(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6724828号
(P6724828)

(45)発行日　令和2年7月15日(2020.7.15)　(24)登録日　令和2年6月29日(2020.6.29)

(51)Int.Cl.		F I		
G10H 1/00	(2006.01)	G10H	1/00	C
G10K 15/04	(2006.01)	G10K	15/04	302Z
H03H 17/06	(2006.01)	H03H	17/06	633Z
H03H 17/08	(2006.01)	H03H	17/08	B

請求項の数 16　(全 32 頁)

(21)出願番号　特願2017-49459(P2017-49459)
(22)出願日　平成29年3月15日(2017.3.15)
(65)公開番号　特開2018-151589(P2018-151589A)
(43)公開日　平成30年9月27日(2018.9.27)
審査請求日　平成31年3月7日(2019.3.7)

(73)特許権者　000001443
カシオ計算機株式会社
東京都渋谷区本町１丁目６番２号
(74)代理人　100074099
弁理士　大菅　義之
(74)代理人　100121083
弁理士　青木　宏義
(74)代理人　100138391
弁理士　天田　昌行
(72)発明者　横田　益男
東京都羽村市栄町３丁目２番１号　カシオ計算機株式会社　羽村技術センター内

審査官　岩田　淳

最終頁に続く

(54)【発明の名称】フィルタ演算処理装置、フィルタ演算方法、及び効果付与装置

(57)【特許請求の範囲】
【請求項１】
フィルタ次数が可変のフィルタ演算処理に用いる、少なくとも指定されたフィルタ次数に対応する数の係数データを記憶している係数記憶部と、
前記フィルタ演算処理の対象となる入力データを順次取得していく取得部と、
前記取得部により過去に取得された、少なくとも前記指定されたフィルタ次数に対応する数の入力データを遅延データとして記憶している入力データ記憶部と、
前記取得部で取得される入力データ又は前記入力データ記憶部に記憶されている各遅延データと、前記係数記憶部に記憶されている各係数データとを乗算しながら累算していく処理を、前記指定されたフィルタ次数に対応する回数繰り返すことで前記フィルタ演算処理を実行し、実行結果を出力する制御部と、
を有し、
前記制御部は、前記フィルタ次数を、予め決められた数以内で任意に指定することが可能であり、
前記係数記憶部は、少なくとも前記予め決められた数の前記係数データを記憶し、
前記入力データ記憶部は、少なくとも前記予め決められた数の前記入力データを記憶する、
フィルタ演算処理装置。
【請求項２】
前記制御部は、前記係数記憶部に記憶されている少なくとも前記予め決められた数の前

記係数データのうち、少なくとも前記指定された数の前記係数データを作業用のメモリに読み込んだ後、前記作業用のメモリに読み込まれた複数の前記係数データを用いて前記フィルタ演算処理を実行する、
請求項１に記載のフィルタ演算処理装置。
【請求項３】
前記フィルタ演算処理は、ＦＩＲフィルタの演算を行うＦＩＲフィルタ演算処理であり、
前記係数記憶部は、指定されたフィルタ次数に対応する数のＦＩＲ係数データを記憶しており、
前記制御部は、前記取得部で取得される入力データ又は前記入力データ記憶部に記憶されている各遅延データと、前記係数記憶部に記憶されている各ＦＩＲ係数データとを乗算しながら累算していく処理を、前記指定されたフィルタ次数に対応する回数繰り返すことで前記ＦＩＲフィルタ演算処理を実行し、実行結果を出力する、
請求項１または２に記載のフィルタ演算処理装置。
【請求項４】
フィルタ次数が可変のフィルタ演算処理に用いる、少なくとも指定されたフィルタ次数に対応する数の係数データを記憶している係数記憶部と、
前記フィルタ演算処理の対象となる入力データを順次取得していく取得部と、
前記取得部により過去に取得された、少なくとも前記指定されたフィルタ次数に対応する数の入力データを遅延データとして記憶している入力データ記憶部と、
前記取得部で取得される入力データ又は前記入力データ記憶部に記憶されている各遅延データと、前記係数記憶部に記憶されている各係数データとを乗算しながら累算していく処理を、前記指定されたフィルタ次数に対応する回数繰り返すことで前記フィルタ演算処理を実行し、実行結果を出力する制御部と、
を有し、
前記係数記憶部は、フィルタ演算処理のフィルタ次数個分の係数データ組を、前記フィルタ次数が可変の１種類以上の前記フィルタ演算処理毎に記憶し、
前記入力データ記憶部は、現在の１サンプル前から、前記フィルタ次数－１サンプル前までの前記フィルタ演算処理に対する入力データ組を前記遅延データ組として、前記フィルタ演算処理毎に記憶し、
１次分の前記フィルタ演算処理を実行するフィルタ演算部を更に備え、
前記制御部は、各サンプリング周期内で、前記フィルタ演算処理毎に、前記フィルタ次数に対応した時分割長の時分割処理により、前記フィルタ演算部に、前記係数記憶部の係数データと現在の入力データ又は前記入力データ記憶部の遅延データとを順次入力させながら１次分の前記フィルタ演算処理を繰り返し実行させて出力データを出力させる、
フィルタ演算処理装置。
【請求項５】
前記制御部は、前記サンプリング周期内で、当該周期を細分化した周期を有するクロックでカウントされ前記フィルタ次数を有する前記フィルタ演算処理を実行可能な長さの連続区間を個別に割り当て、当該連続区間で、前記フィルタ演算部に、前記クロックに同期させて、前記フィルタ演算処理に対応する前記係数記憶部の係数データと現在の入力データ又は前記入力データ記憶部の遅延データとを順次入力させながら、前記乗算と累算の処理を繰り返し実行させ、前記フィルタ演算処理に対応する前記サンプリング周期の出力データを出力させる制御を、前記フィルタ演算処理毎に実行する、請求項４に記載のフィルタ演算処理装置。
【請求項６】
前記フィルタ演算部は、
前記クロックに同期して、前記係数記憶部が出力する係数データを保持する第１のレジスタと、
現在のサンプリング周期の入力データ又は前記入力データ記憶部が出力する遅延データ

