(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7148413号
(P7148413)

(45)発行日　令和4年10月5日(2022.10.5)

(24)登録日　令和4年9月27日(2022.9.27)

(51)国際特許分類

		FI		
G06F 13/38 (2006.01)		G06F	13/38	340Z
G06F 13/42 (2006.01)		G06F	13/38	350
H04L 47/43 (2022.01)		G06F	13/42	350Z
H04L 49/9057(2022.01)		H04L	47/43	
		H04L	49/9057	

請求項の数　13　（全27頁）

(21)出願番号　特願2018-565372(P2018-565372)
(86)(22)出願日　平成29年6月26日(2017.6.26)
(65)公表番号　特表2019-526844(P2019-526844A)
(43)公表日　令和1年9月19日(2019.9.19)
(86)国際出願番号　PCT/US2017/039198
(87)国際公開番号　WO2018/005322
(87)国際公開日　平成30年1月4日(2018.1.4)
審査請求日　令和2年6月1日(2020.6.1)
(31)優先権主張番号　62/355,166
(32)優先日　平成28年6月27日(2016.6.27)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　62/517,247
(32)優先日　平成29年6月9日(2017.6.9)
最終頁に続く

(73)特許権者　507364838
クアルコム，インコーポレイテッド
アメリカ合衆国 カリフォルニア 92121 サン ディエゴ モアハウス ドライブ 5775
(74)代理人　100108453
弁理士　村山 靖彦
(74)代理人　100163522
弁理士　黒田 晋平
(72)発明者　アンディー・ユウ
アメリカ合衆国・カリフォルニア・92121・サン・ディエゴ・モアハウス・ドライヴ・5775
(72)発明者　アンドリュー・チュン
アメリカ合衆国・カリフォルニア・92
最終頁に続く

(54)【発明の名称】　アイソクロナスデータストリームを制御するためのシステムおよび方法

(57)【特許請求の範囲】

【請求項１】

ユニバーサルシリアルバス(USB)システムにおけるオーディオおよび/またはビデオの通信を制御するための方法であって、

第1のプロセッサ内のUSBドライバにおいて可変サイズのパケットを受信するステップと、

受信された前記可変サイズのパケットに基づいて、前記第1のプロセッサにおいて均一サイズのパケットを組み立てるステップであって、バス周波数、サンプリング周波数、およびパケットあたりのサンプルの数を使用して前記パケットの前記サイズを計算するステップを備える、ステップと、

前記均一サイズのパケットを第2のプロセッサに渡すステップであって、前記第2のプロセッサはアプリケーションプロセッサであり、前記均一サイズのパケットは、前記第2のプロセッサのプロトコルスタック中のアプリケーションレイヤにおいてアプリケーションによって使用可能である、ステップと

を含む、方法。

【請求項２】

前記第1のプロセッサおよび前記第2のプロセッサは単一の集積回路の中に集積される、請求項1に記載の方法。

【請求項３】

前記第1のプロセッサにおいて前記可変サイズのパケットを受信するステップは、前記

可変サイズのパケットをマイクロプロセッサまたはオーディオデジタル信号プロセッサ(ADSP)において受信するステップを含む、請求項1に記載の方法。
【請求項４】
前記第1のプロセッサにおいて前記可変サイズのパケットを受信するステップは、前記可変サイズのパケットを周辺機器とホストとの間の中間デバイスにおいて受信するステップを含み、前記中間デバイスが前記第1のプロセッサを含む、請求項1に記載の方法。
【請求項５】
前記可変サイズのパケットを受信するステップは、前記可変サイズのパケットを周辺機器中の前記第1のプロセッサにおいて受信するステップを含む、請求項1に記載の方法。
【請求項６】
ユニバーサルシリアルバスシステムにおけるオーディオおよび/またはビデオの通信を制御するためのホストのための装置であって、
第1のプロセッサ内のUSBドライバにおいて可変サイズのパケットを受信するための手段と、
受信された前記可変サイズのパケットに基づいて、前記第1のプロセッサにおいて均一サイズのパケットを組み立てるための手段であって、バス周波数、サンプリング周波数、およびパケットあたりのサンプルの数を使用して前記パケットの前記サイズを計算する、手段と、
前記均一サイズのパケットを第2のプロセッサに渡すための手段であって、前記第2のプロセッサはアプリケーションプロセッサであり、前記均一サイズのパケットは、前記第2のプロセッサのプロトコルスタック中のアプリケーションレイヤにおいてアプリケーションによって使用可能である、ための手段と
を備える、装置。
【請求項７】
前記第2のプロセッサはアプリケーションプロセッサであり、前記装置はユニバーサルシリアルバス(USB)ハードウェアをさらに備え、前記第1のプロセッサがオーディオデジタル信号プロセッサ(ADSP)であり、前記ADSPが、前記USBハードウェアを通して前記ADSPにおいて可変サイズのパケットを受信するように構成される、請求項6に記載の装置。
【請求項８】
前記第2のプロセッサはアプリケーションプロセッサであり、前記第1のプロセッサおよび前記第2のプロセッサが、単一のシステムオンチップ(SoC)に含まれる、請求項6に記載の装置。
【請求項９】
前記第1のプロセッサがマイクロプロセッサであるか、または前記ADSPに含まれる、請求項7に記載の装置。
【請求項１０】
前記第1のプロセッサがマイクロプロセッサまたはオーディオデジタル信号プロセッサ(ADSP)である、請求項8に記載の装置。
【請求項１１】
請求項7に記載の前記装置を備えるプロセッサであって、
オーディオデータバッファを備え、前記第1のプロセッサが、ユニバーサルシリアルバス(USB)オーディオクライアント(UAC)である、プロセッサ。
【請求項１２】
前記プロセッサは、USB周辺機器内に位置付けられるように適合される、請求項11に記載のプロセッサ。
【請求項１３】
前記プロセッサは、ホスト中に位置付けられるか、またはセットトップボックス、エンターテインメントユニット、ナビゲーションデバイス、通信デバイス、固定ロケーションデータユニット、モバイルロケーションデータユニット、全地球測位システム(GPS)デバイス、モバイルフォン、セルラーフォン、スマートフォン、セッション開始プロトコル(SI

P)フォン、タブレット、ファブレット、サーバ、コンピュータ、ポータブルコンピュータ、モバイルコンピューティングデバイス、ウェアラブルコンピューティングデバイス、デスクトップコンピュータ、携帯情報端末(PDA)、モニタ、コンピュータモニタ、テレビ、チューナ、ラジオ、衛星ラジオ、音楽プレーヤ、デジタル音楽プレーヤ、ポータブル音楽プレーヤ、デジタルビデオプレーヤ、ビデオプレーヤ、デジタルビデオディスク(DVD)プレーヤ、ポータブルデジタルビデオプレーヤ、自動車、車両構成要素、アビオニクスシステム、ドローン、およびマルチコプターからなるグループから選択されたデバイスに組み込まれるように適合される、請求項11に記載のプロセッサ。

【発明の詳細な説明】

【技術分野】

【０００１】

優先権の主張

本出願は、その内容全体が参照により本明細書に組み込まれる、2016年6月27日に出願された"PROGRAMMABLE RATE-MATCHED DATA RATE OUTPUT REGULATOR FOR ISOCHRONOUS DATA STREAMS"という名称の米国仮特許出願第62/355,166号の優先権を主張する。

【０００２】

本出願はまた、その内容全体が参照により本明細書に組み込まれる、2017年6月9日に出願された"ISOCHRONOUS DATA STREAM CONTROL SYSTEMS AND METHODS"という名称の米国仮特許出願第62/517,247号の優先権を主張する。

【０００３】

本出願は、またその内容全体が参照により本明細書に組み込まれる、2017年6月23日に出願された"SYSTEMS AND METHODS FOR CONTROLLING ISOCHRONOUS DATA STREAMS"という名称の米国特許出願第15/631,807号の優先権を主張する。

【０００４】

本開示の技術は概して、データバス上の任意のデータストリームの扱いに関する。

【背景技術】

【０００５】

コンピューティングデバイスは、現代の生活においてどこでも見られるようになっている。コンピューティングデバイスの人気は、部分的にはコンピューティングデバイス上で利用可能な、高まり続ける機能性のおかげで急増した。機能性の高まりと同時に、コンピューティングデバイスに関連付けられる場合がある補助デバイスの数およびタイプが増加している。いくつかのケースでは、補助デバイスは、スマートフォンへのカメラの集積など、コンピューティングデバイスに集積される場合がある。他のケースでは、補助デバイスは、何らかの形の外部インターフェースを通してコンピューティングデバイスに結合されるオーディオヘッドセットなどの周辺装置であってもよい。両方のケースにおいて、コンピューティングデバイス上で稼動するアプリケーションが必要に応じて補助デバイスと対話できるようにする様々なプロトコルが発生している。

【０００６】

1つの普及しているプロトコルは、ユニバーサルシリアルバス(USB)プロトコルである。USBは、フルスピード(FS)、ハイスピード(HS)、およびスーパースピード(SS)を含む様々な種類で存在する。さらに、USBは、ホストと周辺デバイスとの間の様々なクロック同期方式を可能にする。特に、USBは、周辺デバイスからクロックを同期すること(非同期と称される)、ホストからクロックに同期すること(同期と称される)、およびホストと周辺デバイスとの間でクロック同期責任を共有すること(適応と称される)を企図している。様々な種類およびクロック同期方式が、USBプロトコルを使用するデバイスの数を増やすための設計柔軟性を可能にするが、無数の選択肢が、いくつかの設計決定をより困難にしている。

【０００７】

そのような設計決定は、オーディオおよび/またはビデオストリームがUSBインターフェースを通して伝送されているとき、さらに複雑になる。USBフォームファクタの普遍性に

より、USBホストは、周辺機器からのオーディオ/ビデオキャプチャと、周辺機器へのオーディオ/ビデオ再生との両方に順応できることが期待される。特に、USBホストは、異なる速度、異なるクロック同期方式、異なるサンプリングレート、および可変サイズのデータに順応できることが期待される。従来のシステムは、アプリケーションレイヤにおけるそのような順応に負担をかけ、これはアプリケーションレイヤ中のアプリケーション部分にかなりのバッファリングおよび複雑なアルゴリズムを要求する。さらに、サービス間隔を増大するための現行の提案があり、これらはアプリケーションレイヤを扱うアプリケーションプロセッサに追加負担を課す場合がある。したがって、現在実装されている両方の可変データストリームを扱う際に、より大きい柔軟性を可能にするUSB互換システムであって、異なる入力パラメータを扱うための柔軟性を有するUSB互換システムを提供する方法が必要である。

【発明の概要】

【課題を解決するための手段】

【０００８】

発明を実施するための形態で開示する態様は、アイソクロナスデータストリームを制御するためのシステムおよび方法を含む。本開示の特定の態様は、ほぼどのアイソクロナスデータストリームとも使用されるように設計されるが、ユニバーサルシリアルバス(USB)プロトコルとの使用に好適である。さらに、本開示の態様は、USBプロトコル内の既存の構成可能性に順応し、かつUSBプロトコルにおける提案される将来の変更に順応するように柔軟性がある。システムおよび方法の柔軟性は、(1)USBホストシステム時間とアプリケーションとの間のドリフトおよび(2)USBホストシステムとUSBデバイスクロックとの間のドリフトを算出することによって提供される。これらの2つのドリフト算出に基づいて、次の配信スケジュールをプログラムするようにタイムスタンプが合成されてもよい。このタイムスタンプを使用すると、ジッタ訂正が起こる可能性があり、アプリケーションプロセッサに渡すために、均一サイズのパケットが組み立てられる場合がある。そのような均一サイズのパケットの使用により、アプリケーションレイヤ中でバッファの必要がなくなる場合があり、それによりデータストリームがオーディオデータストリームであるときのユーザエクスペリエンスが向上する場合がある。

