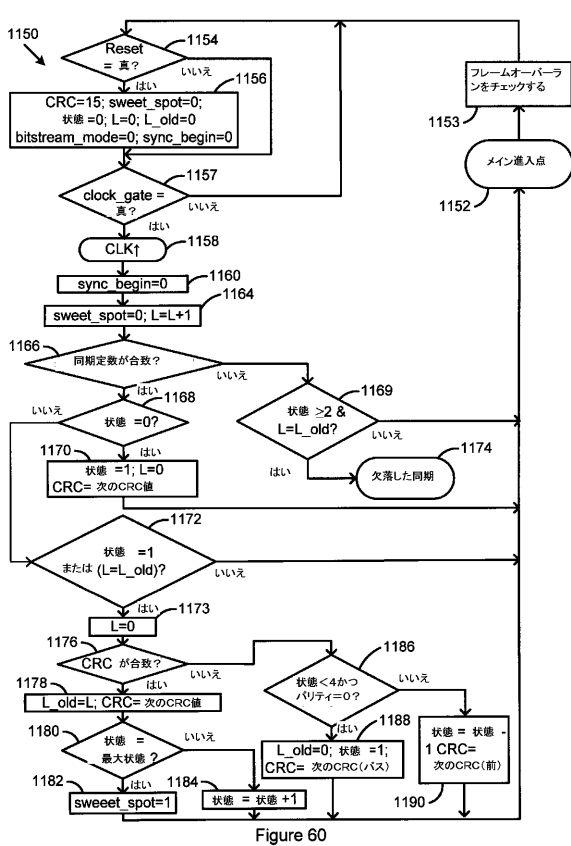


Figure 60

【図61a】

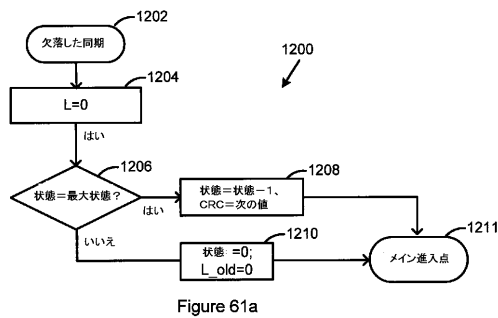


Figure 61a

【図61b】

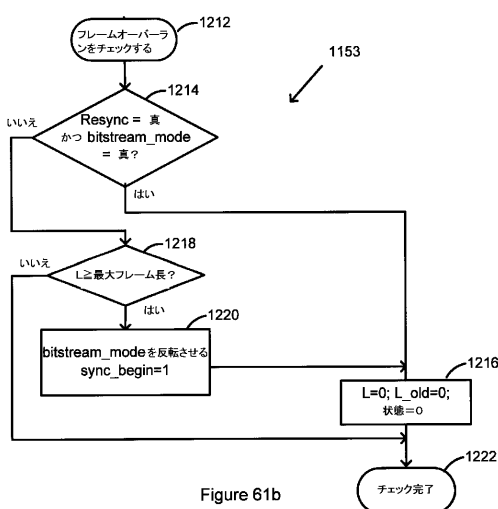


Figure 61b

【図62】

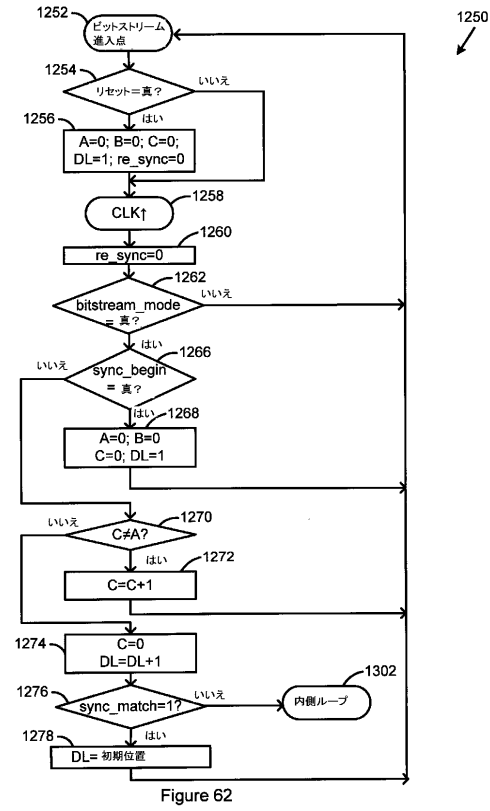


Figure 62

【図63】

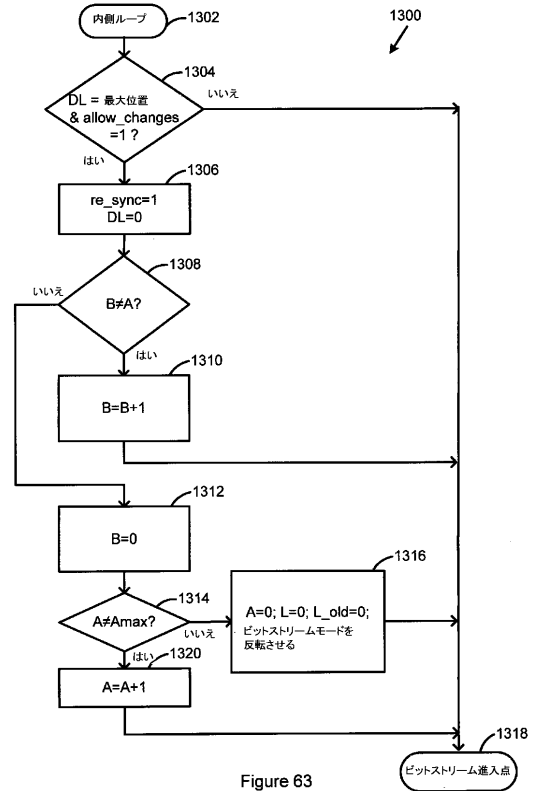


Figure 63

【図64】

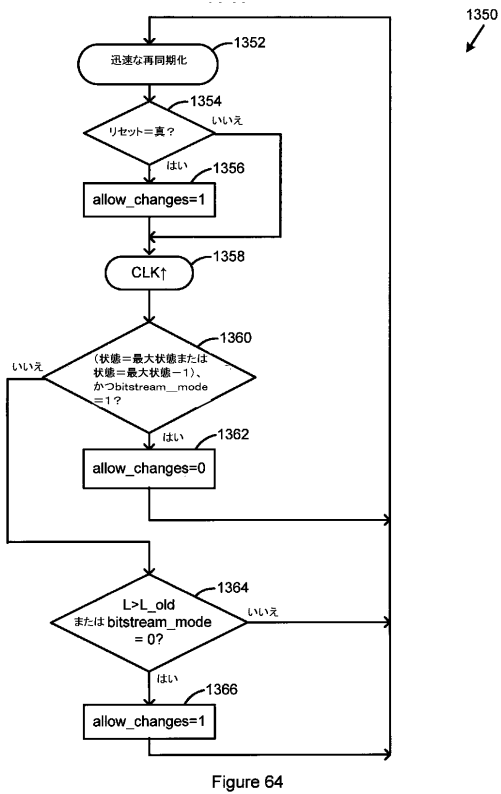


Figure 64

【図65】

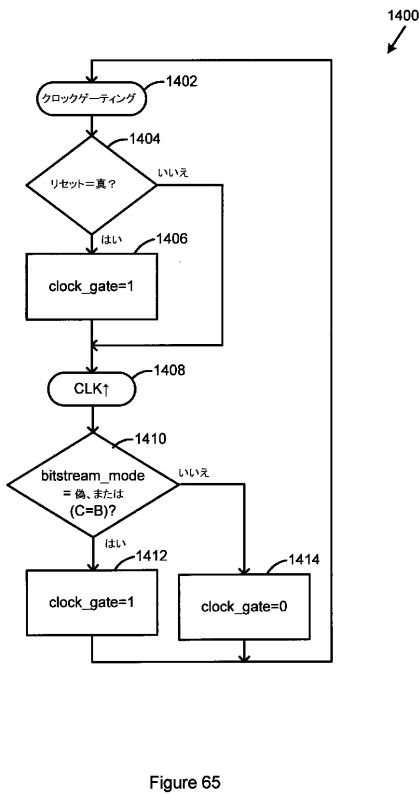


Figure 65

【図66】

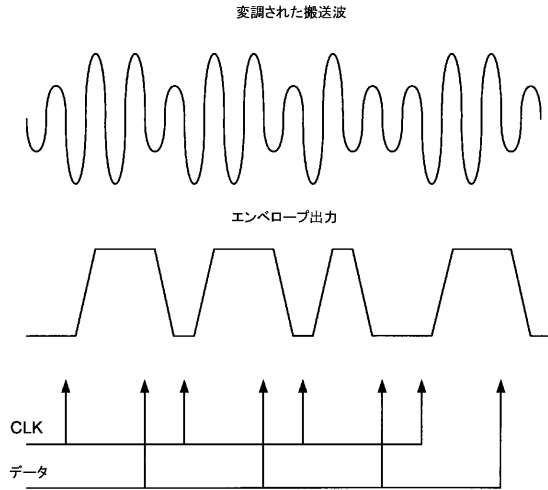


Figure 66

【図67】

| Sコマンドワード | | | | | | | | | | | | | | | | |
|----------|------------|---------|--------------------------|-----|-----|-----|----|----|--------------------|----|----|----|----|----|---------|----|