の何れかを選択する第１のセレクタと、
前記クロックに同期して、前記第１のセレクタが出力するデータを保持する第２のレジスタと、
前記第１のレジスタが出力する前記係数データと前記第２のレジスタが出力するデータとを乗算する乗算器と、
前記クロックに同期して、前記乗算器が出力する乗算結果データを保持する第３のレジスタと、
前記第３のレジスタが出力する前記乗算結果データと現在までの累算データとを加算する加算器と、
前記クロックに同期して、前記加算器が出力する加算結果データを保持する第４のレジスタと、
値がゼロのデータ又は前記第４のレジスタが出力する加算結果データの何れかを選択し、当該選択されたデータを前記累算データとして前記加算器にフィードバックする第２のセレクタと、
を含む請求項４または５に記載のフィルタ演算処理装置。
【請求項７】
前記制御部は、前記フィルタ演算処理により、付与すべき音響効果音のインパルス応答データの前半データ部分をサンプリング周期単位の時間域処理により原音に畳み込むとともに、前記インパルス応答データの後半データ部分を高速フーリエ変換演算を用いたブロック単位の周波数域処理により前記原音に畳み込むように制御する、
請求項１乃至６のいずれか一項に記載のフィルタ演算処理装置。
【請求項８】
原音に対して付与すべき音響効果音のインパルス応答データに対応する複数の係数データが記憶された係数記憶部と、
前記原音に対応する複数の入力データと、前記係数記憶部に記憶された前記複数の係数データとに基づいて、指定されたフィルタ次数に対応するフィルタ演算処理を実行するフィルタ演算処理部と、
前記フィルタ演算処理により、前記インパルス応答データの前半データ部分をサンプリング周期単位の時間域処理により前記原音に畳み込むとともに、前記インパルス応答データの後半データ部分を高速フーリエ変換演算を用いたブロック単位の周波数域処理により前記原音に畳み込むように制御する制御部と、
を有する効果付与装置。
【請求項９】
前記制御部は、前記インパルス応答データの前半データ部分と後半データ部分とを任意に設定する、請求項８に記載の効果付与装置。
【請求項１０】
前記フィルタ演算処理は、サンプリング周期単位の時間域処理により指定されたフィルタ次数のＦＩＲフィルタの演算を行うＦＩＲフィルタ演算処理であり、
前記ブロックのブロックサイズを決定し、前記ＦＩＲフィルタ演算処理における前記フィルタ次数及び前記高速フーリエ変換演算におけるＦＦＴ点数を前記ブロックサイズの２倍に設定する設定部を更に有する、請求項８または９に記載の効果付与装置。
【請求項１１】
前記設定部は、前記インパルス応答データの時間長に応じて前記ブロックサイズを決定する、請求項１０に記載の効果付与装置。
【請求項１２】
前記設定部は、前記原音の発音設定に応じて前記ブロックサイズを決定する、請求項１０に記載の効果付与装置。
【請求項１３】
前記係数記憶部は、前記インパルス応答データの時間域の係数データを記憶し、
前記制御部は、前記ブロックサイズを決定した後、前記インパルス応答データの係数デ

ータから前記ＦＩＲフィルタ演算処理で用いる前記ブロックサイズの２倍のフィルタ次数の前記ＦＩＲ係数データを除いた残りの係数データを、高速フーリエ変換演算により周波数域の畳込み用係数データに変換し、変換された前記周波数域の畳込み用係数データを用いて、前記インパルス応答データの後半データ部分をブロック単位の周波数域処理により前記原音に畳み込む演算処理を実行する、

請求項１０に記載の効果付与装置。

【請求項１４】

前記制御部は、

前記周波数域の畳込み用係数データに、前記決定したブロックサイズを示すデータを付与して記憶し、

その後、前記ブロックサイズを再度決定した場合には、前記再度決定したブロックサイズが、前記記憶していたブロックサイズと異なる場合のみ、前記周波数域の畳込み用係数データを高速フーリエ変換演算により更新する、