【０００９】

この点において、一態様では、USBシステムにおける通信を制御するための方法が開示される。本方法は、USBドライバを有する第1のプロセッサにおいて可変サイズのパケットを受信するステップを含む。本方法はまた、第1のプロセッサにおいて均一サイズのパケットを組み立てるステップを含む。本方法はまた、均一サイズのパケットをプロトコルスタック中のアプリケーションレイヤにおけるアプリケーションによる使用のために第2のプロセッサに渡すステップを含む。

【００１０】

別の態様では、ホストが開示される。ホストはまた、アプリケーションプロセッサを含む。ホストは、USBハードウェアを含む。ホストはまた、オーディオデジタル信号プロセッサ(ADSP)を含む。ADSPはまた、可変サイズのパケットをUSBハードウェアを通してADSPにおいて受信するように構成される。ADSPは、ADSPにおいて均一サイズのパケットを組み立てるように構成される。ADSPはまた、均一サイズのパケットをプロトコルスタック中のアプリケーションレイヤにおけるアプリケーションによる使用のためにアプリケーションプロセッサに渡すように構成される。

【００１１】

別の態様では、ホストが開示される。ホストはまた、アプリケーションレイヤを含む。ホストは、USBハードウェアを含む。ホストはまた、複数のプロセッサを含むシステムオンチップ(SoC)を含む。複数のプロセッサは、第1のプロセッサにおいて可変サイズのパケットを受信するように構成される。複数のプロセッサはまた、第1のプロセッサにおいて均一サイズのパケットを組み立てるように構成される。複数のプロセッサはまた、均一サイズのパケットをプロトコルスタック中のアプリケーションレイヤにおけるアプリケーシ

ョンによる使用のために第2のプロセッサに渡すように構成される。
【００１２】
別の態様では、USBシステムにおけるドリフトを検出するための方法が開示される。本方法は、オーディオ周辺機器とホストとの間のUSBバス上で分数サンプリングレートが使用されると判断するステップを含む。本方法はまた、サービス間隔に渡って分数サンプリングレートに関連付けられた第1の分数剰余を判断するステップを含む。第1の分数剰余に基づいて、本方法はまた、分数剰余をもたないことが要求される間隔の数に対応する整数を算出するステップを含む。本方法はまた、各整数個の間隔ごとにドリフトを確認するステップを含む。
【００１３】
別の態様では、プロセッサが開示される。プロセッサは入力を含む。プロセッサはまた、制御システムを含む。制御システムは、オーディオ周辺機器とホストとの間のUSBバス上で分数サンプリングレートが使用されると判断するように構成される。制御システムはまた、サービス間隔に渡って分数サンプリングレートに関連付けられた第1の分数剰余を判断するように構成される。第1の分数剰余に基づいて、制御システムはまた、分数剰余をもたないことが要求される間隔の数に対応する整数を算出するように構成される。制御システムはまた、各整数個の間隔ごとにドリフトを確認するように構成される。
【００１４】
別の態様では、タイムスタンプを合成するための方法が開示される。本方法は、データ配信ハンドラから稼働コマンドを受信するステップを含む。本方法はまた、高解像度タイマからの出力と、計算された絶対タイムスタンプを合計するステップを含む。
【００１５】
別の態様では、プロセッサが開示される。プロセッサはまた、オーディオデータバッファを含む。プロセッサは、USBオーディオクライアント(UAC)を含む。UACは、可変サイズのパケットを受信するように構成される。UACはまた、均一サイズのパケットを組み立てるように構成される。UACはまた、均一サイズのパケットをプロトコルスタック中のアプリケーションレイヤにおけるアプリケーションによる使用のために第2のプロセッサに渡すように構成される。
【図面の簡単な説明】
【００１６】
【図１】本開示の例示的な態様による、ユニバーサルシリアルバス(USB)ケーブルおよびコネクタを通してリモートオーディオ周辺機器が結合されるモバイル通信デバイスの簡略化斜視図である。
【図２】USB周辺機器からプロセッサ内のアプリケーションレイヤへの従来のオーディオフローのブロック図である。
【図３】本開示の例示的な態様による、USBシステム内でのオーディオフローのブロック図である。
【図４Ａ】本開示のデータレギュレータの代替配置をもつUSBシステムを示す図である。
【図４Ｂ】本開示のデータレギュレータの代替配置をもつUSBシステムを示す図である。
【図５】データレギュレータのブロック図である。
【図５（Ｃｏｎｔ．）】データレギュレータのブロック図である。
【図６】パケットサイズがどのように算出されるか、およびパケットがどのようにアプリケーションレイヤへ渡されるかを示す信号フロー図である。
【図６（Ｃｏｎｔ．）】パケットサイズがどのように算出されるか、およびパケットがどのようにアプリケーションレイヤへ渡されるかを示す信号フロー図である。
【図７】マイクロフォンからUSBホストへの帯域内ドリフト報告プロセスのブロック図である。
【図８】マイクロフォンからUSBホストへの帯域外ドリフト報告プロセスのブロック図である。
【図９】マイクロフォンからホストへの帯域内ドリフト報告プロセスと、ホストがプロセ

スをスピーカーへの再生のためにどのように使用するかとのブロック図である。
【図１０】マイクロフォンからホストへの帯域外ドリフト報告プロセスと、ホストがプロセスをスピーカーへの再生のためにどのように使用するかとのブロック図である。
【図１１】図3のUSBシステムを含むことができる例示的なプロセッサベースシステムのブロック図である。
【発明を実施するための形態】
【００１７】
次に、図面を参照して、本開示のいくつかの例示的な態様について説明する。「例示的」という語は、「例、事例、または例示として機能すること」を意味するために本明細書で使用される。本明細書で「例示的」と記載される任意の態様は、必ずしも他の態様よりも好ましいまたは有利であると解釈されるべきではない。
【００１８】
発明を実施するための形態で開示する態様は、アイソクロナスデータストリームを制御するためのシステムおよび方法を含む。本開示の特定の態様は、ほぼどのアイソクロナスデータストリームとも使用されるように設計されるが、ユニバーサルシリアルバス(USB)プロトコルとの使用に好適である。さらに、本開示の態様は、USBプロトコル内の既存の構成可能性に順応し、かつUSBプロトコルにおける提案される将来の変更に順応するように柔軟性がある。システムおよび方法の柔軟性は、(1)USBホストシステム時間とアプリケーションとの間のドリフトおよび(2)USBホストシステムとUSBデバイスクロックとの間のドリフトを算出することによって提供される。これらの2つのドリフト算出に基づいて、次の配信スケジュールをプログラムするようにタイムスタンプが合成されてもよい。このタイムスタンプを使用すると、ジッタ訂正が起こる可能性があり、アプリケーションプロセッサに渡すために、均一サイズのパケットが組み立てられる場合がある。そのような均一サイズのパケットの使用により、アプリケーションレイヤ中でバッファの必要がなくなる場合があり、それによりデータストリームがオーディオデータストリームであるときのユーザエクスペリエンスが向上する場合がある。
【００１９】
本開示の特定の態様を扱う前に、アイソクロナスデータストリームを制御するためのシステムおよび方法を実装する場合がある例示的なシステムの概要が開示される。上述したように、様々なアイソクロナスデータストリームに適用可能であるが、例示的な態様は、USBオーディオストリームに特に適用可能である。したがって、例示的なシステムは、USBデジタルオーディオシステムである。
【００２０】
この点において、図1は、USBケーブル106上のUSBタイプCコネクタ104に結合するように構成されるUSBタイプCレセプタクル102をもつモバイル通信デバイス100の簡略化斜視図である。USBケーブル106の遠位端には、ヘッドフォン112における複数のスピーカー110とマイクロフォン114とを有するデジタルオーディオヘッドセット108がある。デジタルオーディオ信号は、USBケーブル106を通して、モバイル通信デバイス100とデジタルオーディオヘッドセット108との間を渡る場合がある。マイクロフォン114からのオーディオは、発話パターンはめったに周期的でないので、時間領域において不均等に分散される場合がある。同様に、モバイル通信デバイス100は、どのデータ速度をデジタルオーディオヘッドセット108がサポートするかも先験的に知らず、モバイル通信デバイス100は、どの同期形式をデジタルオーディオヘッドセット108が使用するかも先験的に知らない。
【００２１】
本開示の例示的な態様は、図1のデジタルオーディオヘッドセット108などのオーディオ環境に好適であるが、本開示は、そのように限定されるわけではなく、モバイル通信デバイス100などのコンピューティングデバイスと、ディスプレイ、スピーカー、およびマイクロフォンを有する仮想現実ヘッドセット(またはスピーカーおよびマイクロフォンを有するディスプレイ)との間を渡るオーディオ/ビデオ信号と一緒に使用されてもよい。同様

に、USBタイプCケーブルが上で開示されるが、本開示は、USBの他のバージョンとともに容易に使用可能である。実際、USB速度(たとえば、フルスピード(FS)、スーパースピード(SS)、ハイスピード(HS))のうちのいずれも扱うことができることは、本開示の利点のうちの1つである。

【００２２】

図2は、本開示の態様を実装しないモバイル通信デバイス200においてオーディオ(およびおそらくビデオ)データがどのように扱われるかの簡略化ブロック図を提供する。モバイル通信デバイス200は、デジタルオーディオヘッドセットなどのUSB周辺機器202に結合されてもよい。USB周辺機器202は、非同期、同期、適応、または混合クロック同期モードをサポートしてもよく、1つまたは複数の位相ロックループ(PLLであって、2つが図示される)または遅延ロックループ(DLLであって、図示されていない)を含んでもよい。USB周辺機器202は、マイクロフォンを通すなどして、データ(Data INと称される)を(キャプチャと称されることがある)、ならびにヘッドフォンの中のスピーカーを通すなどして、出力データ(Data OUTと称される)を(再生と称されることがある)受信する場合がある。データは、USBケーブル206を通すなどして、モバイル通信デバイス200との間で、および適切なレセプタクル(図2には示さず)を通して、モバイル通信デバイス200内のUSBハードウェアコントローラ208に渡される。USBハードウェア(図中ではHWと称されることがある)コントローラ208は、システムオンチップ(SoC)210に通信可能に接続される。SoC210は、オーディオデジタル信号プロセッサ(ADSP)212およびアプリケーションプロセッサ(図中ではAPと称される)214を含んでもよい。ADSP212は、USBオーディオクライアント(UAC)ドライバ216を含んでもよい。USB周辺機器202からのデータは、USBハードウェアコントローラ208において受信され、SoC210に渡される。USB周辺機器202からのデータはジッタが多く、可変データフレームサイズ(USBハードウェアコントローラ208とUACドライバ216との間の様々なサイズのボックスによって、記号により図示される)を含むことに留意されたい。USB周辺機器202の1つまたは複数のPLLが高速または低速で稼働する場合、さらなる変動性が起こる場合がある。USBプロトコルには、フレーム内に一定数のサンプルがなければならないという要件がないので、またさらなる変動性が起こる場合がある。そのような変動性は、USBプロトコルの柔軟性および魅力に寄与するものの一部であるが、そのような変動性は概して、オーディオ処理では扱いにくい。USBハードウェアコントローラ208が、その内部バッファ(図示せず)中にデータを有するとき、USBハードウェアコントローラ208は、UACドライバ216に対して割込みを生成する。USBハードウェアコントローラ208は、タイムスタンプ機能をもたない。UACドライバ216は、割込みを受信し、USBハードウェアコントローラ208のバッファを排出させ、アプリケーションプロセッサ214に一定量のデータを与えようと試みる。1ミリ秒(1000Hzの共通USBバス転送速度に対応する)に相対した分数である、44.1キロヘルツ(kHz)という共通サンプリングレートなどの分数オーディオサンプリングがあるとき、UACドライバ216は、10個のパケットのうちの9つの中の44個のサンプルと、45個のサンプルをもつ1つのパケットとをもつデータを送信することになる。アプリケーションプロセッサ214中のデータ処理回路機構218は、そのバッファ220を高解像度システムタイマ222とともに使用して、アプリケーションレイヤアルゴリズム224にデータが与えられる前に、変動性を取り除く。非同期サンプルレートコンバータ(ASRC)226が、一定の持続時間に渡って、ドリフトと、サンプルのジッタが多いクラスタとを訂正する、このプロセスを支援してもよい。この配置は、アプリケーションプロセッサ214に負担をかけ、アプリケーションレイヤアルゴリズム224のための追加プログラミングを要求する。ADSP212およびアプリケーションプロセッサ214は、別個のプロセッサであるものとして記載されるが、両方のデバイスは、単一の集積回路(IC)の中に集積されてもよいことに留意されたい。図示されないが、ハードウェア直接メモリアクセス(DMA)コントローラがデータ割込みを生成してもよく、高解像度システムタイマ222からのハードウェアラッチタイムスタンプがハードウェアレジスタ中に記憶される。このタイムスタンプは、USBパケットに容易には関連付けられず、したがってドリフト検出を支援するために容易には利用可能でない。