| ビット | S15 | S14 | S13 | S12 | S11 | S10 | S9 | S8 | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |
| フィールド | IRQS (r/w) | PAR (w) | CONSTANT SYNC SYMBOL (w) | | | | | | DYNAMIC SYMBOL (w) | | | | | | ACK (r) | |

Figure 67

【図68】

| ビット | 0-15 | 16-31 | 32-47 | 48-63 | 64-79 | 80-95 | 96-111 |
|-------|------|--------|-------|--------|-------|--------|--------|
| コンテンツ | S | オーディオ1 | X | オーディオ1 | Y | オーディオ1 | S |

パリティ: マスタが書き込む S14
ACK/NACK: スレーブが応答する S0

Figure 68

【図69a】

| レジスタ | 目的 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|--------|--------|-------------------|---------|------------|-----|----------------|----|----|----|
| 0x0000 | データポート | DATA (7:0) | | | | | | | |
| 0x0001 | | DATA (15:8) | | | | | | | |
| 0x0002 | 設定ポート | ACTIVATE | PREPARE | IRQ23 MASK | DIR | PORT | | | |
| 0x0003 | | CHANNEL SELECTION | | | | | | | |
| 0x0004 | | START | | | | | | | |
| 0x0005 | | SUBGROUP | | | | REPEAT | | | |
| 0x0006 | | SYNCHRONIZE | | | | CHANNEL LENGTH | | | |
| 0x0007 | | 行未拡張 | | | | SKIP | | | |
| 0x0008 | | FRAME STRUCTURE | | | | | | | |
| 0x0009 | | PCLKD | | | | TDM bit column | | | |

Figure 69a

【図69b】

| B5 | B4 | B3 | サブグループ当たり最大チャネル |
|----|----|----|-----------------|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 3 |
| 0 | 1 | 1 | 4 |
| 1 | 0 | 0 | 5 |
| 1 | 0 | 1 | 6 |
| 1 | 1 | 0 | 7 |
| 1 | 1 | 1 | 8 (全て) |