請求項１３に記載の効果付与装置。

【請求項１５】

装置が、

フィルタ次数が可変のフィルタ演算処理のフィルタ次数を、予め決められた数以内で任意に指定し、

少なくとも前記予め決められた数の係数データを記憶している係数記憶部から、前記指定されたフィルタ次数に対応する数の係数データを取得し、

前記フィルタ演算処理の対象となる複数の入力データを順次取得するとともに、前記指定されたフィルタ次数に対応する数の入力データを遅延データとして順次記憶させていき、

順次取得される入力データ又は記憶している各遅延データと、前記取得した各係数データとを乗算しながら累算していく処理を、前記指定されたフィルタ次数に対応する回数繰り返すことで前記フィルタ演算処理を実行し、実行結果を出力する、

フィルタ演算方法。

【請求項１６】

装置が、

原音に対して付与すべき音響効果音のインパルス応答データに対応する複数の係数データを取得し、

前記原音に対応する複数の入力データと、前記取得した前記複数の係数データとに基づいて、指定されたフィルタ次数に対応するフィルタ演算処理を実行し、

前記フィルタ演算処理により、前記インパルス応答データの前半データ部分をサンプリング周期単位の時間域処理により前記原音に畳み込むとともに、前記インパルス応答データの後半データ部分を高速フーリエ変換演算を用いたブロック単位の周波数域処理により前記原音に畳み込むように制御する、

フィルタ演算方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、ＦＩＲ（Ｆｉｎｉｔｅ　Ｉｍｐｕｌｓｅ　Ｒｅｓｐｏｎｓｅ：有限インパルス応答）を利用したフィルタ演算処理装置、フィルタ演算方法、及びそれを用いた効果付与装置に関する。

【背景技術】

【０００２】

オーディオ信号の直接音に対してインパルス応答を畳み込むことにより残響や共鳴効果を付与する残響付与装置において、その畳込み手段として、時間領域で畳み込みを行うＦＩＲフィルタを用いる技術（例えば特許文献１）や、周波数領域で畳み込みを行なうＦＦ

Ｔ／ｉＦＦＴ（Ｆａｓｔ Ｆｏｕｒｉｅｒ Ｔｒａｎｓｆｏｒｍ／ｉｎｖｅｒｓｅ ＦＦＴ：高速フーリエ変換／逆高速フーリエ変換）を用いる技術（例えば特許文献２）が知られている。

【０００３】

また、ＦＩＲとＦＦＴの両方を用い、初期反射音部分はＦＩＲを用い、後期残響音のみＦＦＴで畳み込む方法（例えば非特許文献１）も知られている。

【先行技術文献】

【特許文献】

【０００４】

【特許文献１】特開２００３－２８０６７５号公報

【特許文献２】特開２００５－２１５０５８号公報

【非特許文献】

【０００５】

【非特許文献１】Ｗｉｌｌｉａｍ Ｇ．Ｇａｒｄｎｅｒ “Ｅｆｆｉｃｉｅｎｔ Ｃｏｎｖｏｌｕｔｉｏｎ ｗｉｔｈｏｕｔ Ｉｎｐｕｔ－Ｏｕｔｐｕｔ Ｄｅｌａｙ”, Ｊ．Ａｕｄｉｏ Ｅｎｇ．Ｓｏｃ．, Ｖｏｌ．４３, Ｎｏ．３, ｐ．１２７,１９９５ Ｍａｒｃｈ

【発明の概要】

【発明が解決しようとする課題】

【０００６】

しかし、ＦＩＲフィルタを用いる技術は、残響が長くインパルス応答時間が長くなるような場合には、演算量が膨大となり、残響付与装置として実現することが困難になるという課題があった。

【０００７】

また、ＦＦＴ／ｉＦＦＴを用いる技術は、ブロックサイズ分の入力信号がバッファリングされないとＦＦＴ処理を開始することができないので、効果音の発生において遅延を生じてしまうという課題があった。

【０００８】

更に、ＦＩＲとＦＦＴの両方を用いる技術では、ＦＩＲの出力とＦＦＴの出力を切れ目無く接続するためには、ＦＩＲのフィルタ次数をＦＦＴのポイント数（＝ブロックサイズ×２）に合わせる必要がある。ここで、残響音や共鳴音の特性を変更したり、インパルス応答の特性によって、ＦＦＴのポイント数を調整する必要が生じる場合がある。しかし、従来、ＦＦＴのポイント数の変更に合わせてＦＩＲのフィルタ次数を柔軟に変更することは困難であった。このため、原音に対して残響効果などを与える場合などにおける効果付与のバリエーションが不十分になってしまうという課題があった。

【０００９】

そこで、本発明は、フィルタ次数を柔軟に変更可能で、同時に複数種類のＦＩＲフィルタ演算処理を実行可能なフィルタ演算処理装置、フィルタ演算方法、及びそれを用いた効果付与装置を実現することを目的とする。

