

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4376973号
(P4376973)

(45) 発行日 平成21年12月2日(2009.12.2)

(24) 登録日 平成21年9月18日(2009.9.18)

(51) Int.Cl.

H04B 14/04 (2006.01)
H04L 7/00 (2006.01)

F 1

H04B 14/04
H04L 7/00B
A

請求項の数 10 (全 12 頁)

(21) 出願番号 特願平10-537636
 (86) (22) 出願日 平成10年1月16日(1998.1.16)
 (65) 公表番号 特表2001-513295(P2001-513295A)
 (43) 公表日 平成13年8月28日(2001.8.28)
 (86) 国際出願番号 PCT/US1998/000832
 (87) 国際公開番号 WO1998/038766
 (87) 国際公開日 平成10年9月3日(1998.9.3)
 審査請求日 平成16年12月21日(2004.12.21)
 (31) 優先権主張番号 08/806,853
 (32) 優先日 平成9年2月26日(1997.2.26)
 (33) 優先権主張国 米国(US)

(73) 特許権者
 エリクソン インコーポレイテッド
 アメリカ合衆国 テキサス州 75024
 , ブラノ, レガシー ドライブ 63
 00
 (74) 代理人
 弁理士 大塚 康徳
 (74) 代理人
 弁理士 高柳 司郎
 (74) 代理人
 弁理士 大塚 康弘
 (74) 代理人
 弁理士 木村 秀二
 (74) 代理人
 弁理士 下山 治

最終頁に続く

(54) 【発明の名称】サンプルレートコンバータ

(57) 【特許請求の範囲】

【請求項 1】

第1のサンプルレートの入力データストリームを、第2のサンプルレートの出力データストリームに変換するサンプルレート変換の方法であって、

a) 前記第1のサンプルレートの複数の入力サンプルを含む入力データストリームを受信するステップと、

b) 前記入力サンプルの補間により、前記第2のサンプルレートの複数の出力サンプルを含む出力データストリームを生成するステップと、

を有し、前記出力データストリームを生成するステップにおいては、

i) 保持手段に保持された整数値 $a_{c,c}$ を0に設定し、

ii) 出力サンプルの計算の前に、前記 $a_{c,c}$ を予め定められた整数値 A だけ増加させ、

iii) 出力サンプルの計算の後に、前記 $a_{c,c}$ を予め定められた整数値 B だけ減少させ、

iv) 前記 $a_{c,c}$ の増加後と前記 $a_{c,c}$ の減少後とにおいて、出力サンプルを計算するか次の入力サンプルを選択するかを、前記 $a_{c,c}$ が0未満の場合は次の入力サンプルを選択し、前記 $a_{c,c}$ が0以上の場合は出力サンプルを計算すると決定し、

v) 前記 $a_{c,c}$, 前記 A , 前記 1 以上の入力サンプルの値に基づいて、前記出力サンプルを計算し、

前記整数値 A 、 B は、前記第1のサンプルレートの逆数 T_x 、前記第2のサンプルレートの逆数 T_y について、比率 A / B が T_x / T_y に等しい関係にある

ことを特徴とする方法。

10

20

【請求項 2】

前記入力サンプルの補間は線形補間であることを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記出力データストリームを生成するステップの v)においては、m 番目の出力サンプル値 Y_m を、式

$$Y_m = (\underline{a} \underline{c} \underline{c})^* X_n + (A - \underline{a} \underline{c} \underline{c})^* X_{n+1}) / A$$

により計算し、ここで、 X_n は現在の入力サンプルの値であり、 X_{n+1} は X_n の後に続く次の入力サンプルの値である

ことを特徴とする請求項 2 に記載の方法。

【請求項 4】

前記 A は 2 の累乗であることを特徴とする請求項 1 から 3 のいずれか 1 項に記載の方法。

【請求項 5】

第 1 のサンプルレートの入力データストリームを、第 2 のサンプルレートの出力データストリームに変換するサンプルレート変換の装置であって、

a) 前記第 1 のサンプルレートの複数の入力サンプルを含む入力データストリームを受信する手段と、

b) 前記入力サンプルの補間ににより、前記第 2 のサンプルレートの複数の出力サンプルを含む出力データストリームを生成する手段と、

を備え、前記出力データストリームを生成する手段は、

i) 保持手段に保持された整数値 $a \underline{c} \underline{c}$ を 0 に設定し、

ii) 出力サンプルの計算の前に、前記 $a \underline{c} \underline{c}$ を予め定められた整数値 A だけ増加させ、

iii) 出力サンプルの計算の後に、前記 $a \underline{c} \underline{c}$ を予め定められた整数値 B だけ減少させ、

iv) 前記 $a \underline{c} \underline{c}$ の増加後と前記 $a \underline{c} \underline{c}$ の減少後とにおいて、出力サンプルを計算するか次の入力サンプルを選択するかを、前記 $a \underline{c} \underline{c}$ が 0 未満の場合は次の入力サンプルを選択し、前記 $a \underline{c} \underline{c}$ が 0 以上の場合は出力サンプルを計算すると決定し、

v) 前記 $a \underline{c} \underline{c}$, 前記 A , 前記 1 以上の入力サンプルの値に基づいて、前記出力サンプルを計算し、