Figure 69b

【図69c】

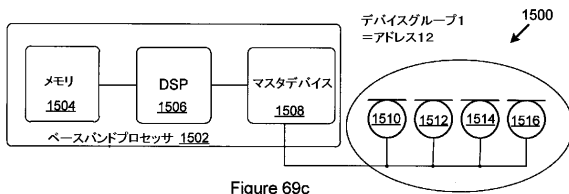


Figure 69c

【図70】

| X3 | X2 | X1 | X0 | 関数 | 使用される関数データ | 要求される | フローキーサポート |
|----|----|----|----|------------------|------------|-------|-----------|
| 0 | 0 | 0 | 0 | ハードリセット | いいえ | はい | はい |
| 0 | 0 | 0 | 1 | ソフトリセット | いいえ | はい | はい |
| 0 | 0 | 1 | 0 | デバイスIDを要求する | いいえ | はい | はい |
| 0 | 0 | 1 | 1 | グループアドレスを選択する | はい | はい | はい |
| 0 | 1 | 0 | 0 | BANK選択を反転させる | はい | はい | はい |
| 0 | 1 | 0 | 1 | 能力情報を要求する | いいえ | いいえ | いいえ |
| 0 | 1 | 1 | 0 | IRQソースを読み取る | いいえ | いいえ | いいえ |
| 0 | 1 | 1 | 1 | デバイス自己試験を行う | データ | いいえ | はい |
| 1 | 0 | 0 | 0 | 電力レベルを設定する | データ | いいえ | はい |
| 1 | 0 | 0 | 1 | 変換利得を設定する | データ | いいえ | はい |
| 1 | 0 | 1 | 0 | 遅延を設定する | データ | いいえ | はい |
| 1 | 0 | 1 | 1 | 校正データを要求する | いいえ | いいえ | いいえ |
| 1 | 1 | 0 | 0 | アドレスを読み取って増分する | データ | いいえ | いいえ |
| 1 | 1 | 0 | 1 | アドレスを書き込んで増分する | データ | いいえ | いいえ |
| 1 | 1 | 1 | 0 | アドレスで関数を実行する | アドレス | いいえ | はい |
| 1 | 1 | 1 | 1 | 間接的にアドレスで関数を実行する | アドレス | いいえ | はい |