【課題を解決するための手段】

【００１０】

態様の一例では、フィルタ次数が可変のフィルタ演算処理に用いる、少なくとも指定されたフィルタ次数に対応する数の係数データを記憶している係数記憶部と、前記フィルタ演算処理の対象となる入力データを順次取得していく取得部と、前記取得部により過去に取得された、少なくとも前記指定されたフィルタ次数に対応する数の入力データを遅延データとして記憶している入力データ記憶部と、前記取得部で取得される入力データ又は前記入力データ記憶部に記憶されている各遅延データと、前記係数記憶部に記憶されている各係数データとを乗算しながら累算していく処理を、前記指定されたフィルタ次数に対応する回数繰り返すことで前記フィルタ演算処理を実行し、実行結果を出力する制御部と、を有し、前記制御部は、前記フィルタ次数を、予め決められた数以内で任意に指定するこ

とが可能であり、前記係数記憶部は、少なくとも前記予め決められた数の前記係数データを記憶し、前記入力データ記憶部は、少なくとも前記予め決められた数の前記入力データを記憶する。

【発明の効果】

【００１１】

本発明によれば、フィルタ次数を柔軟に変更することが可能で、同時に複数種類のＦＩＲフィルタ演算処理を実行可能なフィルタ演算処理装置、フィルタ演算方法、及びそれを用いた効果付与装置を実現することが可能となる。

【図面の簡単な説明】

【００１２】

【図１】電子楽器の実施形態の例を示すブロック図である。

【図２】電子楽器における楽音データの流れの例を示す図である。

【図３】残響・共鳴装置のブロック図である。

【図４】ＣＯＮＶ演算処理部の動作説明図（その１）である。

【図５】ＣＯＮＶ演算処理部の動作説明図（その２）である。

【図６】ＣＯＮＶ演算処理部の動作説明図（その３）である。

【図７】ＦＦＴ演算を用いた周波数域畳込み処理における処理遅延の説明図である。

【図８】ＦＩＲフィルタ演算処理部の機能構成例を示すブロック図である。

【図９】ＦＩＲフィルタ演算処理部とＣＯＮＶ演算処理部の処理関係を示す説明図である。

【図１０】サンプリング周期中に複数系統のＦＩＲフィルタ演算処理を実行可能な本実施形態のＦＩＲフィルタ演算処理部の説明図である。

【図１１】フィルタ演算処理装置のハードウェア構成例を示す図である。

【図１２】フィルタ演算処理装置によって実行される１つのＦＩＲフィルタ演算処理部の動作例を示すタイミングチャートである。

【図１３】図１２のタイミングチャートと図８で説明したＦＩＲフィルタ演算処理部の機能構成例との関係を説明する図である。

【図１４】全体動作の制御処理例を示すメインフローチャートである。

【図１５】残響・共鳴更新処理の詳細処理例を示すフローチャートである。

【図１６】ブロック更新処理の詳細処理例（その１）を示すフローチャートである。

【図１７】ブロック更新処理の詳細処理例（その２）を示すフローチャートである。

【発明を実施するための形態】

【００１３】

以下、本発明を実施するための形態について図面を参照しながら詳細に説明する。本実施形態の電子楽器の効果付与部に用いられるフィルタ演算処理装置は、フィルタ次数を柔軟に変更可能なＦＩＲフィルタ演算処理を実行することができ、また、フィルタ次数やインパルス応答特性が異なる複数のＦＩＲフィルタ演算処理を、それらの組合せを柔軟に変更しながら同時に実行することが可能である。本実施形態における効果付与部は、このようなＦＩＲフィルタ演算処理による時間域畳込み処理と、ＦＦＴ演算を用いた周波数域畳込み処理とが組み合わされて実装される。この場合、周波数域畳込み処理におけるＦＦＴ点数と共にＦＩＲのフィルタ次数を柔軟に決定することが可能となるため、電子楽器の楽音に、応答性を犠牲にすることなく、再現性の高い残響、共鳴効果を付与することが可能となる。

【００１４】

図１は、電子楽器１００の実施形態の例を示すブロック図である。電子楽器１００は、ＣＰＵ（中央演算処理装置）１０１と、ＲＯＭ（リードオンリーメモリ）１０２と、ＲＡＭ（ランダムアクセスメモリ）１０３と、音源（ＴＧ：Ｔｏｎｅ　Ｇｅｎｅｒａｔｏｒ）１０４と、効果付与部１０５と、鍵盤１０６と、ペダル１０７と、操作子１０８が、システムバス１０９に接続された構成を有する。また、音源（ＴＧ）１０４の出力は、サウンドシステム１１０に接続される。