前記整数値 A 、 B は、前記第 1 のサンプルレートの逆数 T_x 、前記第 2 のサンプルレートの逆数 T_y について、比率 A / B が T_x / T_y に等しい関係にある

ことを特徴とする装置。

【請求項 6】

前記入力サンプルの補間は線形補間であることを特徴とする請求項 5 に記載の装置。

【請求項 7】

前記出力データストリームを生成する手段が v)において前記出力サンプルを計算する際は、m 番目の出力サンプル値 Y_m を、式

$$Y_m = (\underline{a} \underline{c} \underline{c})^* X_n + (A - \underline{a} \underline{c} \underline{c})^* X_{n+1}) / A$$

により計算し、ここで、 X_n は現在の入力サンプルの値であり、 X_{n+1} は X_n の後に続く次の入力サンプルの値である

ことを特徴とする請求項 6 に記載の装置。

【請求項 8】

前記 A は 2 の累乗であり、当該 A による除算を行うためのビットシフタを更に備えることを特徴とする請求項 5 ~ 7 のいずれか 1 項に記載の装置。

【請求項 9】

第 1 のサンプルレートの入力データストリームを、当該第 1 のサンプルレートよりも小さな第 2 のサンプルレートの出力データストリームに変換するサンプルレート変換の方法であり、

a) 前記第 1 のサンプルレートの複数の入力サンプルを含む入力データストリームを受信するステップと、

b) 前記入力サンプルの補間ににより、前記第 2 のサンプルレートの複数の出力サンプルを含む出力データストリームを生成するステップと、

10

20

30

40

50

を有し、前記出力データストリームを生成するステップにおいては、

- i) 保持手段に保持された整数値 a_{ccc} を 0 に設定し、
- ii) 出力サンプルの計算の前に、前記 a_{ccc} を予め定められた整数値 A だけ増加させ、
- iii) 出力サンプルの計算の後に、前記 a_{ccc} を予め定められた整数値 B だけ減少させ、
- iv) 前記 a_{ccc} の増加後に、出力サンプルを計算するか次の入力サンプルを選択するかを、前記 a_{ccc} が 0 未満の場合は次の入力サンプルを選択し、前記 a_{ccc} が 0 以上の場合は出力サンプルを計算すると決定し、
- v) 前記 a_{ccc} , 前記 A , 前記 1 以上の入力サンプルの値に基づいて、前記出力サンプルを計算し、

前記整数値 A 、 B は、前記第 1 のサンプルレートの逆数 T_x 、前記第 2 のサンプルレートの逆数 T_y について、比率 A / B が T_x / T_y に等しい関係にあることを特徴とする方法。

【請求項 10】

第 1 のサンプルレートの入力データストリームを、当該第 1 のサンプルレートよりも大きな第 2 のサンプルレートの出力データストリームに変換するサンプルレート変換の方法であって、

- a) 前記第 1 のサンプルレートの複数の入力サンプルを含む入力データストリームを受信するステップと、
- b) 前記入力サンプルの補間ににより、前記第 2 のサンプルレートの複数の出力サンプルを含む出力データストリームを生成するステップと、

を有し、前記出力データストリームを生成するステップにおいては、

- i) 保持手段に保持された整数値 a_{ccc} を 0 に設定し、
- ii) 出力サンプルの計算の前に、前記 a_{ccc} を予め定められた整数値 A だけ増加させ、
- iii) 出力サンプルの計算の後に、前記 a_{ccc} を予め定められた整数値 B だけ減少させ、
- iv) 前記 a_{ccc} の減少後に、出力サンプルを計算するか次の入力サンプルを選択するかを、前記 a_{ccc} が 0 未満の場合は次の入力サンプルを選択し、前記 a_{ccc} が 0 以上の場合は出力サンプルを計算すると決定し、

前記整数値 A 、 B は、前記第 1 のサンプルレートの逆数 T_x 、前記第 2 のサンプルレートの逆数 T_y について、比率 A / B が T_x / T_y に等しい関係にあることを特徴とする方法。

【発明の詳細な説明】

発明の分野

この発明はデジタル信号のフィールドサンプルレート変換に関する。より詳しくは、この発明は、入力サンプルと出力サンプルのタイミング関係を計算するために整数アキュムレータを用いるサンプルレート変換方法に関する。

発明の背景

多くの電子的アプリケーションにおいて、信号はデジタル的に表現され処理される。デジタル語、すなわちサンプルは、規則的な時間間隔で、データの値を表現する。この規則的な間隔は、しばしばサンプルレートと呼ばれ、サンプル間隔期間の逆数を表すキロヘルツ (k Hz) で典型的に表現される。

データの利用可能なサンプルレートが、希望するサンプルレートと相違するときの、種々な状況がある。サンプルデータの特性と、利用可能データと希望データの相違の大きさに応じて、あるアンプルレートから他のサンプルレートへ、信号の意味を故意に変更せずに変換するために、いくつかの方法を使用できる。