【００２３】
本開示の例示的な態様は、ジッタ訂正がそこから適用されてもよく、合成タイムスタンプがそこから算出されてもよいエラーなしドリフト検出を提供する。この合成タイムスタンプを使用して、バッファを排出させるのに使用される次の配信スケジュールが算出されてもよい。さらに、算出をアプリケーションプロセッサの外に再位置付けすることによって、均一なデータフレームサイズがアプリケーションプロセッサに与えられてもよく、このことが、オーディオ品質を向上し、可能性としては省電力機会を与える場合がある。本開示の利益のうちの1つは、ホストとデバイスとの間のどの形のクロック同期手法(非同期、同期、または適応)ならびに様々なデータ速度、異なるサンプリングレート、可変サイズのデータ、異なるUSB速度(HS、FS、SS)、および異なるサービス間隔にも順応するための本開示の柔軟性である。本開示は、厳密にはハードウェアにおいて実装されてもよいが、本開示の柔軟性は、ソフトウェアの使用により向上され、変数は、どの構成にも順応するようにより容易に調節される。本開示のシステムおよび本開示の態様を実装するのに使用される場合がある様々なシグナリングの詳細を探求する前に、柔軟性を作成するのに使用される式の概要を提示する。

【００２４】
以下のセクションは、数学集約的であり、興味ある読者を念頭に置いているが、本開示の例示的な態様を理解するのに不可欠でない場合がある。数学的計算により本開示の理解を混乱させないことを好む読者向けに、例示的な態様の考察は、図3を参照して以下で再度始まる。

【００２５】
ホストによって使用される基本的ドリフト補償レートに合致するオーディオバッファ配信モデルは、以下のように表される場合がある。

$$ticks_{next}=ticks_{reference}+ticks_{offset}+D_1+D_2...+D_M \quad (式1)$$

【００２６】
式1において、$ticks_{next}$("Tnext"とも称される)は、次の配信スケジュールをプログラムするのに効果的に使用される合成タイムスタンプである。$ticks_{reference}$("Tref"とも称される)は、第1の合成タイムスタンプのタイムスタンプである。$Ticks_{offset}$("Toffset"とも称される)は、バッファの配信に使用される$ticks_{reference}$からのデルタであり、再生およびキャプチャのためのバッファのピックアップのタイミングとしても働く。式1において、各D_iは、デバイスクロックとUSB時間参照との間の総ドリフトを表す。ほとんどの状況では、検討するべきただ3つのクロック、すなわちUSBホストクロック、オーディオアプリケーションクロック、およびUSBデバイスクロックがある。USBホストクロックは、他の2つのクロックの両方のためのシステム時間参照として働き、したがって式1は通常次のように単純化する。

$$ticks_{next}=ticks_{reference}+ticks_{offset}+D_{app\text{-}usb}-D_{device\text{-}usb} \quad (式2)$$

【００２７】
式2は、オーディオキャプチャ経路とオーディオ再生経路との両方に役立つ。$D_{app\text{-}usb}$は、オーディオアプリケーションクロックとUSBホストクロックとの間の時間差である。$D_{device\text{-}usb}$は、USBホストクロックに対してUSBデバイスクロックがどれだけ速く進んでいるかである。これらの値は、一緒になって、正味のシステムドリフト(すなわち、オーディオサンプルがより高速またはより低速で動いているか)を与える。オーディオキャプチャ経路について、$D_{device\text{-}usb}$が正のとき、デバイスは、USBホストがオーディオサンプルをクリアするよりも高速にオーディオサンプルを配信している。$D_{app\text{-}usb}$が正のとき、オーディオアプリケーションは、USBホストがオーディオサンプルを配信するよりも高速にオーディオサンプルを取り出している。オーディオ再生経路上で、$D_{app\text{-}usb}$が正のとき、オーディオアプリケーションは、USBホストがオーディオサンプルをクリアするよりも高速にオーディオサンプルを配信している。$D_{device\text{-}usb}$が正のとき、デバイスは、USBホストがオーディオサンプルを配信するよりも高速にオーディオサンプルを取り出している。この値は、オーディオを合成および/または補間するために、非同期サンプルレートコンバー

タ(ASRC)に渡され、ASRCは、どれだけ訂正すべきかを知ることができる。

【００２８】

キャプチャおよび再生経路についてのドリフト$D_{device-usb}$は、明示的または暗黙的に判断されてもよい。ドリフトは、データフローの方向(すなわち、デバイスからホスト(一般にキャプチャ)またはホストからデバイス(一般に、再生))に基づいて取得される。ドリフト情報のソースは、USBが何を広告するかと、高レベルオペレーティングシステム(HLOS)によってUSBエンドポイントペア向けにどのアイソクロナス同期モードが選択されるかとに依存する。実際、キャプチャおよび再生経路の間には、アイソクロナス同期モードの20通りの組合せがある。

【００２９】

ドリフト情報のソースは、以下のTable 1(表1)に要約される。$D_{device-usb}$は、Table 1(表1)ではD_{device}と短縮される。

【表１】

In/Out	Sync	適応	Async 暗黙的	Async 明示的	なし
Sync	In: 式 12 Out: D_{device}=0	In: 式 12 Out:D_{device}=0	N/A	In: 式 12 Out: 式 6	In: 式 12 Out: なし
Adaptive	In: 式 12 Out: D_{device}=0	In: 式 12 Out: D_{device} = 0	N/A	In: 式 12 Out: 式 6	In: 式 12 Out なし
Async	In: 式 12 Out: D_{device}=0	In: 式 12 Out:D_{device}=0	In: 式 12 Out: D_{device} = $D_{device, in}$	In: 式 12 Out: 式 6	In: 式 12 Out: なし
なし	In: なし Out: D_{device}=0	In: なし Out:D_{device}=0	In: 式 12 Out: D_{device} = $D_{devicee, in}$	In: なし Out: 式 6	N/A

Table 1 ドリフト情報のソース

【００３０】

Table 1(表1)は、オーディオアプリケーションクロックがUSBホストクロックと同相であると仮定している($D_{app-usb}$=0)。この仮定により、すべての同期および適応再生(Out)経路が$Out:D_{device}$=0を有するようになる。

【００３１】

本開示の例示的な態様は、サンプリング周波数、サンプリング間隔、サンプルサイズ、バス速度、クロック同期モードなどの本質的にどの変化についてもドリフトを検出するための技法を提供する。この柔軟性は、これらの可変入力に順応するとともに適切なドリフト検出を可能にする汎用式により達成される。

【００３２】

品質、環境、製造精度はすべて、ある非同期クロックが、システム中の別の非同期クロックと比較して、正確に作動できることに影響することを了解されたい。キャプチャ経路に沿って複数のクロックがあり、再生経路上に複数のクロックがあるシステムが存在する。経路についての正味のドリフトは、経路に沿った各サブシステムクロックの間の時間微分の合計である。本開示は、適切な周波数でドリフトを測定することによって、エラーなしドリフト検出が可能にされ、必要のない測定が回避されることを示し、これにより、省電力が可能になる場合がある。

【００３３】

USBシステムにおけるオーディオストリーミングは、そのようなオーディオストリーミングが、アイソクロナス転送モードを使用することを期待されるという点で、問題点を追加する。それは、誤り検査も試行もないリアルタイム専用帯域幅モードである。オーディオサンプルが、オーディオパケットの形でバンドルされ、オーディオパケットは、毎(マイ

クロ)フレームごとに送信されてもよい。各そのようなフレームは、物理レイヤによってHSそれともFS USB転送モードが選択されるかに依存して、125μsまたは1msのいずれかである。USBプロトコルは、そのようなフレームを省電力のために、および多大なネットワーク待ち時間を扱うためにバーストで送信することをサポートする。サービス間隔当たりのフレームの数は、$2^{binterval-1}$によって記述され、bintervalは現在、1と16との間の値である。この数を拡大するために、USBプロトコルについて、運営団体の間で議論が行われてきた。サービス間隔当たりのフレームの数は固定だが、バーストごとに送信されるオーディオサンプルの数は可変であることが可能である。

【００３４】

ドリフトを評価するのに適しているとみなされる要因は、ソース測定単位を使用して累積ドリフトを保つことを含む。ある単位から別の単位への変換は概して、丸めまたは切り捨て誤差をもち込む場合がある除算演算を伴う。そのような切り捨て誤差の累積が、ホストとデバイスとの間の時間の解釈の相違につながる場合がある。ソース測定単位での累積を保つことによって、いかなる切り捨て誤差も一時的であり、システムによって些細なジッタとみなされるはずである。

【００３５】

さらなる因子が、最大許容システムジッタである。妥当な許容システムジッタは、オーディオシステムによって実ドリフトとして解釈されるのを回避するための正確な1つよりも少ないオーディオサンプルである。したがって、許容システムジッタは、オーディオサンプリング周波数の関数であってもよい。許容ジッタが十分に小さい場合、純粋なソフトウェア実装では、イベントにタイムスタンプするための割込みをサービスするために十分に高速に反応することができない場合があるので、ハードウェア支援が必要な場合がある。

【００３６】

これらの検討が与えられると、USBオーディオデバイスの瞬時周波数フィードバックをクロックソースとみなすとき、式6が導出されてもよい。そのような事例において、F_fは、USBデバイスがUSBホストに報告する、フレーム当たりのオーディオサンプルの平均数である。瞬時周波数F_fは、

$Period_{FS}=2^{10-bRefresh}$ のフレーム （式3）

ごとにFS USB転送モードでホストに報告される。

【００３７】

または、HS USB転送モードでは、

$Period_{HS}=2^{(binterval-1)}$ のマイクロフレーム （式4）

ごとである。

【００３８】

瞬時ドリフトはしたがって、

$\Delta drift=F_{fk}-F_{fk-1}$ （式5）

【００３９】

であり、ホストがフィードバック

$Ticks_{conv}(D)=(D*1000)/f_s*19.2MHz$ （式6）

を受信したときに計算される。

【００４０】

ここで、f_sはサンプリング周波数である。19.2MHzは、1つの例示的な高解像度システムタイマの速度であることに留意されたい。高解像度システムタイマが異なる速度を有する場合、異なる値がここで代入されるべきであり、式6は以下の汎用式になる。