Figure 70

【図69d】

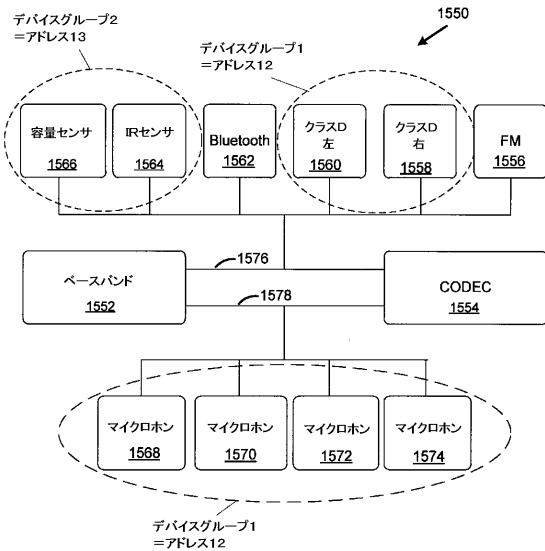


Figure 69d

【図71】

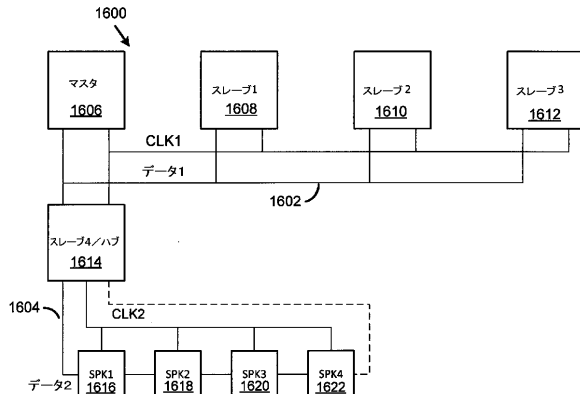


Figure 71

【図72】

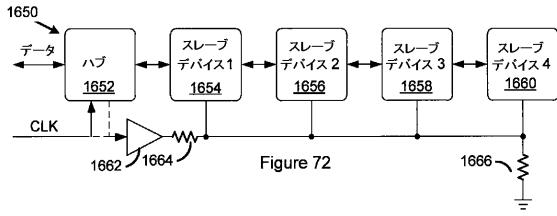


Figure 72

【図73】

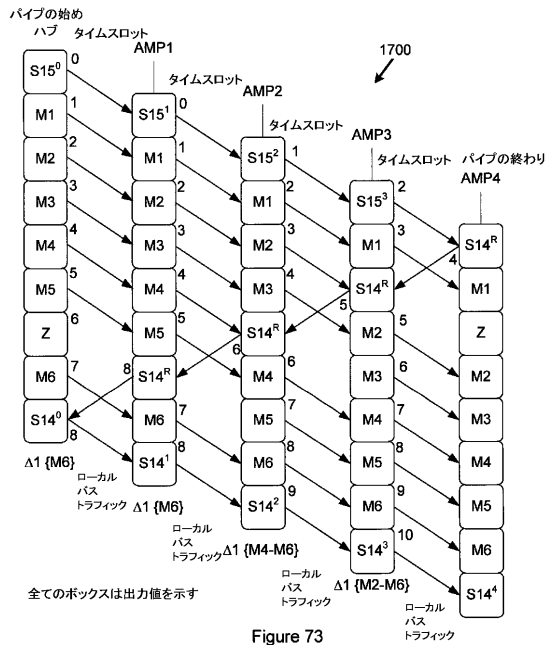


Figure 73

【図76】

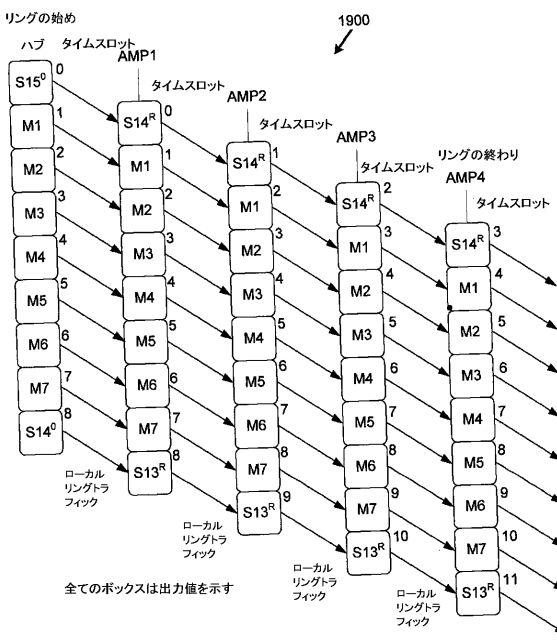


Figure 76

【図74】

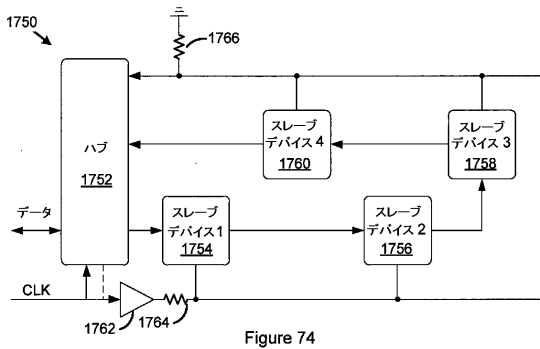


Figure 74

【図75】

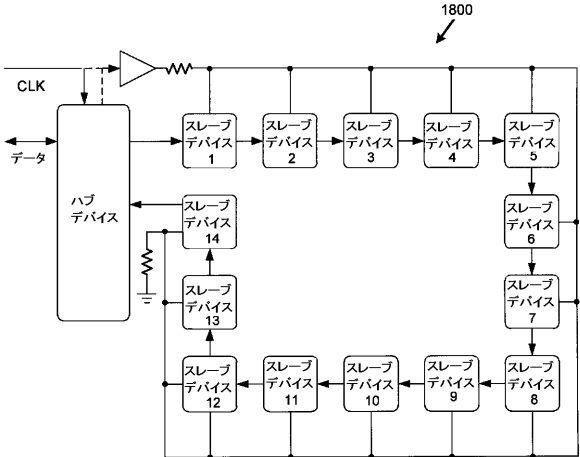


Figure 75

【図77a】

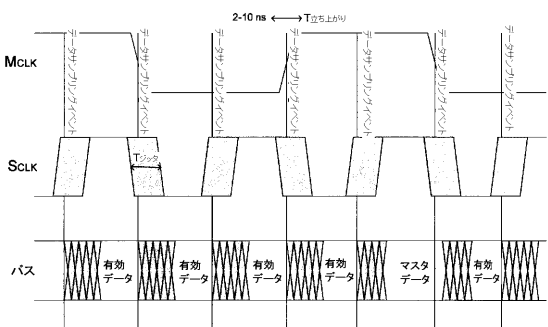


Figure 77a

【図77b】

| パラメータ(オーバーロックデバイスのみ) | 最小 | 通常 | 最大 | 単位 |
|--|------|-----|-------|-----|
| バス周波数、オーバーロック=1x | 0 | 9.6 | 19.20 | MHz |
| バス周波数、オーバーロック=2x | 2.40 | | 7.14 | MHz |
| バス周波数、オーバーロック=4x | 1.20 | | 3.57 | MHz |
| バス周波数、オーバーロック=6x | 0.80 | | 2.45 | MHz |
| VCO範囲*, オーバーロック=2x, 4x, あよび6x | 4.80 | | 15.5 | MHz |
| クロック半周期T _{clock} , オーバーロック=1x | 26 | | — | ns |
| クロック半周期T _{clock} , オーバーロック=2x | 70 | | 225 | ns |
| クロック半周期T _{clock} , オーバーロック=4x | 140 | | 450 | ns |