サンプルレート削減の第1の一般的な技法は、デシメーションと呼ばれる。それは、利用可能なサンプルレートが、希望のサンプルレートの整数倍であるときの選択法である。デシメータが、整数 d により入力サンプルを削減して、出力サンプルレートを生成する。入力信号に高周波成分が含まれていれば、ローパスフィルタで最初にこれを除去して、エイ

10

20

30

40

50

リアシング効果を防止しなければならない。このレート削減は、d番目の出力サンプル毎に、d-1の入力を廃棄するだけで遂行される。

デシメーションの主な欠点は、レート削減が整数に限られることである。その上、アンチエイリアシングフィルタが必要なときは、これが非常に計算集中的なものになるかも知れない。

サンプルレート変換の第2の一般的な方法は、補間として知られている。補間は、整数倍によりサンプルレートを増大するために、しばしば使用される方法である。サンプルレートをIまで増大するために、基本補間器が、ゼロの値を有するI-1のサンプルを、全ての入力サンプルの間に挿入する。この結果のサンプルは、それから、より高いレートにおいて、アンチエイリアシングローパスフィルタを通じて、フィルタされる。

補間法は、デシメーションと同じ基本的な欠点のいくつかを有する。レート増大は整数に限定され、また、補間に必要なローパスフィルタリングは高価である。

補間とデシメーションは、それ自体では、整数のサンプルレートでの変更ができるだけである。多くの場合、これが必要なフレキシビリティを与えない。恐らくはいくつかの段階で補間とデシメーションを組み合わせることにより、サンプルレート変換を微調整することが可能になる。例えば、入力サンプルレートが194.4 kHzで、希望のサンプルレートが153.6 kHzであれば、整数因数による相違ではない。その代わりに、比率64/81により、希望のレートを利用可能なレートに関係付けることができる。希望のサンプルレートを達成するために、最初にデータを64の因数で補間して、それから81の因数でデシメートする。しかしながら、大きな補間とデシメーションの因数は、必要なローパスフィルタに非常に厳しい制約を課す。フィルタの要件を小さくするために、補間とデシメーションを、いくつかの段階で遂行する。すなわち、8で補間し、続いて9でデシメーション、続いて8で補間、続いて9でデシメーションすると、合計で64/81変換になる。補間とデシメーションの組み合わせを使用すれば、整数に限定されない一層広い範囲のレート変換が可能になる。しかしながら、そうしたアプローチの大きな欠点は、フィルタリングのコストである。

入力レートと出力レートが近接しているときに有用なもう一つの技法は、線形補間として知られている。この方法は1次線形補間のアプローチを用いて、2つの入力サンプルおよび入力サンプルと出力サンプルの時間的相対位置の関数としての各出力サンプルを推定する。図1を参照すると、m番目の出力は次式により計算される。

$$y_m = x_n * k + x_{n+1} * (1-k)$$

ここで $k = ((n+1) * T_x - m * T_y) / T_x$

これらの式において、 x_n はn番目の入力サンプルであり、 y_m はm番目の出力サンプル、 T_x は入力サンプル期間、 T_y は出力サンプル期間である。 T_x を入力サンプル期間とすると、n番目の入力サンプルの時間は、

$$tx_n = n * T_x$$

である。同様にm番目の出力サンプルの時間は、

$$ty_m = m * T_y$$

である。出力サンプル番号mを計算するには、

$$n * T_x \leq m * T_y < (n+1) * T_x$$

になるように、nを選択する。

1次補間法の1つの欠点は、ある出力サンプルの計算に使用するためにどの特定入力サンプルを使用すべきかを知らせるのが難しいことである。 T_x と T_y は整数であることが稀であり、丸め誤差が間違ったサンプルを選択する原因になり得る。mとnが大きくなると非常に大きくなる2つの数の引き算を含むkの計算もまた、数の正確さの限界による誤差の影響を受けやすくなる。

これらの欠点は、kの値を有するテーブルを使用することにより、緩和できる。入力レートが194.4 kHzで出力レートが153.6 kHzの例により続けると、サンプルの時間上の相対位置は、81個の入力サンプルに渡って反復する。こうして、テーブルの中

10

20

30

40

50

の k の値を事前に計算して記憶できる。テーブルを使用することにより、事前に計算された値を記憶するための費用によって、全てのサンプルについて k を計算する必要をなくすことができる。事前に計算されたテーブルを使用することの欠点は、テーブルの値を事前に計算して記憶しなければならず、これが追加のハードウェア資源を必要とすることである。同一のコンバータを、種々の入力サンプルレートと出力サンプルレートの組み合わせで使用するときに、このことは特に問題になる。