$Ticks_{conv}(D)=(D*1000)/f_s*f_{timer}$ （式6A）

【００４１】

仮想パケットが1つの仮想フレームであるという定義的等価性から生じるUSB2.0信号からクロックをリカバリするには困難を伴う。したがって、非線形データストリームからクロックをリカバリするためのソリューションが必要とされる。そのようなソリューションは、各クロック水晶が少なくとも500ppmの精度を有するという仮定を付け加える。仮想

フレーム当たりのサンプルの数は、次のように定義される。

$$numSamplesPerVirtualFrame=f_s/f_t*2^{(binterval-1)} \quad (式7)$$

【００４２】

ここで、f_sはサンプリング周波数であり、f_tはサービス間隔周波数であり、bintervalは上で定義された通りである。表記を簡単にするために、numSamplesPerVirtualFrameは、NSPVFと短縮されてもよい。

【００４３】

さらに、整列乗数が必要とされ、次のように定義される。

$$alignmentMultiplier=1000000/GCD(MOD(NSPVF*1000000,1000000),1000000) \quad (式8)$$

【００４４】

ここで、1000000は、分数精度を増やすための非常に大きいベース10値として恣意的に選ばれる。式7および8から、期待されるサンプル数は、次のように算出されてもよい。

$$expectedNumSamples=NSPVF*alignmentMultiplier \quad (式9)$$

【００４５】

alignmentMultiplierは、安定したドリフト判断が可能になる前のホストによって必要とされる最小数の仮想フレームを表す。expectedNumSamplesは、受信されることが期待されるサンプルの数である。NSPVFは、視覚的明瞭さについての中間変数であり、浮動小数点ではない。受信された仮想フレームの各alignmentMultiplier数について、$\Delta drift$は、以下によって計算される。

$$\Delta drift=numSamplesReceived-expectedNumSamples \quad (式10)$$

【００４６】

したがって、オーディオセッションの開始からの正味のドリフトは、以下によって計算される。

$$D=D_{net\ drift}+\Delta drift \quad (式11)$$

【００４７】

D個のオーディオサンプルからシステムタイマ(Qtimerと称されることがある)ティックへの変換は、以下のようになる。

$$Ticks_{conv}(D)=D_{net\ drift}/f_s*19.2MHz \quad (式12)$$

【００４８】

やはり、19.2MHzは高解像度システムタイマの速度であることに留意されたい。高解像度システムタイマが異なる値を有する場合、そのような異なる値はここで代入されるべきであり、以下を得る。

$$Ticks_{conv}(D)=D/f_s*f_{timer} \quad (式12A)$$

【００４９】

ドリフト情報およびクロック検出情報が上で概説されたので、レートマッチングが行われてもよい。レートマッチングを用いると、以下で概説するように、均一なサンプルサイズが作成され、アプリケーションプロセッサに送信される場合がある。ただし、均一なサンプルサイズに対処する前に、レートマッチングを説明するためにより多くの数学的計算が提示される。特に、これはどのように$ticks_{offset}$を算出するかを定義するのを助ける。

【００５０】

欠けているドリフトを思い起こされたい。

$$ticks_{next}=ticks_{reference}+ticks_{offset} \quad (式13)$$

【００５１】

ここで、$ticks_{offset}$は以下のように定義される。

【数１】

$$ticks_{offset} = \frac{f_t}{1\ khz} * \frac{f_d}{f_s} * i \qquad (式14)$$

ここで、f_dは配信周波数であり、iはあらゆる$tick_{next}$においてインクリメントし、$i=f_s$のとき、iがオーバーフローするのを回避するためにラップアラウンドする。ラップアラウンドポイントにおいて、$ticks_{reference}=ticks_{next}$であり、したがってi=0である。

【００５２】

上記で説明した数学的計算を用意した上で、本開示の例示的な態様について、ここで説明する。この点において、図3は、本開示の例示的な態様を実装するモバイル通信デバイス300においてオーディオ(および、おそらくビデオ)がどのように扱われるかの簡略化ブロック図である。

【００５３】

モバイル通信デバイス300は、アプリケーションプロセッサ302およびADSP304を含む。例示的な態様では、アプリケーションプロセッサ302およびADSP304は単一のSoC306の中にあってもよい。同様に、概念的には別々のプロセッサとして記載されるが、これらのプロセッサは、単一のホストプロセッサの一部であってもよい。またさらに、「アプリケーションプロセッサ」または「ADSP」など、特定の機能として記載されるが、従来はそのような呼称で呼ばれていない他のプロセッサが、本開示の範囲から逸脱することなく、匹敵する機能性をやはり実装してもよいことを了解されたい。アプリケーションプロセッサ302は、USBハードウェアコントローラ308と通信してもよく、コントローラ308は、USBインターフェース312を通して、ヘッドセットなどのUSB周辺機器310と通信し、インターフェース312は、USBレセプタクル、USBコネクタ、およびUSBケーブルを含んでもよい。

【００５４】

図2のUSB周辺機器202と同様に、USB周辺機器310は、非同期、同期、適応、または混合クロック同期モードをサポートしてもよく、1つまたは複数のPLL(2つが図示される)またはDLL(図示せず)を含んでもよい。USB周辺機器310は、マイクロフォン(上述したように、キャプチャと称されることがある)を通すなどして、データ(Data Inと称される)をならびにヘッドフォン中のスピーカーを通すなどして(上述したように、再生と称されることがある)、出力データ(Data Outと称される)を受信する場合がある。データは、USBインターフェース312を通して、モバイル通信デバイス300との間で渡される。

【００５５】

ADSP304は、UACドライバ314を含んでもよい。UACドライバ314は、USBハードウェアコントローラ308と通信するのに、ホストコントローラインターフェース(HCI)(図示せず)を使用してもよい。従来のシステムでは、ADSP304はUSBハードウェアコントローラ308と通信しないので、UACドライバ314中にはHCIがない。ただし、本開示の例示的な態様は、USBハードウェアコントローラ308とADSP304との間の通信を可能にする。したがって、そのような通信を遂行するために、HCIが設けられる場合がある。UACドライバ314は、不安定であり可変サイズのデータフレームをUSBハードウェアコントローラ308から受信する。

【００５６】

本開示の例示的な態様は、UACドライバ314に1つまたは複数のバッファ316を追加し、かつ高解像度システムタイマ318をUACドライバ314に結合し、そうすることによって、UACドライバ314は、安定した、厳密な、固定のデータフレームサイズをアプリケーションプロセッサ302(またはアプリケーションを扱う他のプロセッサ)中のデータ処理回路機構320に渡すことができるようになる。またさらに、UACドライバ314は、信号322を通して、正味の再生およびキャプチャ遅延をデータ処理回路機構320に与えてもよい。デ

ータ処理回路機構320に均一なデータフレームを与えることによって、アプリケーションレイヤアルゴリズム324は、データをそれ程多量にバッファリングしなくてもよく、図2のデータ処理回路機構218に関連付けられた訂正を実施しなくてもよい。アプリケーションレイヤアルゴリズム324が均一なデータフレームを受信するとしても、アプリケーションプロセッサ302は、ドリフト訂正情報および/またはジッタ問題に作用するように信号322を処理するのを支援する場合があるASRC326を含んでもよい。やはり、アプリケーションプロセッサ302は、単一のマイクロプロセッサとしてADSP304とマージされてもよく、または異なるベンダによって異なる名称を与えられる場合があることに留意されたい。

【００５７】

図3は、UACドライバ314をADSP304中に位置付けることを企図しているが、図4Aおよび図4Bに示すように、他の位置も可能であることを了解されたい。

【００５８】

この点において、図4Aは、マイクロフォンなどからデータをキャプチャし、データをUACデータレギュレータ(UACデータreg)404に与えるデジタルオーディオコンバータ(DAC)402付きヘッドセット400(または他のUSB周辺機器)を示す。UAC reg404は、パケットサイズを均一にし、パケットをハードウェアコントローラ406に与え、コントローラ406は、パケットをケーブル408を介してUSBホスト410に渡す。USBホスト410は、ホストハードウェアコントローラ412を用いてパケットを受信する。アプリケーションレイヤ(具体的には図示せず)中のアプリケーション414(図面ではAPPとラベル付けされる)は、均一なパケットを受信し、よく理解されている方法で処理する。そのような配置では、USBホスト410は、図2のUSBホストと同様に動作してもよいが、ヘッドセット400がUSBホスト410に送信する均一なパケットから利益を得る。ヘッドセット400中の増大した回路機構は、ヘッドセット400のコストを増加させる場合があるが、レガシーUSBホストに利益を提供する場合がある。

【００５９】

図4Bにおいて、USBホスト410は不変のままであるが、データレギュレータをヘッドセット400中に置く代わりに、ドングル420などの中間デバイス420中にUACデータレギュレータ418が設けられる。ドングル420は、ケーブル422のホスト側422Aまたは周辺機器側422Bにあることが可能である。つまり、ケーブル422は、ドングル420がUSBホスト410のUSBレセプタクルに挿入された状態でドングル420とヘッドセット424との間で伸長してもよく、またはケーブル422は、ドングル420がヘッドセット424のUSBレセプタクルに挿入された状態で、USBホスト410とドングル420との間で伸長してもよい。さらに別の可能性(図示される)として、ドングルはケーブル422中にあってもよく、ケーブル422は、USBホスト410およびヘッドセット424のそれぞれのレセプタクルに挿入される。

【００６０】

図5は、図3のUACドライバ314の中に実装される場合があるデータレギュレータのブロック図である。バッファ316(図5ではFIFOとも称される)は、可変サイズのデータパケット500を受信する。帯域内ドリフト検出器502は、データ利用可能割込み信号504を受信すると、バッファ316中のデータパケット500のサイズを読み取る。代替として、帯域外ドリフト検出器506が、非同期フィードバックパケット信号508およびデータ利用可能割込み信号504を受信する。検出器502または506のうちの1つが、マルチプレクサ510によって読取りを受ける。マルチプレクサ510は、検出タイプ設定信号512によって、検出器502および506の出力の間で選択する。マルチプレクサ510は、デバイスドリフト累算器514に信号を出力する。同時に、データ利用可能割込み信号504がローカルクロックドリフト検出器516に与えられ、検出器516は、ローカルクロックドリフト累算器518に信号を与える。加算器520が、ローカルクロックドリフト累算器518の出力($D_{app\text{-}usb}$)からデバイスドリフト累算器514出力($D_{device\text{-}usb}$)を減算し、信号522を出力する。信号522は、$D_{app\text{-}usb}$-$D_{device\text{-}usb}$に対応する。

【００６１】

図5への参照を続けると、データ利用可能割込み信号504は、初期参照ハンドラ524にも与えられる。初期参照ハンドラ524は、高解像度クロック関数526にカウンタ読取りを出力する。高解像度クロック関数526はまた、ローカルクロックドリフト検出器516からカウンタ読取りを受信する。高解像度クロック関数526はまた、クロック値を変えさせることになるハイレゾタイマF_t値設定を受信する場合がある。この値は、動作中に変わる見込みはないが、システム初期化時などに設定されることが可能であることに留意されたい。高解像度クロック関数526は、高解像度システムタイマ318と相互運用する。初期参照ハンドラ524も、ジッタ遅延要素528に加えられ、タイムスタンププラス遅延信号530を開始するための初期Trefを設定するのに使用される。