| クロック半周期T _{clock} , オーバーロック=6x | 204 | | 900 | ns |
| 出力を無効にする時間, T _{off} | -2 | | 5 | ns |
| 出力を有効にする時間, T _{on} | 9 | | | ns |
| データが安定化される時間, T _{setup} | | | 16 | ns |
| 内部データ遅延, スleepデバイス, T _{del} | 2 | 3 | 4 | ns |
| スレepデバイスPLL不確実性, T _{skitter} | -4 | 0 | +4 | ns |
| PLLがrmsジッタを含む | -250 | | 250 | ps |
| マスタサンプリング不確実性, T _{skitter} | -1 | 0 | 1 | ns |
| T _{lock} | | | 10 | ms |

Figure 77b

【図77c】

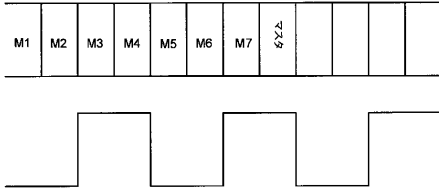


Figure 77c

【図77d】

| B0 | B1 | オーバーラッピング係数 | 電力[%] |
|----|----|-------------|-------|
| 0 | 0 | 1 | 100 |
| 0 | 1 | 2 | 66.7 |
| 1 | 0 | 4 | 50.0 |
| 1 | 1 | 6 | 44.4 |

Figure 77d

【図78a】

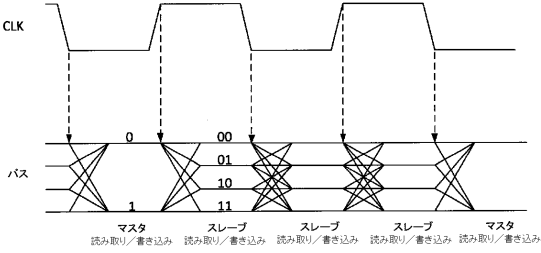


Figure 78a

【図78b】

| ビット番号 | ビットストリームサンプル | バイナリサンプル |
|---------------------------|-------------------------------------|-------------------------------------|
| スロット中の第1のビット(N) | 第1のビット(B _N) | 第1のビット(B _N) |
| スロット中の最後のビット(N+1)までの後続ビット | B _{N+1} XOR B _N | B _{N+1} XOR B _N |