【００１５】
ＣＰＵ１０１は、ＲＯＭ１０２からＲＡＭ１０３にロードされた制御プログラムを実行することにより、鍵盤１０６や操作子１０８からの演奏操作情報に基づいて、音源１０４に発音指示を出す。
【００１６】
音源（ＴＧ）１０４は、上記発音指示に従って、ＲＯＭ１０２又はＲＡＭ１０３から波形データを読み出すことにより、楽音データを生成する。この楽曲データは、効果付与部１０５を介して、サウンドシステム１１０に出力される。このとき、例えばペダル１０７が踏まれると、効果付与部１０５が、楽音データにリバーブなどの残響音付与やピアノの弦の共鳴音付与などの効果付与を実行する。この結果、効果付与部１０５から出力された楽音データは、サウンドシステム１１０において、デジタル－アナログ変換器によってアナログ楽音信号に変換され、アナログアンプで増幅され、スピーカから放音される。
【００１７】
図２は、図１の構成を有する電子楽器における楽音データの流れの例を示す図である。音源（ＴＧ）１０４は、ＣＨ１～ＣＨｎのｎチャネルの発音チャネルの楽音データを生成する楽音生成部２０１（ＣＨ１）～２０１（ＣＨｎ）を備え、それぞれ、鍵盤１０６での押鍵に基づいて発生する図１のＣＰＵ１０１からの発音指示に従い、押鍵毎に独立した楽音データを生成する。発音チャネルＣＨｉに対応する楽音生成部２０１（ＣＨｉ）（１≦ｉ≦ｎ）は、波形データを生成する波形生成部ＷＧ．ＣＨｉ、生成された波形データの音色を加工するフィルタ処理部ＴＶＦ．ＣＨｉ、及び生成された波形データの振幅エンベロープを加工するアンプエンベロープ処理部ＴＶＡ．ＣＨｉを備える。
【００１８】
ミキシング部２０２内の４つのミキサ２０３、２０４、２０５、及び２０６はそれぞれ、各楽音生成部２０１（ＣＨｉ）（１≦ｉ≦ｎ）が出力する各楽音データに所定のレベルを乗算した後に累算することにより、Ｌｃｈ（左チャネル）直接音出力データ２０７、Ｒｃｈ（右チャネル）直接音出力データ２０８、Ｌｃｈ効果音入力データ２０９、及びＲｃｈ効果音入力データ２１０をそれぞれ、効果付与部１０５に出力する。なお、図２において、各ミキサ２０３、２０４、２０５、及び２０６内の記号「＊、Σ」は、入力したデータに対して所定のレベルを乗算後に累算して出力することを示している。
【００１９】
Ｌｃｈ効果音入力データ２０９及びＲｃｈ効果音入力データ２１０はそれぞれ、効果付与部１０５内の残響・共鳴装置２１１において残響・共鳴効果が付与され、Ｌｃｈ効果音出力データ２１１及びＲｃｈ効果音出力データ２１２として出力される。効果付与部１０５内で、Ｌｃｈ効果音出力データ２１１は、Ｌｃｈ直接音出力データ２０７と加算され、Ｌｃｈ楽音出力データ２１３として、図１のサウンドシステム１１０に出力される。同様に、Ｒｃｈ効果音出力データ２１２は、Ｒｃｈ直接音出力データ２０８と加算され、Ｒｃｈ楽音出力データ２１４として、サウンドシステム１１０に出力される。サウンドシステム１１０では、Ｌｃｈ楽音出力データ２１３及びＲｃｈ楽音出力データ２１４はそれぞれ、Ｌｃｈアナログ楽音信号及びＲｃｈアナログ楽音信号に変換されて、それぞれアナログアンプにより増幅されて、Ｌｃｈ及びＲｃｈの各スピーカから放音される。
【００２０】
図３は、図２の効果付与部１０５内の残響・共鳴装置２１１のブロック図である。残響・共鳴装置２１１は、残響・共鳴装置２１１（Ｌｃｈ）と残響・共鳴装置２１１（Ｒｃｈ）とを有する。残響・共鳴装置２１１（Ｌｃｈ）は、Ｌｃｈ効果音入力データ２０９を入力して、Ｌｃｈの残響・共鳴効果を付与し、Ｌｃｈ効果音出力データ２１１を出力する。残響・共鳴装置２１１（Ｒｃｈ）は、Ｒｃｈ効果音入力データ２１０を入力して、Ｒｃｈの残響・共鳴効果を付与し、Ｒｃｈ効果音出力データ２１２を出力する。両者とも同じ構成であるため、以下、特に言及しない場合は、Ｌｃｈ及びＲｃｈの区別なく説明する。
【００２１】
残響・共鳴装置２１１に入力するＬｃｈ効果音入力データ２０９又はＲｃｈ効果音出力

データ２１０は、ＦＩＲフィルタ演算処理部３０３とＣＯＮＶ演算処理部３０４に並列に入力する。ＦＩＲフィルタ演算処理部３０３は、残響・共鳴音のインパルス応答データの前半データ部分をサンプリング周期単位の時間域処理により、Ｌｃｈ効果音入力データ２０９又はＲｃｈ効果音入力データ２１０（原音）に畳み込む時間域畳込み部である。ＣＯＮＶ演算処理部３０４は、上記インパルス応答データの後半データ部分をブロック単位のＦＦＴ演算を用いた周波数域処理により、Ｌｃｈ効果音入力データ２０９又はＲｃｈ効果音入力データ２１０（原音）に畳み込む周波数域畳込み部である。ＦＩＲフィルタ演算処理部３０３とＣＯＮＶ演算処理部３０４は、図２の効果付与部１０５であるＤＳＰ内に実装されているＲＡＭ３０６を共通領域として使用しながら、演算処理を実行する。

【００２２】

ＦＩＲフィルタ演算処理部３０３とＣＯＮＶ演算処理部３０４の各出力は加算部３０５で加算されて、Ｌｃｈ効果音出力データ２１１又はＲｃｈ効果音出力データ２１２として出力される。