発明の要約

この発明は、先行技術の欠点のいくつかを克服するサンプルレート変換の方法と装置である。この発明の方法は、テーブルの事前の計算と記憶を必要とせずに、テーブルに基づくアプローチの堅牢性を提供する補間法である。この方法はまた、包含する計算を単純化し、また丸め誤差への感度を低くする。この方法は整数アキュムレータを使用して、他のサンプルレートでの複数の入力サンプルを含む入力データに基づく1つのサンプルデータでの複数の出力データを含む1つの出力データストリームを生成する。一層詳しくは、この方法は整数アキュムレータを使用して、入力サンプルと出力サンプルの間のタイミング関係を追跡する。アキュムレータの値に基づいて、この方法は、現在の出力サンプルを計算するために正しい入力サンプルが使用されているかどうかを判定する。もしそうならば、入力サンプルとアキュムレータ値の関数として、出力サンプルが計算される。アキュムレータ値を維持するために単純な整数演算を使用することにより、さもなければ必要とされる、非テーブルに基づくアプローチの柔軟性とコンパクトさを維持しながら正しい入力が使用されていることの確認のための、不必要な大量の計算を、この発明は避けることができる。

この発明の装置は、入力サンプルレジスタ、整数アキュムレータ、好ましくはビットシフトレジスタを含み、これら全ては、この発明のサンプルレート変換方法を実行するように構成される。

【図面の簡単な説明】

図1は、線形補間を描写する図面である。

図2は、この発明のサンプルレート変換法の論理的フローチャートである。

図3は、入力サンプル($\dots x_{n-3}, x_{n-2}, x_{n-1}, x_n, x_{n+1}, \dots$)と出力サンプル($\dots y_{m-2}, y_{m-1}, y_m, y_{m+1}, \dots$)の間のタイミング関係を描写する図面である。

図4は、AがBよりも小さいことが知られているときに適用できる単純化されたサンプルレート変換法の論理的フローチャートである。

図5は、AがBよりも大きいことが知られているときに適用できる単純化されたサンプルレート変換法の論理的フローチャートである。

図6は、サンプルレートコンバータのための装置の好ましい実施例のブロック図である。

図7は、図4の方法を使用する図6の装置のためのコントローラの好ましい実施例の論理的フローチャートである。

発明の詳細な説明

好ましい実施例を描写する図面を参照しながら、以下にこの発明を一層詳細に説明する。しかしながら、この発明は多くの異なった実施例を採用することができるものであり、本書に説明される実施例にこの発明を限定することを意図しない。

この発明の方法は補間の変形であり、入力サンプルと出力サンプルの時間的位置関係の計算を容易にするために整数アキュムレータ220を使用するものである。線形補間のために、この方法は2つの正の整数の定数AおよびBを使用して、2つの入力サンプルと1つの出力サンプルのタイミング関係を計算する。入力信号のサンプル期間が T_x であって出力信号のサンプル期間が T_y であれば、そのときはAとBが下記の方程式を満足するよう $A/B=T_x/T_y$ に、AとBが選択される。

$$A/B=T_x/T_y$$

AとBの実際の値またはサイズは、望ましい正確さのレベルとその他の考慮点により異なる。実際、入力サンプル期間 T_x は、A個のステップに量子化される。これらのステップを小さく保つ(Aを大きく保つ)ことにより、付加される量子化雑音を小さく保つことが

10

20

30

40

50

できる。しかしながら、Aを表現するために多数のビットを使用することは、ハードウェア資源を浪費するかもしれない。例えば、入力信号を194.4 kHzのサンプルレートで利用でき、希望する出力レートが153.6 kHzであるとすると、

$$A/B = T_x/T_y = ((1/194,400)/(1/153,600)) = 153,600/194,400 = 64/81$$

こうして、入力サンプル期間を64個のステップ(A=64)へ量子化したり、または64の倍数のステップにしても、AとBが整数でありながらこの固有の比率を維持できる。こうして、Bが1296であれば、Aは1024であり得る。しかしながら、Aが64の場合は、この値を表現するのに6ビットだけで済む($2^6 = 64$)が、Aが1024の場合は10ビットが必要である($2^{10} = 1024$)。いずれの場合も、1つの付加ビットが必要でありえることに注意されたい。

1組の入力サンプルと所与の出力サンプルについて、時間的相対関係を追跡する変数「acc」の値を反復的に計算するために、AおよびBの値が使用される。単純な項において、accは2つの関数の働きをする。第1に、カレントの出力サンプルを計算するために入力サンプルの正しいペアが使用されているかどうかを決定するために、accが使用される。第2に、出力サンプル値が正しく推定されるように、入力サンプルペアの各メンバへ適当な重みを割り当てるために、accが使用される。これらの機能をaccがどのように遂行するかの詳細は、以下の説明により明らかになるであろう。

図2は、この発明の論理的なフローチャートである。処理の開始において、整数アキュムレータ220がゼロにセットされ、変数mと変数nもゼロにセットされる(ボックス10)。変数mはカレント出力サンプルのシーケンス番号を表現する整数カウンタである。変数nはカレント入力サンプルのシーケンス番号を表現する整数カウンタである。下記の説明から明らかになるが、フローチャート内のmとnの唯一の目的は、入力サンプルと出力サンプルがどのように関連するかを明らかにすることにより、読者がアルゴリズムを理解するのに役立つことである。この発明を実行するのに、mもnも、記憶や計算する必要がない。