【００６２】

バッファ316は、データ信号532(「データ読取り」とラベル付けされる)をデータ配信ハンドラ534に出力し、ハンドラ534は、高解像度システムタイマ318の出力536も受信する。データ配信ハンドラ534はまた、ASRCがどのサイズバッファを処理することを期待するかを示す出力バッファサイズ設定コマンドを(おそらくはASRC326から)受信してもよい。信号530は加算器538に与えられ、加算器538は、信号530にTrefを加え、中間信号540を生成し、信号540にはToffsetが加えられて信号542が生成され、信号542は加算器544に渡される(加算器544は本質的に、式6または式12のいずれかを必要に応じて実施する)。加算器544は、信号542、信号522、および出力536を加えて、合成タイムスタンプ546を生成する(本質的には式2)。データ配信ハンドラ534は、加算器544向けの稼働コマンドを出力し、一定数のサンプルをASRC326に与える。ASRC326はまた、合成タイムスタンプ546を受信し、再サンプリングされたデータ548を出力する。具体的には図示されていないが、サンプリング周波数設定コマンドも、上記のように算出を支援するために受信される場合がある。

【００６３】

ある例示的な態様では、このデータレギュレータはソフトウェアとして実装される。別の例示的な態様では、このデータレギュレータは、ハードウェアで実装されてもよい。

【００６４】

図6は、データプロセッサ中のアプリケーションが図3のUACドライバ314を使用したいときに生じる場合がある信号およびプロセスを表す信号フロー図である。最初に、アプリケーションが、アクティブ化セットアップ段階においてセットアップ情報を提供する。セットアップ情報は、サンプリングレート、バス転送周波数、バッファサイズ、クロックリカバリモードなどを含んでもよい。このセットアップ情報は、UACドライバ314のデータレートレギュレータ(図5参照)に与えられる。データレートレギュレータは、リクエストされているレートで、USBハードウェアコントローラ308からデータを正確に、安定して(ジッタなしで)どのように配信するかを算出する。この算出のためのプロセスは上述された。この図の中のタイマ/クロック要素は、図3の高解像度システムタイマ318であるが、他のタイマを使用することもできよう。

【００６５】

図6は、アプリケーションプロセッサ302など、データプロセッサ中のアプリケーションがUACドライバ314を使用したいときに生じる場合がある信号およびプロセスを表す信号フロー図600である。最初に、アプリケーションがアクティブ化セットアップ段階においてセットアップ情報を提供する(ブロック602)。アプリケーションプロセッサ302は、ASRC326において入力および出力サンプリング周波数を設定し、入力サンプリングレート周波数、バス転送周波数、サービス間隔(バス転送周波数以上である)、出力バッファサイズ、クロックリカバリモード(非同期、適応、または同期)、任意のハードウェアインターフェース固有セットアップパラメータを送信し、バッファ316用のどの物理メモリも、UACドライバ314に、特にUACドライバ314中のデータレギュレータに登録する。最後に、アクティブ化コマンド(信号604)がデータレギュレータに送信される。データレギュレータは、ハードウェアインターフェース固有セットアップパラメータをUSBハードウェアコントローラ308に渡し(信号606)、書き込むべき次の空きバッファ空間をプログラムする(

信号608)。USBハードウェアコントローラ308は、データ準備完了イベント信号610をデータレギュレータに送信する。この信号610は、データレギュレータに、高解像度システムタイマ318を読み取らせ(信号612)、USBハードウェアコントローラ308からデータサイズを読み取らせ(信号614)、クロック値をTrefの中に記憶することと、Tjitter(バッファサイズから導出され、明示的にフィードバック駆動でない場合は、受信されたデータサイズ)をTrefに加えることと、i=0、$D_{device-usb}$=0、$D_{app-usb}$=0を初期化することと、次のToffsetを計算し、Tnext(式2)を計算することとを含む一連のアクションを実施させる(全体としてブロック616参照)。データレギュレータは次いで、Tnextを高解像度システムタイマ318用にプログラムし(信号618)、書き込むべき次の空きバッファ空間をプログラムする(信号620)。

【００６６】

図6への参照を続けると、システムは定常状態に入り、データレギュレータは、USBハードウェアコントローラ308から次のデータ準備完了イベントを受信し(信号622)、これはクロック読取り信号624と、データレギュレータに正味のドリフト($D_{device-usb}$および$D_{app-usb}$)を更新させるデータサイズ読取り信号626とをトリガする(全体としてブロック628参照)。

【００６７】

どこかの時点で、USBハードウェアコントローラ308は、非同期クロックフィードバックイベント(信号630)をデータレギュレータに送信してもよく、このイベントは、データレギュレータに、$D_{device-usb}$を更新させる(全体としてブロック632参照)。

【００６８】

どこかの他のときに、高解像度システムタイマ318は、タイマ満了イベント信号634をデータレギュレータに送信してもよい。この信号634に応答して、データレギュレータは、iを1だけインクリメントし、iがサンプリング周波数に等しい場合、TrefをTnextに、およびi=0と設定し、次のToffsetを計算し、式2を計算してもよい(全体としてブロック636参照)。データレギュレータは、データ利用可能信号638をアプリケーションプロセッサ302に送信し、Tnextをプログラムし(信号640)、書き込むべき次の空きバッファ空間をプログラムしてもよい(信号642)。アプリケーションプロセッサ302は、正味のドリフトまたはタイムスタンプをデータレギュレータから読み取り(信号644)、UACドライバ314(信号646)および/またはUSBハードウェアコントローラ308(信号646A)中のバッファ316からデータを読み取る。

【００６９】

アプリケーションプロセッサ302は、新たな正味のドリフトおよび前の正味のドリフトから訂正すべきサンプルの数を計算し(ブロック648)、そのファイルシステムに、たとえばデータ、データ長、訂正すべきサンプル、および変数を訂正するための持続時間をもつ書込みコマンドを使用することによって、データを書き込む。データはボイスパケットであってもよいことに留意されたい。必要な場合、ドリフト訂正は、知覚可能グリッチを削減するために、構成可能期間に拡張されてもよい。ただし、拡張された期間を用いても、そのような訂正が、従来のシステムにおいて使用されることがある10秒ではなく、25msのオーダーで起こることが期待される。次いで、プロセスは非アクティブ化する(ブロック660)。

【００７０】

本開示の追加態様は、エラーなしドリフト検出を提供するとともに将来の計画される省電力イニシアチブをサポートするための技法を提供することにさらに留意されたい。この点において、比較的一般的な44.1kHzなどの分数サンプリングレートは、周辺デバイスにおける累算器とホストにおける累算器との間の位相不一致のせいで、ドリフトの誤検出に役立つことを了解されたい。ドリフト検出を支援するためのタイムスタンプを含むシグナリングプロトコルとは対照的に、USBプロトコルは、周辺デバイスからホストへのタイムスタンプを含まない。そうではなく、ホストは、パケット化されたUSBデータを受信するだけである。各USBパケットの中では、データの量は可変である。分数サンプリングレー

トおよび未知のパケットサイズに伴う問題は、当業界において十分に立証されている。普通のソリューションは、10分間などの長期間に渡ってサンプルの時間平均をとり、次いでドリフトの訂正を実施するものである。サンプルの時間平均を組み立てる際の長い遅延は、訂正が適用される前に待ち時間を生じる。訂正が適用されるまで、ユーザは、低下したオーディオエクスペリエンスを受ける場合がある。同様に、訂正の粒度は、瞬時またはランダムドリフトイベントには適切でない場合がある。

【００７１】

本開示の例示的な態様は、エラーなしドリフト検出を可能にする。これは、例の使用により、最もよく説明される。サンプリング周波数(Fs)が44.1kHzであること、USBバス転送速度が1000Hz(すなわち、毎ミリ秒1サンプル)であること、および11のbinterval(パケット当たりのサンプル)を仮定すると、ホストは、間隔ごとに45158.4サンプルを受信することを期待することになる。分数サンプルは、USB規則の下で送信不可能である。周辺デバイス累算器は、ホストにサンプルが伝送されると開始するが、ホスト累算器は、受信の後まで遅延されるので、累算器は位相はずれになる。第2の間隔において、周辺累算器は90316.8である。やはり、ホスト累算器に相対したドリフトとして現れるのは分数サンプルである。外部ドリフトなしで時間がたつと、このドリフトは、1と0との間をトグルするが、時々必要とされない訂正を行わせる場合がある。

【００７２】

前のソリューションでのようにドリフトを時間平均する代わりに、本開示の例示的な態様は、分数剰余を評価し、整数に達するのに要求される間隔の数を見つける。本例では、分数剰余が0.4の場合、整数に達するのに要求される間隔の数は5である。(0.4=2/5、分母が5であるので、5つの間隔)。UACドライバ314は、そのように算出された間隔の数によって判断された境界において累算器を調べてもよい。したがって、この例では、UACドライバ314は、5つの間隔ごとにドリフトを確認する。分数サンプリングレートによって引き起こされるファントムドリフトは存在しないので、ドリフトが検出された場合、それは、訂正(すなわち、補間またはデシメーションなど)が行われなければならない実ドリフトである。さらに、中間サンプル中のドリフトを無視することによって、算出が見合わせられてもよく、これは省電力という結果になる場合がある。

【００７３】

USBプロトコルは、2つの形のドリフト報告を企図する。第1は、インバウンド信号が調査され、ドリフトを判断するために既知の値と比較される暗黙的ドリフト検出である。第2は、周辺デバイスによってホストに送信されるドリフトの明示的帯域外シグナリングであり、周辺デバイスは、受信されたサンプルを期待される数のサンプルと比較し、これらの2つの値の間のいかなるドリフトも報告し返す。USBプロトコルは、暗黙的ドリフト検出がどのように実施されるかについて言及しておらず、USBプロトコルは、(暗黙的または明示的のいずれかで)検出された任意のドリフトについてホストがどのように訂正するかについても言及していない。本開示は、上記のいくつかの式ならびにドリフト検出およびその訂正を扱うためのプロセスについて説明した。図7～図10は、オーディオソース(図7および図8)とオーディオシンク(図9および図10)との両方についての2つの可能ドリフト報告可能性ならびに訂正プロセスを示す。特に、図7は、オーディオソース、すなわちマイクロフォン700のための帯域内ドリフト報告プロセスを示す。データは、マイクロフォン700によってキャプチャされ、一定レートで、可変サイズのデータパケット中で、USBデバイスドライバ704を通してUSBホスト中のUSBホストドライバ706に渡される(ブロック702)。USBホストドライバ706は、マイクロフォン700からのデータからドリフト情報を暗黙的に導出し、抽出されたドリフト情報は、オーディオクライアントへの配信のタイミングをとるためのTref+Toffsetを判断し、タイマをプログラムするのに使用され(ブロック708)、その間データはバッファ710中に記憶される。Tref+Toffsetを判断するための公式については上述された。ブロック708の出力に基づくタイマトリガ712において、可変レートでの一定数のパケット(ブロック714)が、バッファ710からASRC716に送信される。同時に、ドリフト情報は、正味の再生遅延を報告し(ブロック718)、合成タイムスタ

ンプを生成する(ブロック720)のに使用される。ASRC716は、再サンプリングされたデータを出力する(ブロック722)。一定数のパケットは実際固定であり、レートを変えるとドリフトが訂正される。つまり、パケット配信は、1つのドリフトを訂正するように加速され、または他の方向でのドリフトを訂正するように減速される場合がある。