【００２３】

図４から図６は、図３のＣＯＮＶ演算処理部３０４の動作説明図である。まず、図４は、ブロックサイズがＮ点の畳込み例を示している。ＦＦＴ演算を用いた畳込みはそのままでは巡回畳込みとなるため、本実施形態では、インパルス応答データ（ｃｏｅｆ）と、Ｌｃｈ効果音入力データ２０９又はＲｃｈ効果音入力データ２１０（以下、ＬｃｈとＲｃｈを区別せずにこれらをまとめて効果音入力データ（ｓｉｇ）と呼ぶ）とを２Ｎ点でそれぞれＦＦＴ演算する際に、各ブロックの線状畳込みが得られるようにする。

【００２４】

ここでは、２Ｎ点の各ＦＦＴ演算４０３、４０４に先だって、４０１として示されるように、Ｎ点のインパルス応答データ（ｃｏｅｆ）（太枠部分）にＮ点のゼロデータが付加されて２Ｎ点のデータとされる。また、４０２として示されるように、効果音入力データ（ｓｉｇ）がブロックサイズＮ点分シフトされながらオーバーラップされて（太枠部分→破線太枠部分）２Ｎ点のデータとされる。そして、インパルス応答データ（ｃｏｅｆ）から生成された上記２Ｎ点のデータ４０１と効果音入力データ（ｓｉｇ）から生成された上記２Ｎ点のデータ４０２とに対して、それぞれ４０３及び４０４として示されるようにＦＦＴ演算が実行される結果、周波数領域の２Ｎ点のデータ４０５及び４０６が得られる。

【００２５】

続いて、４０７として示されるように、周波数領域の２Ｎ点のデータ４０５及び４０６が各周波数点毎に複素乗算されることにより、２Ｎ点の複素乗算結果データ４０８が得られる。

【００２６】

更に、４０９として示されるように、２Ｎ点の複素乗算結果データ４０８に対してｉＦＦＴ演算が実行される結果、畳込みが行われた２Ｎ点の時間域データ４１０が得られる。

【００２７】

そして、２Ｎ点の時間域データ４１０のうちの前半のＮ点のデータ（太枠部分）が、ブロックサイズがＮ点の重畳保留法での線状畳込み結果となり、このようにして生成されるブロックサイズＮ点ずつのデータが図３のＬｃｈ効果音出力データ２１１又はＲｃｈ効果音出力データ２１２として出力される。

【００２８】

図５は、インパルス応答データ（ｃｏｅｆ）側もブロックサイズＮ点ずつ分割される場合の、ブロックサイズがＮ点の畳込み例を示している。この例では、各Ｎ点毎の畳込み結果が足し合わせられることで、長い時間のインパルス応答もＮ点ずつの小さなブロックに分割して畳込み演算を実行することができる。

【００２９】

図５において、５０１として示されるように、インパルス応答データ（ｃｏｅｆ）が、それぞれのブロックサイズがＮずつの例えば分割データＮ１、Ｎ２、Ｎ３、Ｎ４に分割される。また、５０２として示されるように、効果音入力データ（ｓｉｇ）も、それぞれの

ブロックサイズがＮずつの例えば分割データＮ１、Ｎ２、Ｎ３、Ｎ４に分割される。

【００３０】

その後は、図４の場合と同様にして、２Ｎ点の各ＦＦＴ演算５０５、５０６に先だって、５０３として示されるように、インパルス応答データ（ｃｏｅｆ）中の上述のように分割されたＮ点の分割データ（太枠部分）にＮ点のゼロデータが付加されて２Ｎ点のデータとされる。また、５０４として示されるように、効果音入力データ（ｓｉｇ）が上述のように分割されて得られるブロックサイズＮ点分ずつシフトされながらオーバーラップされて（太枠部分→破線太枠部分）２Ｎ点のデータとされる。そして、インパルス応答データ（ｃｏｅｆ）を分割したＮ点のデータから生成された上記２Ｎ点のデータ５０３と効果音入力データ（ｓｉｇ）から生成された上記２Ｎ点のデータ５０４とに対して、それぞれ５０５及び５０６として示されるようにＦＦＴ演算が実行される結果、周波数領域の２Ｎ点のデータ５０７及び５０８が、５０９及び５１０のように順次２Ｎ点ずつ得られる。ここで、インパルス応答データ（ｃｏｅｆ）の分割データＮ１、Ｎ２、Ｎ３、Ｎ４（図５の５０１）から生成される２Ｎ点ずつの周波数データ群５０９（Ｎ１、Ｎ２、Ｎ３、Ｎ４）は、インパルス応答データ（ｃｏｅｆ）に変化がない限り、ＦＦＴ演算により演算しメモリにプリセットしておくことができる。また、効果音入力データ（ｓｉｇ）から生成される２Ｎ点ずつの周波数データ群５１０は、インパルス応答データ（ｃｏｅｆ）に対応する周波数データ群５０９（Ｎ１、Ｎ２、Ｎ３、Ｎ４）と同じ分割データ数分だけ、ｔ（Ｎ）、ｔ（Ｎ－１）、ｔ（Ｎ－２）、ｔ（Ｎ－３）というようにリングバッファ形式で順次メモリに記憶されてよい。