アキュムレータ220は、変数「acc」を表現する整数値を含んでいる。主要処理ループは、アキュムレータ220内の整数値にAを加えることにより開始する(ボックス20参照)。それからaccがゼロよりも小さいかどうかをチェックする(ボックス30)。もしそうならば、nを1つだけ増して(ボックス40)、次の入力サンプルを選択することにより、カレントの入力サンプルの役割を仮定して、それはボックス20へ帰る。accがゼロよりも小さくなければ、出力サンプルを計算する(ボックス50)。単純な項において、ボックス30のacc値チェックは、計算される特定の出力サンプル値が、現在のサンプル期間、すなわち2つの現在選択されている入力値(nとn+1)の間、またはちょうど入力サンプル x_{n+1} にあるかどうかの決定を表現する。

この発明のために、現在の入力サンプル期間は、現在の入力サンプルから次に続く入力サンプルまでの期間と定義される。こうして、もし入力サンプルレートが1Hzであり、現在の入力サンプルが(数列0, 1, 2, 3, 4, ..., nの)4番目であれば、現在の入力サンプル期間は4秒から5秒までである。

出力サンプルの計算(ボックス50)は、下記の式により遂行される。

$$y_m = (acc * x_n + (A - acc) * x_{n+1}) / A$$

この式は修正された線形補間式である。この式において、 x_n は出力サンプルの前に起きる最後の入力サンプルであり、 x_{n+1} は出力サンプルの後の最初の入力サンプルの値である。入力サンプルの上に直接に出力サンプルがあたる場合は、accが整数ゼロに等しく、式は短くまとめられて、

$$y_m = (A * x_{n+1}) / A = x_{n+1}$$

になる。こうして、入力サンプルに正しく対応する値 x_{n+1} が出力サンプル(y_m)のために使用される。しかしながら、直接オーバーラップの一番最初の場合の特例として、最初の入力サンプル(x_0)と最初の出力サンプル(y_0)において、accはAに等しく、従ってこの式はこの場合短くまとめられて、

10

20

30

40

50

$$y_0 = (A * x_0) / A = x_0$$

になる。

出力サンプルが計算された（ボックス 50）後で、acc の値が B（ボックス 60）でデクリメントされる。ボックス 70 で、acc のこの新しい値がチェックされて、それがゼロよりも大きいかまたは等しいかがチェックされる。もしそうならば、m を 1 つだけインクリメントして、処理がボックス 50 へ戻る。もしそうでないならば、そのときは m と n を 1 フインクリメントする（ボックス 90）。簡単な項では、ボックス 70 の acc 値チェックは、次の出力サンプルが入力サンプルの同一ペア内に起こるかどうかの決定を表現する。

主処理ループ（ボックス 20 ボックス 100）を通じて、サンプルがなくなるまで（ボックス 100）処理が継続され、なくなった時点で停止する（ボックス 110）。この仕方で期間 T_x の入力サンプルレートが期間 T_y の出力サンプルレートに変換される。 10

作動中のこの方法の一例として、図 3 を参照されたい。入力サンプルレートが出力サンプルレートよりも速いと仮定すると、A が B よりも小さいことを意味する。議論の便宜のために、入力サンプルレート 1.4 kHz と出力サンプルレート 1.0 kHz に対応して、A が 10 で B が 14 であると仮定する。更に、この変換処理はこの発明により進行中であり、今やサンプル x_{n-3} とサンプル y_{m-2} を処理中であると仮定する。この時点で、ボックス 20 に入り、acc は -4 に等しい。ボックス 20 で、acc が今や $6 (-4 + 10)$ になるように、acc を A でインクリメントする。6 は 0 よりも大きいので（ボックス 30）、 x_{n-3} と x_{n-2} に基づいて、出力サンプル y_{m-2} が計算される。さて、acc が -8 ($+6 - 14$) に等しくなるように、14 だけデクリメントされる（ボックス 60）。主処理ループのこの第 2 パスの間に、 x_{n-2} と x_{n-1} を使用して y_{m-1} の値が計算されて、acc が $-12 (-8 + 10 - 14)$ になるように調節される。主処理ループの第 3 パスにおいて、acc が $-2 (-12 + 10)$ になるように、A によりインクリメントされる。今や、acc が 0 よりも小さく、現在の入力サンプル（この時点で x_{n-1} ）が除去されるので、次の入力サンプル x_n が現在の入力サンプルの位置を取り、acc は $8 (-2 + 10)$ へ増加される。それから x_n と x_{n+1} を使用して出力サンプル y_m が計算される。主処理ループの第 3 パスの結果（ボックス 100）において、acc は $-6 (8 - 14)$ に等しい。図 3 に示すように、出力サンプル y_m を計算するのに、入力サンプル x_{n-1} と x_n ではなく入力サンプル x_n と x_{n+1} が y_m が使用される理由は、 y_m が x_n と x_{n+1} の間にあるためである。 20