【００７４】

同様に、図8は、実質的に同様であるが、マイクロフォン800のための帯域外ドリフト報告プロセスを反映している。特に、ドリフト検出は、マイクロフォン800の出力に基づいて、USBデバイスドライバ802によって実施される。USBデバイスドライバ802は次いで、帯域外ドリフト報告(ブロック804)を出力し、また一定レートでの可変サイズのデータパケット(ブロック806)を送信する。ドリフト情報とデータとの両方が、USBホスト中のUSBホストドライバ808に与えられる。ドリフト情報は、オーディオクライアントへの配信のタイミングをとるためのTref+Toffsetを判断し、上で説明した式を使用してタイマをプログラムする(ブロック810)のに使用され、その間データはバッファ812中に記憶される。ブロック810の出力に基づくタイマトリガ814において、バッファ812は、一定数のパケットを可変レートでASRC818に送信する(ブロック816)。同時に、ドリフト情報は、正味の再生遅延を報告し(ブロック820)、合成タイムスタンプを生成する(ブロック822)のに使用される。ASRC818は、再サンプリングされたデータを出力する(ブロック824)。やはり、可変レートの使用によりドリフト訂正が可能になる。

【００７５】

対照的に、図9および図10は、再生経路上でのドリフトの影響を探査する。この点において、図9は、帯域内ドリフト報告プロセスを示す。マイクロフォン900が、図7のマイクロフォン700として作用してもよいが、より興味深いのはスピーカー902である。スピーカー902は、USBデバイスドライバ904からデータを受信する。USBデバイスドライバ904は、USBホストドライバ906からデータを受信する。USBホストドライバ906は、USBホストドライバ906に着信したデータを上述したUSB参照と比較して、ドリフト情報を判断する。このドリフト情報は、オーディオクライアントへの配信のタイミングをとるためのTref+Toffsetを判断し、上述した式を使用して、タイマをプログラムする(ブロック908)のに使用される。この判断は、タイマトリガを生成し(ブロック910)、正味記録遅延を報告し(ブロック912)、合成タイムスタンプを作成する(ブロック914)のを助けるのに使用される。タイマトリガ(ブロック910)において、可変レートでの一定数のパケットがフェッチされ(ブロック916)、オーディオモジュール918に与えられ、モジュール918は、パケットをバッファ920にバッファリングする。バッファ920は、一定レートでの可変サイズのデータパケット(ブロック922)を解放し、データパケットをUSBホストドライバ906に与え、ドライバ906は、パケットをUSBデバイスドライバ904を通してスピーカー902に渡す。可変サイズのデータパケットの使用によりドリフトが訂正される。スピーカー方向でのドリフトの訂正は、マイクロフォン900とスピーカー902との両方が同じソースによりクロッキングされるならば、帯域内ドリフト検出器によりUSBホストドライバ906において検出されたドリフトから推論することが可能である。

【００７６】

同様に、図10は、帯域外ドリフト報告プロセスを示す。マイクロフォン1000が、上で説明した図8のマイクロフォン800として作用してもよい。より興味深いのは、スピーカー1002である。スピーカー1002は、帯域外ドリフト情報およびデータ(ブロック1004)をUSBデバイスドライバ1006に渡す。USBデバイスドライバ1006は、USBホストドライバ1008からデータを受信し、同様に帯域外ドリフト情報をUSBホストドライバ1008に渡す。このドリフト情報は、オーディオクライアントへの配信のタイミングをとるためのTref+Toffsetを判断し、タイマをプログラムする(ブロック1010)のに使用される。この判断は、タイマトリガを生成し(ブロック1012)、正味記録遅延を報告し(ブロック1014)、合成タイムスタンプを作成する(ブロック1016)のを助けるのに使用される。タイマトリガ(ブロック1012)において、可変レートでの一定数のパケットがフェッチされ(ブロック1018)、オーディオモジュール1020に与えられ、モジュール1020は、パケットをバッファ

1022にバッファリングする。バッファ1022は、一定レートでの可変サイズのデータパケット(ブロック1024)を解放し、データパケットをUSBホストドライバ1008に与え、ドライバ1008は、パケットをUSBデバイスドライバ1006を通してスピーカー1002に渡す。やはり、可変サイズのデータパケットの使用により、ドリフト訂正が可能になる。

【００７７】

上述したように、例示的な態様は、将来の企図される省電力も可能にする。この可能性は、可変データおよびサンプリングレートを扱うのに使用される汎用(アグノスティックと称されることがある)アルゴリズムによって可能にされる。つまり、上の式において、式は、サンプリングレートとしてのアグノスティックf_sおよびバス転送速度(すでにFS、SS、およびHSを企図している)としてのf_tで始まる。これらのアグノスティック値をアプリケーションレイヤアルゴリズム324において使用することによって、他の新たなサンプリングレートまたは他の非標準サンプリングレートが順応される。アグノスティック手法は、DLLの適切な推定を可能にする。binterval(パケット当たりのサンプルの数)の増加により、パケットのサイズが増大し、バッファ316を埋めるのにかかる時間も増大することを了解されたい。バッファ316が埋められている間、アプリケーションプロセッサ302はアイドルであるので、アプリケーションプロセッサ302は、低電力モードまたはスリープモードに入れられてもよい。バッファ316を埋めるのに長くかかるほど(すなわち、パケットごとにより多数のサンプル)、アプリケーションプロセッサ302はスリープモードに長くいる場合がある。アプリケーションプロセッサ302がスリープモードに長くいるほど、より多くの電力が節約される。したがって、当業界には、パケット当たりのサンプルの数を増大するというプレッシャーがある。アプリケーションレイヤアルゴリズム324において汎用bintervalを有することによって、本開示の例示的な態様は、オーディオデバイス記述子の中により大きいbinterval値を受け入れ、したがってパケット当たりのサンプルの数におけるいかなる将来の変更にも順応し、したがって将来の省電力を可能にしてもよい。

【００７８】

本明細書で開示する態様によるアイソクロナスデータストリームを制御するためのシステムおよび方法は、任意のプロセッサベースのデバイス内で提供されるか、これに集積されてもよい。例には、限定はしないが、セットトップボックス、エンターテインメントユニット、ナビゲーションデバイス、通信デバイス、固定ロケーションデータユニット、モバイルロケーションデータユニット、全地球測位システム(GPS)デバイス、モバイルフォン、セルラーフォン、スマートフォン、セッション開始プロトコル(SIP)フォン、タブレット、ファブレット、サーバ、コンピュータ、ポータブルコンピュータ、モバイルコンピューティングデバイス、ウェアラブルコンピューティングデバイス(たとえば、スマートウォッチ、ヘルスまたはフィットネストラッカー、アイウェアなど)、デスクトップコンピュータ、携帯情報端末(PDA)、モニタ、コンピュータモニタ、テレビ、チューナ、ラジオ、衛星ラジオ、音楽プレーヤ、デジタル音楽プレーヤ、ポータブル音楽プレーヤ、デジタルビデオプレーヤ、ビデオプレーヤ、デジタルビデオディスク(DVD)プレーヤ、ポータブルデジタルビデオプレーヤ、自動車、車両構成要素、アビオニクスシステム、ドローン、およびマルチコプターが含まれる。

【００７９】

この点において、図11は、本明細書に記載されるドリフト検出、レートマッチングおよび均一なパケットアセンブリを実施するUSBシステムを利用することができるプロセッサベースシステム1100の例を示す。この例では、プロセッサベースシステム1100は、各々が1つまたは複数のプロセッサ1104を含む、1つまたは複数の中央処理ユニット(CPU)1102を含む。CPU1102は、一時的に記憶されたデータへの高速アクセスのためにプロセッサ1104に結合されるキャッシュメモリ1106を有してもよい。CPU1102は、システムバス1108に結合され、プロセッサベースシステム1100内に含まれるマスタデバイスとスレーブデバイスとを相互結合することができる。よく知られているように、CPU1102は、アドレス情報、制御情報、およびデータ情報をシステムバス1108を介して交換することによって、これらの他のデバイスと通信する。たとえば、CPU1102は、スレーブデバイ

スの一例として、メモリコントローラ1110にバストランザクション要求を通信することができる。図11には示さないが、複数のシステムバス1108が設けられてよく、各システムバス1108は異なるファブリックを構成する。

【００８０】

他のマスタおよびスレーブデバイスがシステムバス1108に接続されることが可能である。図11に示されるように、これらのデバイスは、例として、メモリシステム1112、1つまたは複数の入力デバイス1114、1つまたは複数の出力デバイス1116、1つまたは複数のネットワークインターフェースデバイス1118、および1つまたは複数のディスプレイコントローラ1120を含むことができる。入力デバイス1114は、限定はしないが、入力キー、スイッチ、音声プロセッサなどを含む、任意のタイプの入力デバイスを含むことができる。出力デバイス1116は、限定はしないが、オーディオインジケータ、ビデオインジケータ、他の視覚インジケータなどを含む、任意のタイプの出力デバイスを含むことができる。ネットワークインターフェースデバイス1118は、ネットワーク1122との間のデータの交換を可能にするように構成される任意のデバイスであることが可能である。ネットワーク1122は、限定はしないが、ワイヤードネットワークまたはワイヤレスネットワーク、プライベートネットワークまたは公衆ネットワーク、ローカルエリアネットワーク(LAN)、ワイヤレスローカルエリアネットワーク(WLAN)、ワイドエリアネットワーク(WAN)、Bluetooth(登録商標)ネットワーク、およびインターネットを含む、任意のタイプのネットワークであることが可能である。ネットワークインターフェースデバイス1118は、必要に応じて、任意のタイプの通信プロトコルをサポートするように構成されることが可能である。メモリシステム1112は、1つまたは複数のメモリユニット1124(0～N)を含むことができる。

【００８１】

CPU1102はまた、システムバス1108を介してディスプレイコントローラ1120にアクセスして、1つまたは複数のディスプレイ1126に送信される情報を制御するように構成されてもよい。ディスプレイコントローラ1120は、表示されるべき情報を1つまたは複数のビデオプロセッサ1128を介してディスプレイ1126へ送信し、1つまたは複数のビデオプロセッサ1128は、表示されるべき情報をディスプレイ1126にとって適切なフォーマットに処理する。ディスプレイ1126は、限定はしないが、陰極線管(CRT)、液晶ディスプレイ(LCD)、プラズマディスプレイ、発光ダイオード(LED)ディスプレイなどを含む、任意のタイプのディスプレイを含むことができる。

【００８２】

本明細書で開示する態様に関して説明した様々な例示的な論理ブロック、モジュール、回路、およびアルゴリズムが、電子ハードウェア、メモリの中もしくは別のコンピュータ可読媒体の中に記憶されるとともにプロセッサもしくは他の処理デバイスによって実行される命令、または両方の組合せとして実装されてもよいことを当業者はさらに了解されよう。本明細書で説明するデバイスは、例として、任意の回路、ハードウェア構成要素、集積回路(IC)、またはICチップにおいて採用される場合がある。本明細書で開示したメモリは、任意のタイプおよびサイズのメモリである場合があり、所望の任意のタイプの情報を記憶するように構成される場合がある。この互換性について明確に説明するために、様々な例示的な構成要素、ブロック、モジュール、回路、およびステップについて、上記では概してそれらの機能性に関して説明してきた。そのような機能性がどのように実装されるのかは、特定の適用例、設計選択、および/またはシステム全体に課される設計制約によって決まる。当業者は、説明された機能性を特定の適用例ごとに様々な方法で実装してよいが、そのような実施態様の決定は、本開示の範囲からの逸脱を引き起こすものと解釈されるべきではない。

【００８３】

本明細書で開示する態様に関連して説明した様々な例示的な論理ブロック、モジュール、および回路は、プロセッサ、デジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)もしくは他のプログラマブル論理デ