【００３１】

続いて、５１１として示されるように、周波数データ群５０９（Ｎ１、Ｎ２、Ｎ３、Ｎ４）と周波数データ群５１０（ｔ（Ｎ）、ｔ（Ｎ－１）、ｔ（Ｎ－２）、ｔ（Ｎ－３））とが、２Ｎ点ずつ各周波数点毎に複素乗算されることにより、それぞれ２Ｎ点の複素乗算結果データＮ１＊ｔ（Ｎ）、Ｎ２＊ｔ（Ｎ－１）、Ｎ３＊ｔ（Ｎ－２）、Ｎ４＊ｔ（Ｎ－３）が得られ、更にこれらに対してｉＦＦＴ演算が実行される結果、畳込みが行われた２Ｎ点ずつの時間域データ５１３が得られる。

【００３２】

そして、２Ｎ点ずつの時間域データ５１３のうちのそれぞれ前半のＮ点ずつのデータ（太枠部分）が、図３のＬｃｈ効果音出力データ２１１又はＲｃｈ効果音出力データ２１２として出力される。

【００３３】

図６は、図３のＣＯＮＶ演算処理部３０４の動作を説明するための簡単な計算例を示した図である。

【００３４】

例えば、

```
conv(coef=[2 3 4 5 6 7 8 9]
sig= [1 2 3 4 5 6 7 8 9 10])
```

をブロックサイズＮ＝２で計算した場合、畳込み結果の９番目と１０番目は、200と244になるが、これを求めるためには、

```
ifft(fft([0 0 2 3]) *  fft([7 8 9 10]))
ifft(fft([0 0 4 5]) *  fft([5 6 7 8]))
ifft(fft([0 0 6 7]) *  fft([3 4 5 6]))
ifft(fft([0 0 8 9]) *  fft([1 2 3 4]))
```

の各計算結果の前半部分を足し合わせればよい。なお、「fft()」及び「ifft()」はそれぞれ、括弧内のデータ群に対するＦＦＴ演算及びｉＦＦＴ演算を示す。

【００３５】

上記演算例により、ＦＦＴ演算を使った線状畳込みを実現できるが、ＦＦＴ演算はブロック単位で計算が実行されるため、ブロックの頭で遅延が生じてしまう。図７は、ＦＦＴ演算を用いた周波数域畳込み処理における処理遅延の説明図である。

【００３６】
例えば、ブロック区間Ｔ１では、Ｎ１ブロックのデータが入力される。次のブロック区間Ｔ２では、Ｎ２ブロックのデータの入力とＮ１ブロックのデータの畳込み処理（ＦＦＴ演算、複素乗算、及びｉＦＦＴ演算）が実行される。更に続くブロック区間Ｔ３では、Ｎ３ブロックの入力とＮ２ブロックのデータの畳込み処理と、Ｎ１ブロックのデータの出力処理が実行される。ブロック区間Ｔ４では、Ｎ４ブロックの入力とＮ３ブロックのデータの畳込み処理と、Ｎ２ブロックのデータの出力処理が実行される。ブロック区間Ｔ５では、Ｎ５ブロックの入力とＮ４ブロックのデータの畳込み処理と、Ｎ３ブロックのデータの出力処理が実行される。
【００３７】
図７に示されるタイミング関係からわかるように、ＦＦＴ演算を使った線状畳込み処理では、７０１として示されるように、ブロックサイズＮに対して２Ｎの処理遅延が発生する。従って、図３の残響・共鳴装置２１１では、ＣＯＮＶ演算処理部３０４で発生する上記２Ｎの処理遅延をカバーするために、フィルタ次数＝２ＮのＦＩＲフィルタ演算処理部３０３が実装される。
【００３８】
図８は、ＦＩＲフィルタ演算処理部３０３の機能構成例を示すブロック図である。この図は、乗算処理部８０１と、累算処理部８０２と、遅延処理部８０３とからなる１次のフィルタ演算部８００が、＃０から＃２Ｎ－１までカスケードに接続された、フィルタ次数＝２Ｎの直接型構成のＦＩＲフィルタを示している。但し、最終段＃２Ｎ－１は、遅延処理部８０３は不要である。
【００３９】
具体的には、効果音入力データ（図３のＬｃｈ効果音入力データ２０９又はＲｃｈ効果音入力データ２１０）は、第０次の乗算処理部８０１（＃０）にて第０次のＦＩＲ係数データが乗算され、その乗算結果データは、第０次の累算処理部８０２（＃０）にて前段の累算結果データ（第０次の前段は無いため値０のデータ）に累算される。また、効果音入力データは第０次の遅延処理部８０３（＃０）にて１サンプリング周期分遅延させられる。
【００４０】
次に、第０次の遅延処理部８０３（＃０）の出力は、第１次の乗算処理部８０１（＃１）にて第１次のＦＩＲ係数データが乗算され、その乗算結果データは、第１次の累算処理部８０２（＃１）にて前段の第０次の累算処理部８０２（＃０）の累算結果データに累算される。また、第０次の遅延処理部８０３（＃０）の出力データは第１次の遅延処理部８０３（＃１）にて１サンプリング周期分遅延させられる。
【００４１】
以下同様にして、第０次から第２Ｎ－１次まで、各次のＦＩＲ演算処理が実行される。一般的に、第ｉ次（０≦ｉ≦２Ｎ－１）のＦＩＲ演算処理遅延処理部８０３（＃ｉ－１）の出力（ｉ＝０の場合は効果音入力データの入力）は、第ｉの乗算処理部８０１（＃ｉ）にて第ｉ次のＦＩＲ係数データが乗算され、その乗算結果データは、第ｉ次の累算処理部８０２（＃ｉ）にて前段の第ｉ－１次の累算処理部８０２（＃ｉ－１）の累算結果データ（ｉ＝０の場合は値０のデータ）に累算される。また、第ｉ－１次の遅延処理部８０３（＃０）の出力データ（ｉ＝０の場合は効果音入力データ）は第ｉ次の遅延処理部８０３（＃ｉ）にて１サンプリング周期分遅延させられる。
【００４２】
最終段第２Ｎ－１次の累算処理部８０２（＃２Ｎ－１）の累算結果データは、畳込み結果データとして出力される。なお、最終段第２Ｎ－１次の遅延処理部８０３（＃２Ｎ－１）は不要である。
【００４３】
以上の図８に示される機能構成を有するＦＩＲフィルタ演算処理部３０３において、＃０から＃２Ｎ－２までの遅延処理部８０３は、リングバッファ式のメモリに効果音入力デ