この説明から理解できるように、変数 acc は、入力サンプルと出力サンプルの間のタイミング関係をダイナミックに追跡するために使用される。B よりも A が小さいこの例において、ボックス 30 において acc がゼロよりも小さいとき、この入力サンプルは、「特別な」1つまたはそれ以上の位置を進められる。他の状況において、A よりも B が小さいときは、ボックス 70 において、acc がゼロに等しいかまたはより大きいとき、同一の入力サンプルペアを使用して、2つまたはそれ以上の出力サンプルを計算し得る。こうして、正しい入力サンプルペアを使用して各所与の出力サンプルを計算されていることを検査する処理によって、変数 acc が使用されることが理解できる。 30

もし定数 A が定数 B よりも大きい、またはその反対であることが知られていれば、図 2 のアルゴリズムを少し簡単にできる。図 4 は、A が B よりも小さいことが知られているときのための簡単化されたフローチャートを示し、同一の参照番号を使用している。図 5 は、A が B より大きいことが知られているときのための簡単化された論理的なフローチャートである。図 4 と図 5 のフローチャートは、A と B の間の関係が知られているときは、比較とループバックのステップを省略でき、これにより処理を単純化できることを示す。 40

上記の処理のために、出力サンプル（ y_m ）は定数 A による除算を必要とする。除算をハードウェアにインプリメンテーションするのはやや高価であるので、代わりに $1 / A$ を予め計算しておいて、乗算を使用することができます。より好ましくは、代わりに、容易に割り算できる値を A のために選択しても良い。例えば、A が 2 の累乗であれば、簡単なバイナリビットシフトとして除算をインプリメントできる。 50

例示の目的のために、入力サンプルが 194.4 kHz で利用でき、希望するサンプルレートが 153.6 kHz であるとする。これは下記を意味する。

$$A/B = T_x/T_y = 153,600/194,400 = 64/81$$

こうして、A は 64 に等しくなり得るし、B は 81 に等しくなり得る。そうならば、それから入力サンプル期間 T_x を 64 ステップに分割できる。例えば A を 1536 に増加し、これに応じて B を 1944 に増加すれば、より大きな正確さが得られる。しかしながら、好ましくは A を $1024 (2^{10})$ のような 2 の累乗にして、従って B を 1296 のようにする。A が 1024 であれば、除算は、10 回のバイナリ右シフトとしてインプリメントできる。

上記の議論は線形補間のアプローチの利用を仮定している。しかしながら、この発明の方法は、2 次または 3 次またはこの分野で知られた他の方法のような、他の補間アプローチをも利用できる。これら他の線形補間アプローチのいくつかは、所与の出力サンプルを計算するために、2つよりも多い入力サンプルを使用する必要がある。2つの入力サンプルだけが必要ならば、そのときは1つの整数アキュムレータ 220 を使用するだけでよい。

2つよりも多い入力サンプルが必要ならば、複数の整数アキュムレータ 220 を使用して、入力サンプルと出力サンプルの間の種々のタイミング関係を追跡できる。代わりに、1つのアキュムレータを使用して、必要な全ての入力サンプルと計算すべき出力サンプルの間の関係を追跡することができる。これは、1つの入力サンプルへのタイミング関係が知られれば、他の入力サンプルへのタイミング関係は、A の更に遠くへの整数インクリメントであるだけだからである。もし（線形補間以外の）異なった補間アプローチを使用すれば、各所与の出力サンプルを計算するために、異なった公式も使用されるであろうことは明らかである。しかしながら、出力サンプル値はそれでも、少なくとも複数の入力サンプルと 1つまたはそれ以上のアキュムレータ値の関数であろう。

サンプルレートコンバータ 200 の可能なハードウェアインプリメンテーションのプロック図を、図 6 に示す。このサンプルレートコンバータ 200 は、コントローラ 210、整数アキュムレータ 220、マルチプレクサ 230、加算器 240、同 250、減算器 260、乗算器 270、同 280、入力サンプルレジスタ 290、ビットシフタ 300 を含む。コントローラ 210 は、コンバータ 200 の全般的な機能を制御する。アキュムレータ 220 は、整数演算を使用して、入力サンプルと出力サンプルの時間的位置関係を追跡する。マルチプレクサ 230 は、A と B の値について、ソース 180、同 190 に接続されている。入力サンプルはレジスタ 290 へ逐次的に供給される。ビットシフタ 300 は、適当なビットシフトを遂行して、A による除算を反映し、また各出力サンプルについて出力サンプル値を出力する。

二者択一的に、コントローラ 210 の諸機能は、図 6 に示すように单一のデバイス内に集めるよりも、むしろコンバータ 200 の中で分散させることができる。また、加算器 240、同 250、減算器 260、乗算器 270、同 280、入力サンプルレジスタ 290 のようなコンバータ 200 の 2 つ以上の構成部品もまた、集積論理演算装置に結合できるが、これは一層高価になり得る。