バイス、個別ゲートもしくはトランジスタ論理、個別ハードウェア構成要素、または本明細書で説明する機能を実行するように設計されたそれらの任意の組合せを用いて実装または実施されてもよい。プロセッサは、マイクロプロセッサであってもよいが、代替としてプロセッサは、任意の従来のプロセッサ、コントローラ、マイクロコントローラ、またはステートマシンであってもよい。プロセッサはまた、コンピューティングデバイスの組合せ(たとえば、DSPとマイクロプロセッサとの組合せ、複数のマイクロプロセッサ、DSPコアと連携した1つもしくは複数のマイクロプロセッサ、または任意の他のそのような構成)として実装されてもよい。

【００８４】

本明細書で開示する態様は、ハードウェアにおいて具現され、またハードウェアに記憶された命令において具現される場合があり、命令は、たとえば、ランダムアクセスメモリ(RAM)、フラッシュメモリ、読取り専用メモリ(ROM)、電気的プログラマブルROM(EPROM)、電気的消去可能プログラマブルROM(EEPROM)、レジスタ、ハードディスク、リムーバブルディスク、CD-ROM、または当技術分野において知られている任意の他の形態のコンピュータ可読媒体内に存在する場合がある。例示的な記憶媒体は、プロセッサが記憶媒体から情報を読み取り、記憶媒体に情報を書き込むことができるように、プロセッサに結合される。代替として、記憶媒体はプロセッサと一体化されてもよい。プロセッサおよび記憶媒体は、ASICに存在してもよい。ASICは、リモート局内に存在してもよい。代替として、プロセッサおよび記憶媒体は、個別構成要素としてリモート局、基地局、またはサーバの中に存在してもよい。

【００８５】

本明細書の例示的な態様のいずれかにおいて説明した動作ステップは、例について説明するためのものであることにも留意されたい。述べた動作は、示されたシーケンス以外の多くの異なるシーケンスで実施されてもよい。さらに、単一の動作ステップで説明される動作は、実際にはいくつかの異なるステップで実施されてもよい。さらに、例示的な態様において説明する1つまたは複数の動作ステップは組み合わせられる場合がある。当業者には容易に明らかになるように、フローチャート図に示される動作ステップが数多くの異なる変更を受ける場合があることを理解されたい。情報および信号が様々な異なる技術および技法のいずれかを使用して表される場合があることも当業者は理解されよう。たとえば、上記の説明全体に渡って参照される場合があるデータ、命令、コマンド、情報、信号、ビット、シンボル、およびチップは、電圧、電流、電磁波、磁場もしくは磁気粒子、光場もしくは光学粒子、またはそれらの任意の組合せによって表されてもよい。

【００８６】

本開示の上記の説明は、あらゆる当業者が本開示を作成または使用することを可能にするために提供されている。本開示の様々な変更が当業者に容易に明らかになり、本明細書で定義する一般原理は、本開示の趣旨または範囲から逸脱することなく他の変形形態に適用されてもよい。したがって、本開示は、本明細書において説明される例および設計に限定されるものでなく、本明細書において開示される原理および新規の特徴と一致する最も広い範囲を与えられるべきである。

【符号の説明】

【００８７】

100 モバイル通信デバイス
102 USBタイプCレセプタクル
104 USBタイプCコネクタ
106 USBケーブル
108 デジタルオーディオヘッドセット
110 スピーカー
112 ヘッドフォン
114 マイクロフォン
200 モバイル通信デバイス

202 USB周辺機器
206 USBケーブル
208 USBハードウェアコントローラ
210 システムオンチップ(SoC)
212 オーディオデジタル信号プロセッサ(ADSP)
214 アプリケーションプロセッサ(AP)
216 USBオーディオクライアント(UAC)ドライバ
218 データ処理回路機構
220 バッファ
222 高解像度システムタイマ
224 アプリケーションレイヤアルゴリズム
226 非同期サンプルレートコンバータ(ASRC)
300 モバイル通信デバイス
302 アプリケーションプロセッサ
304 ADSP
306 SoC
308 USBハードウェアコントローラ
310 USB周辺機器
312 USBインターフェース
314 UACドライバ
316 バッファ
318 高解像度システムタイマ
320 データ処理回路機構
324 アプリケーションレイヤアルゴリズム
326 ASRC
400 ヘッドセット
402 デジタルオーディオコンバータ(DAC)
404 UACデータレギュレータ(UACデータreg)、UAC reg
406 ハードウェアコントローラ
408 ケーブル
410 USBホスト
412 ホストハードウェアコントローラ
414 アプリケーション
418 UACデータレギュレータ
420 ドングル、中間デバイス
422 ケーブル
424 ヘッドセット
500 データパケット
502 帯域内ドリフト検出器、検出器
506 帯域外ドリフト検出器、検出器
510 マルチプレクサ
514 デバイスドリフト累算器
516 ローカルクロックドリフト検出器
518 ローカルクロックドリフト累算器
520 加算器
524 初期参照ハンドラ
528 ジッタ遅延要素
534 データ配信ハンドラ
536 出力
538 加算器

544 加算器
546 合成タイムスタンプ
548 データ
600 信号フロー図
700 マイクロフォン
704 USBデバイスドライバ
706 USBホストドライバ
710 バッファ
712 タイマトリガ
716 ASRC
800 マイクロフォン
802 USBデバイスドライバ
808 USBホストドライバ
812 バッファ
814 タイマトリガ
818 ASRC
900 マイクロフォン
902 スピーカー
904 USBデバイスドライバ
906 USBホストドライバ
918 オーディオモジュール
920 バッファ
1000 マイクロフォン
1002 スピーカー
1006 USBデバイスドライバ
1008 USBホストドライバ
1020 オーディオモジュール
1022 バッファ
1100 プロセッサベースシステム
1102 中央処理ユニット(CPU)
1104 プロセッサ
1106 キャッシュメモリ
1108 システムバス
1110 メモリコントローラ
1112 メモリシステム
1114 入力デバイス
1116 出力デバイス
1118 ネットワークインターフェースデバイス
1120 ディスプレイコントローラ
1122 ネットワーク
1124 メモリユニット
1126 ディスプレイ
1128 ビデオプロセッサ

【図面】

【図１】

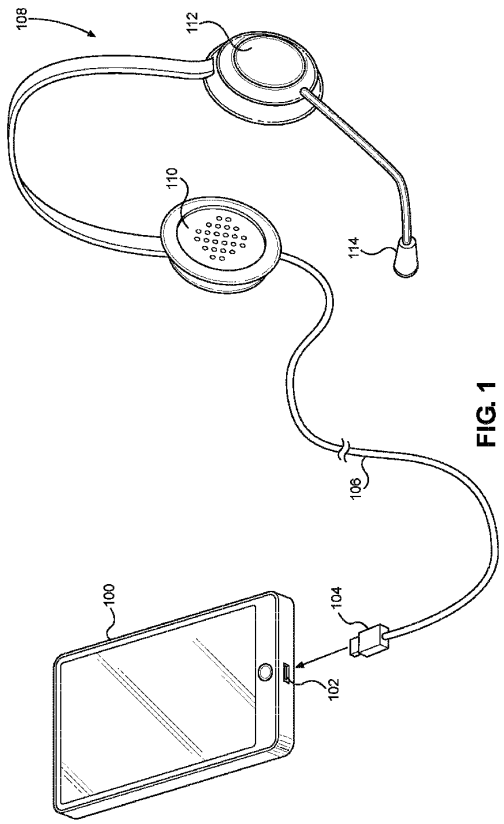
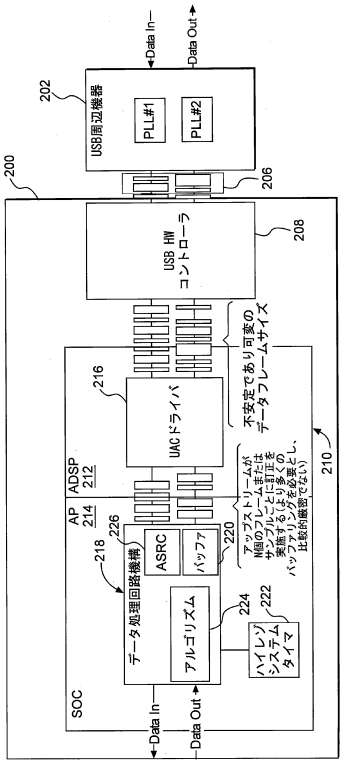
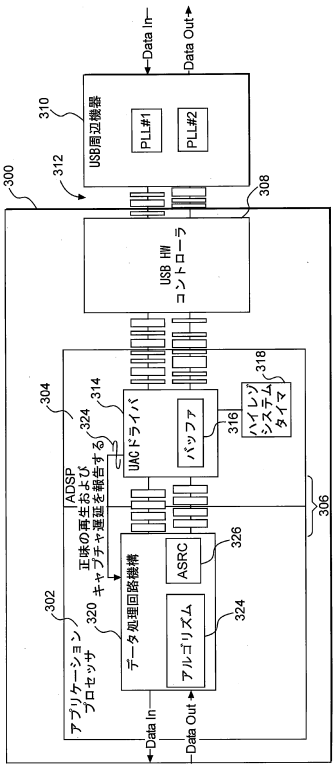