Figure 78b

【図81a】

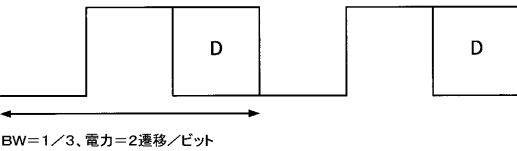


Figure 81a

【図81b】

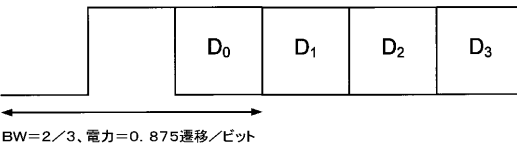


Figure 81b

【図81c】

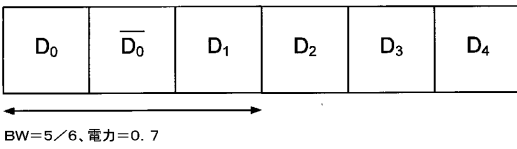


Figure 81c

【図79】

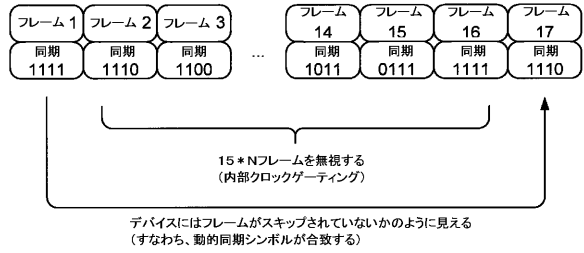


Figure 79

【図80a】

| B3 | B2 | B1 | B0 | おおよその電力[%] | 機能性 | 動作モードの変更 |
|----|----|----|----|------------|---------------------|--|
| 0 | 0 | 0 | 0 | <0.1 | 次のフレーム境界でシャットダウンする | デバイスが自律のみをアクティブ化可能 |
| 0 | 0 | 0 | 1 | <0.1 | 部分シャットダウン(1/256を監視) | マスタがアクティブ化される(送PIN3) |
| 0 | 0 | 1 | 0 | <0.1 | 部分シャットダウン(1/16を監視) | |
| 0 | 0 | 1 | 1 | <0.1 | UARTモード、帯域幅制御 | デバイスがバスロックをアクティブ化する |
| 1 | 0 | 0 | 0 | 2 | 制御のみ、限定された機能性 | FUNCTION SET POWERを使用 してマスタがアクティブ化される |
| 1 | 0 | 0 | 1 | 25 | 機能性を伴う最低電力 | |
| 1 | 0 | 1 | 0 | 50 | 低電力モードII | |
| 1 | 0 | 1 | 1 | 75 | 低電力モードI | |
| 1 | 1 | 0 | 0 | 100 | 通常モード | |
| 1 | 1 | 0 | 1 | 150 | 改良型性能モード | |
| 1 | 1 | 1 | 0 | 200 | 高性能モード | |
| 1 | 1 | 1 | 1 | 400 | 最高性能モード | |

Figure 80a

【図80b】

| パラメータ | 最小 | 典型 | 最大 | 単位 |
|-------------------------|-------|-------|------|-----|
| 設定時間 | 1 | 3 | 10 | μs |
| 周波数範囲 | 2 | 4 | 10 | MHz |
| マスタがDATAを有効にした後、クロックを待つ | 0.033 | 0.040 | 2000 | ms |

Figure 80b

フロントページの続き

(31)優先権主張番号 13/784,457

(32)優先日 平成25年3月4日(2013.3.4)

(33)優先権主張国 米国(US)

(72)発明者 ボールセン, ジェンス クリスチャン

カナダ国 エヌ2ケー 0エー7 オンタリオ, ウォータールー, ユニバーシティ アベニュー
ー イースト 2200, エクステンション 70873

合議体

審判長 大塚 良平

審判官 宮下 誠

審判官 中野 浩昌

(56)参考文献 特開2001-345875(JP,A)

国際公開第00/044125(WO,A1)

特開平08-242223(JP,A)

特開昭63-292841(JP,A)

(58)調査した分野(Int.Cl., DB名)

H04L7/00