図 7 は、図 4 に記述された方法のために、図 6 のコントローラ 210 の好ましいオペレーションの単純化されたフローチャートを示す。初期化すると、コントローラ 210 はアキュムレータ 220 にクリアするように命令し、また入力サンプルレジスタ 290 に第 1 入力サンプルをロードするように命令する（ボックス 310）。この動作が図 4 のボックス 10 に対応することに注意されたい。それからコントローラは、次の入力サンプルが利用可能であることを検査する（ボックス 320）。もしそうでなければ、コントローラ 210 は次の入力サンプルが利用可能になるまでループする。もしそうならば、そのときはコントローラ 210 はマルチプレクサ 230 に、A をロードするように命令し、またアキュムレータ 220 に A によりインクリメントさせる（ボックス 330）。この動作が図 4 のボックス 20 に対応することに注意されたい。それからコントローラ 210 はアキュムレータ 220（ボックス 340）の信号ビットをチェックする（ボックス 340）。この動作が図 4 のボックス 30 に対応することに注意されたい。符号ビットが肯定でなければ、

10

20

30

40

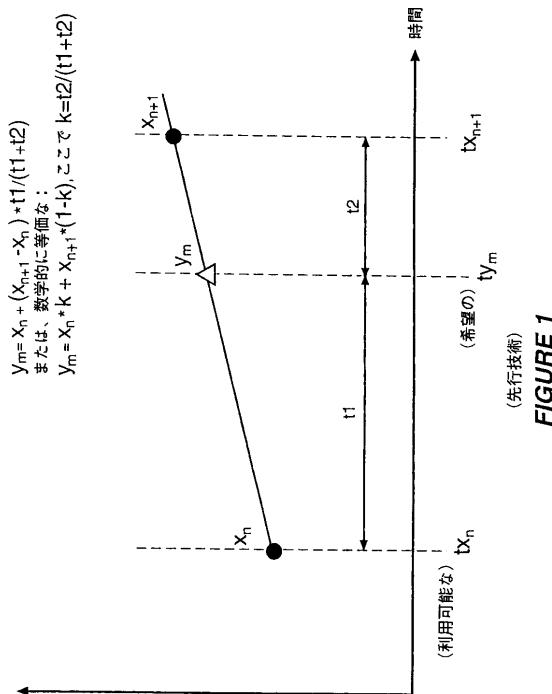
50

コントローラ 210 は、出力サンプル (y_m) を計算させ、マルチプレクサに -B へスイッチさせ、アキュムレータ 220 に、既存のアキュムレータ値に -B を加算させる (ボックス 350)。これらの動作は、図 4 のボックス 50、同 60、同 90 に対応する。ボックス 350 の後か、またはもしボックス 340 の符号ビットが否定であれば、コントローラ 210 は入力サンプルレジスタ 290 に、次の入力サンプルをロードさせる (ボックス 360)。

図 6 のコンバータ 200 は、A が 2 の累乗であるときの上記の線形補間ランプルレート変換の簡単なハードウェアインプリメンテーションである。コンバータ 200 は、 x_1, x_2, \dots, x_n の入力サンプルストリームのサンプルレートを、整数演算を使用して、異なる出力サンプルレートを有する y_1, y_2, \dots, y_m の出力ストリームへ変換することができる。A と B をプログラム可能な定数にすることにより、同一のサンプルレートコンバータ 200 を、いくつかの異なる入力対出力比率で動作するように、プログラムできる。