FIG. 1

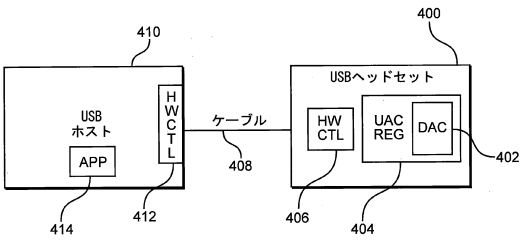
【図２】



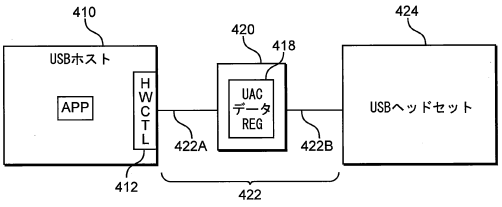
【図３】



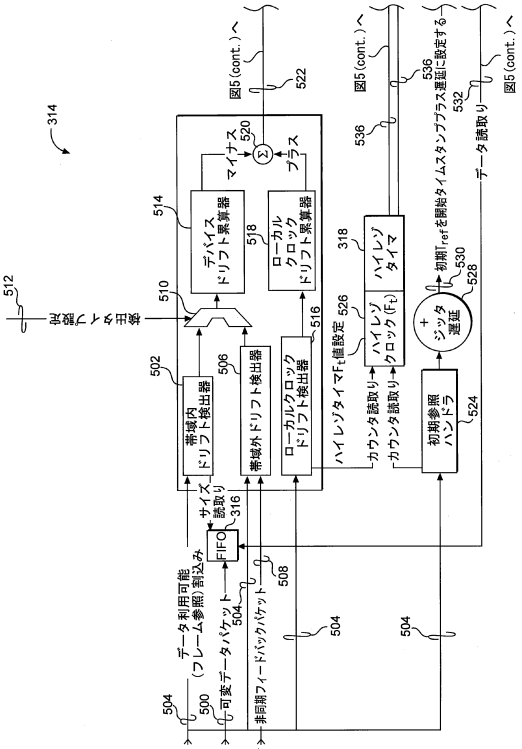
【図４Ａ】



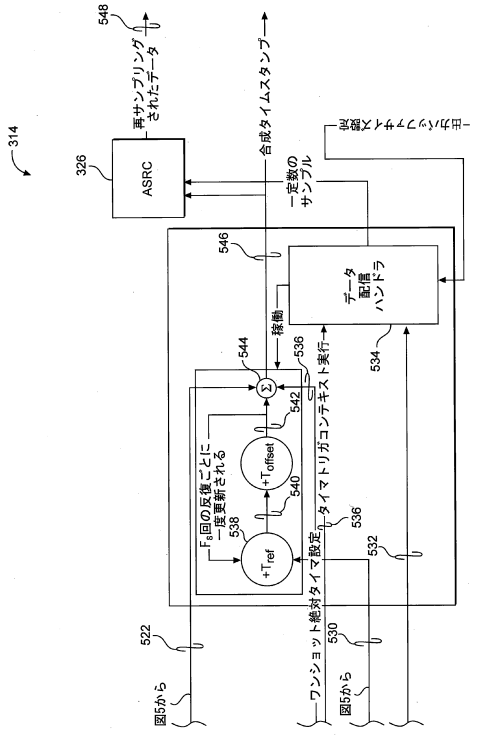
【図４Ｂ】



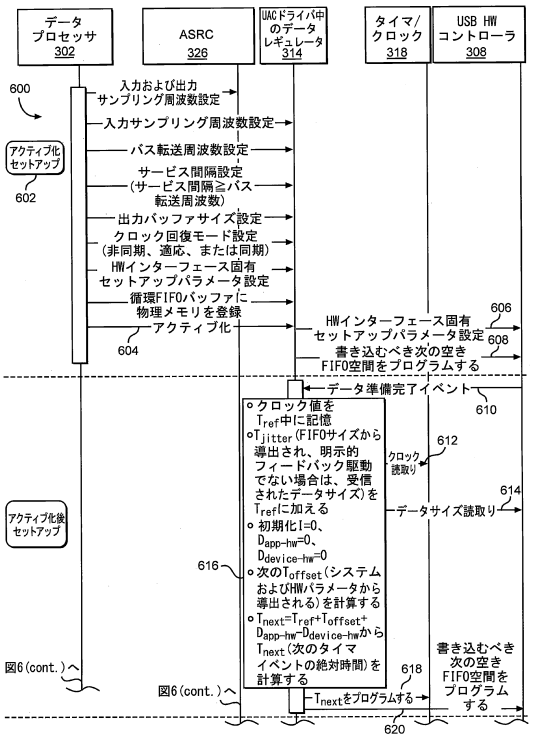
【図５】



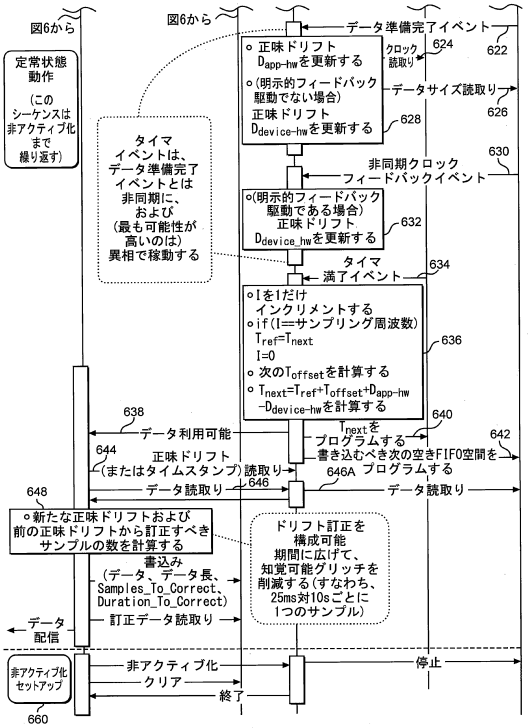
【図５（Ｃｏｎｔ．）】



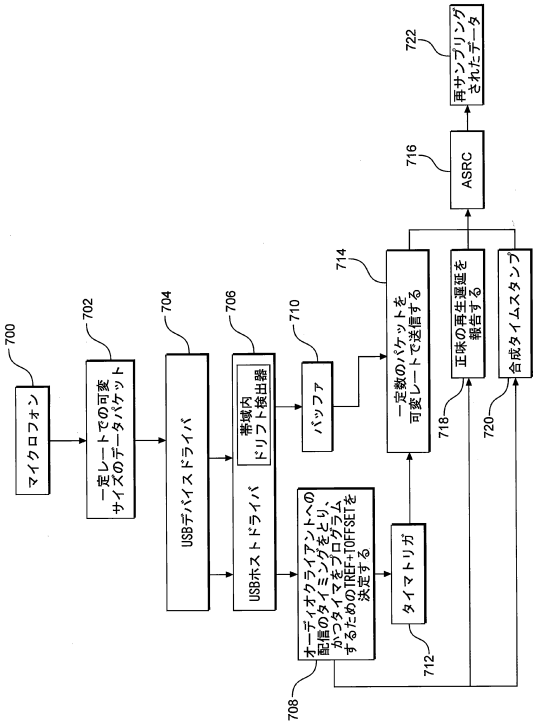
【図６】



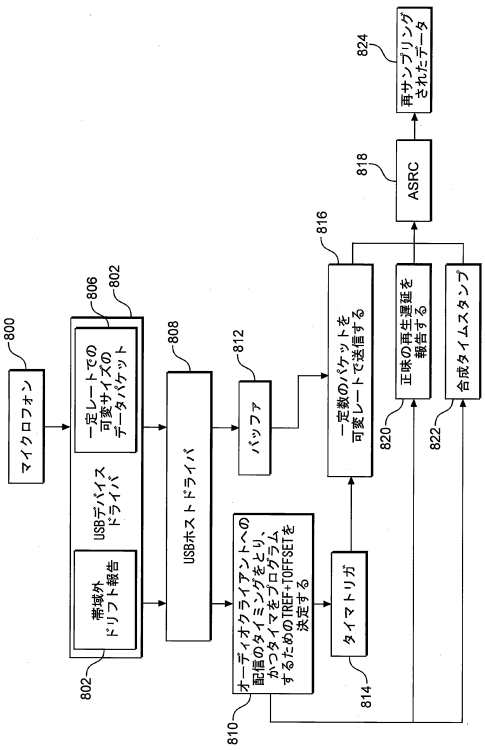
【図６（Ｃｏｎｔ．）】



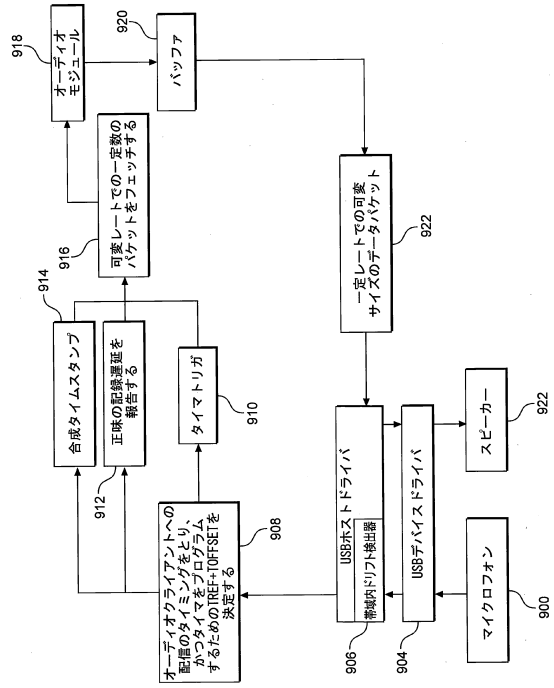
【図７】



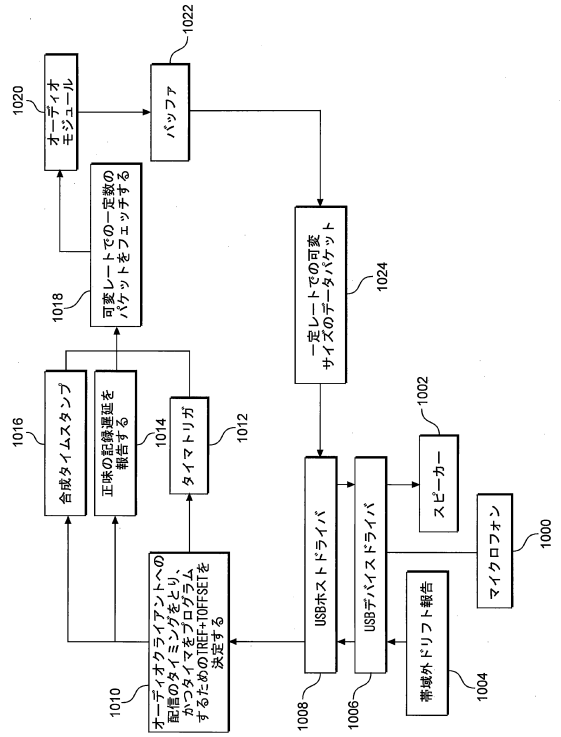
【図８】



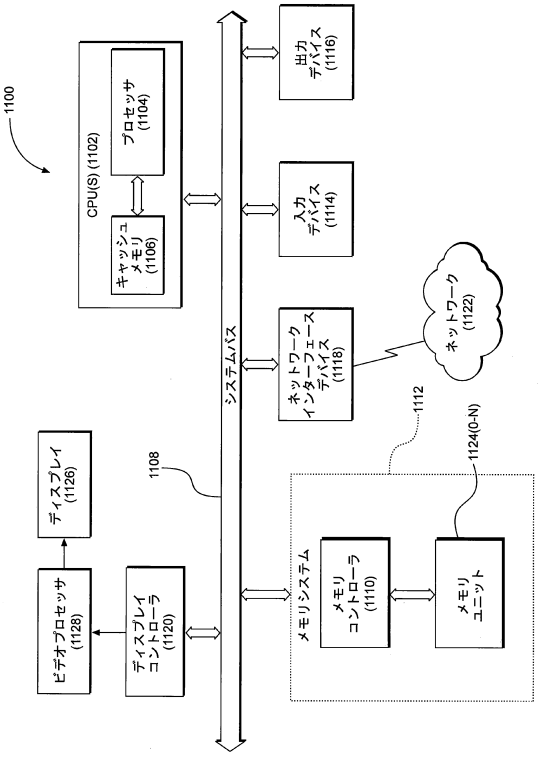
【図９】



【図１０】


オーディオモジュール
1020
バッファ
1022
可変レートでの一定数のパケットをフェッチする
1018
一定レートでの可変サイズのデータパケット
1024
合成タイムスタンプ
1016
正味の記録遅延を報告する
1014
タイマトリガ
1012
スピーカー
1002
マイクロフォン
1000
オーディオクライアントへの配信のタイミングをとり、かつタイマをプログラムするためのTREF+TOFFSETを決定する
1010
USBホストドライバ
1008
USBデバイスドライバ
1006
帯域外ドリフト報告
1004

【図１１】


1100
CPU(S) (1102)
プロセッサ (1104)
キャッシュメモリ (1106)
出力デバイス (1116)
入力デバイス (1114)
ネットワークインターフェースデバイス (1118)
ネットワーク (1122)
システムバス
1108
1112
1124(0-N)
メモリシステム
メモリコントローラ (1110)
メモリユニット
ディスプレイ (1126)
ビデオプロセッサ (1128)
ディスプレイコントローラ (1120)

フロントページの続き

(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　15/631,807
(32)優先日　平成29年6月23日(2017.6.23)
(33)優先権主張国・地域又は機関
米国(US)
１２１・サン・ディエゴ・モアハウス・ドライヴ・５７７５
(72)発明者　アメヤ・クルカルニ
アメリカ合衆国・カリフォルニア・９２１２１・サン・ディエゴ・モアハウス・ドライヴ・５７７５
審査官　田名網 忠雄
(56)参考文献　特開２０１４－０９６１６２（ＪＰ，Ａ）
特開２００２－１５２３０４（ＪＰ，Ａ）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１３／３８－１３／４２
Ｇ０６Ｆ　１３／１０－１３／１４
Ｈ０４Ｌ　４７／４３
Ｈ０４Ｌ　４９／９０５７