10

【図 1】



【図 2】

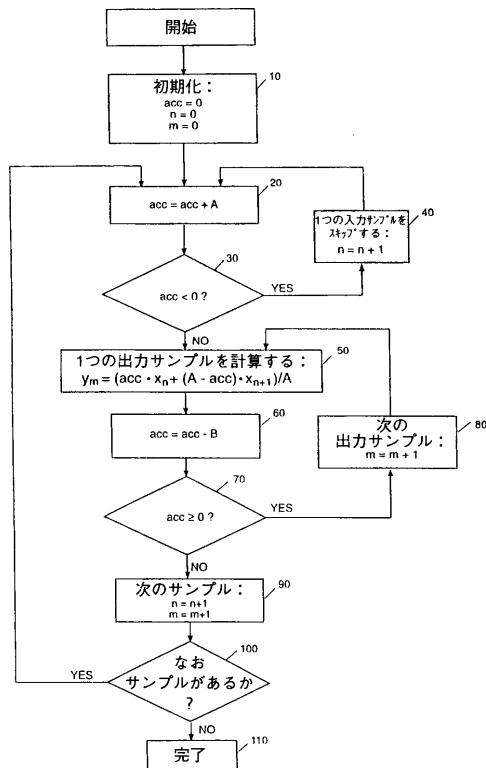


FIGURE 2

【図3】

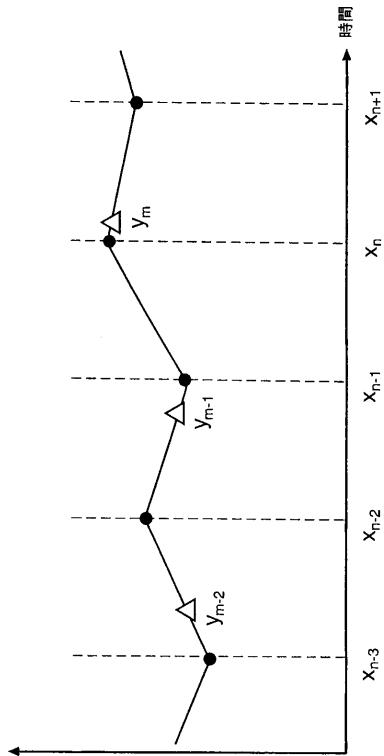


FIGURE 3

【 四 4 】

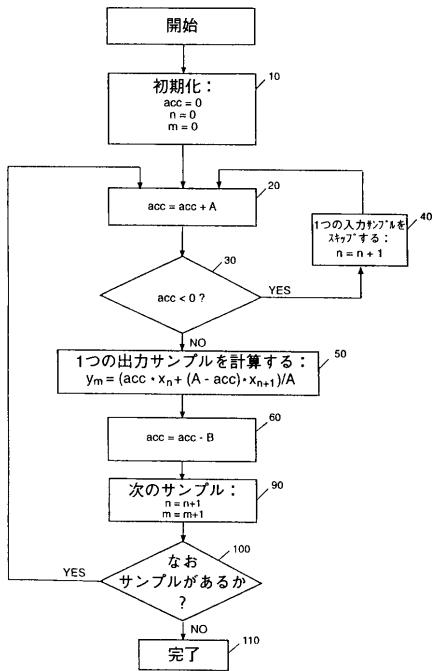


FIGURE 4

【図5】

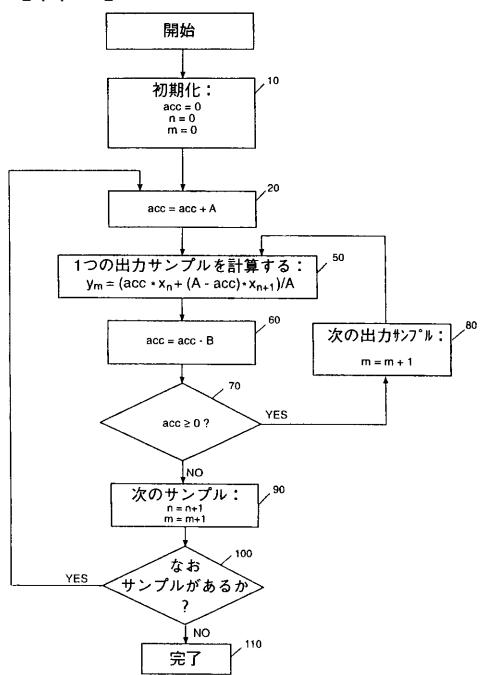


FIGURE 5

【 义 6 】

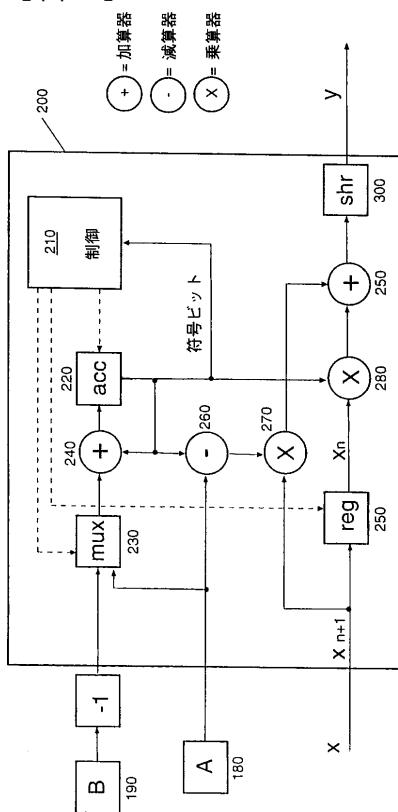


FIGURE 6

【図7】

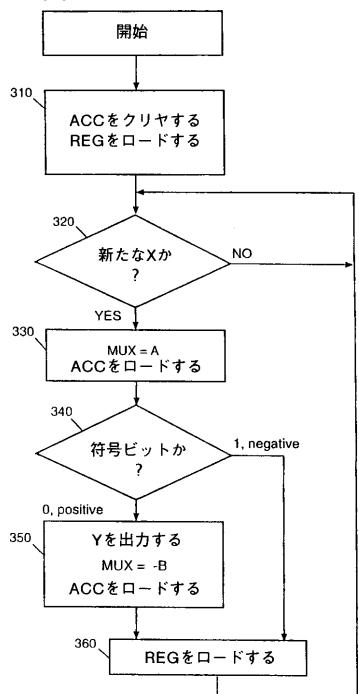


FIGURE 7

フロントページの続き

(74)代理人

弁理士 浅村 皓

(74)代理人

弁理士 浅村 肇

(74)代理人

弁理士 宇都宮 正明

(74)代理人

弁理士 林 銘三

(72)発明者 ホルムクビスト,ペーター,ボ

アメリカ合衆国,ノースカロライナ,モリスビル,ダック ポンド サークル 2207 シー

審査官 藤井 浩

(56)参考文献 特開平07-235861(JP, A)

特開平05-110520(JP, A)

特開昭61-204700(JP, A)

(58)調査した分野(Int.Cl., DB名)

H04B 14/04

H04L 7/00