ータを順次記憶させてゆく処理として実現できる。

【００４４】

また、効果音入力データの信号入力は、図２の音源（ＴＧ）１０４内のミキシング部２０２内のミキサ２０５又は２０６から、サンプリング周期単位で行われる。

【００４５】

各次の乗算処理部８０１及び累算処理部８０２によるＦＩＲ演算処理は、サンプリング周期内で、その周期を細分化した周期を有するクロックに同期して実行され、全次数でのＦＩＲ演算処理及び最終段第２Ｎ－１次の累算処理部８０２（＃２Ｎ－１）からの畳込み結果データの出力は、そのサンプリング周期内で完了する。これにより、図３のＦＩＲフィルタ演算処理部３０３での畳込み処理では、遅延が生じない。

【００４６】

ここで、ＦＩＲフィルタ演算処理部３０３のフィルタ次数がＣＯＮＶ演算処理部３０４のＦＦＴサイズ（ブロックサイズＮの２倍）に等しければ、ＣＯＮＶ演算処理部３０４での処理遅延（図７参照）の影響は無視できる。

【００４７】

即ち、効果音入力データの最初の２ＮサンプルがＦＩＲフィルタ演算処理部３０３に入力されると同時に、効果音入力データが、Ｎサンプルずつシフトされながら２ＮサンプルずつＣＯＮＶ演算処理部３０４に入力されるようにする。図９は、ＦＩＲフィルタ演算処理部３０３とＣＯＮＶ演算処理部３０４の処理関係を示す説明図である。図９に示されるように、まず、ＦＩＲフィルタ演算処理部３０３で、効果音入力データの最初の２Ｎサンプルに対して、サンプリング周期毎に残響音又は共鳴音のインパルス応答データの最初の２Ｎサンプルを畳込み演算して得られる、２Ｎサンプル分の畳込み結果データが、サンプリング周期毎に遅延なく出力される。また、ＣＯＮＶ演算処理部３０４で、効果音入力データをＮサンプルずつオーバーラップさせながら入力した２Ｎサンプル分ずつに対して、残響音又は共鳴音のインパルス応答データの２Ｎ＋１サンプル以降のＮサンプル＋値０のＮサンプル＝２Ｎサンプル分ずつを周波数域で畳込み演算して得られる、Ｎサンプル分ずつの畳込み結果データが、２Ｎサンプル分ずつ遅延して順次出力される。従って、図３の加算部３０５から、ＦＩＲフィルタ演算処理部３０３から最初の２Ｎサンプル分の畳込み結果データを出力させるのに続き、ＣＯＮＶ演算処理部３０４から順次出力されるＮサンプル毎の畳込み結果データをサンプリング周期毎に順次出力させることにより、効果音入力データの全体に対してインパルス応答データを畳込み演算した効果音出力データ（Ｌｃｈ効果音出力データ２１１又はＲｃｈ効果音出力データ２１２）を、図３の残響・共鳴装置２１１から遅延なく出力することが可能となる。

【００４８】

ＦＩＲフィルタ演算処理部３０３は、サンプリング周期毎に処理を実行するが、電子楽器では残響（リバーブ）のタイプを複数有することが多く、このタイプにより畳込みを行う際のインパルス応答は様々な時間のものになり得る。また、共鳴や胴鳴りについても楽器の大きさなどにより多様となり得る。このため処理遅延が無い構成においてブロックサイズを固定化（例えばインパルス応答データが一番長い場合を基にしたブロックサイズに固定化）するのは好ましくない。

【００４９】

また、本実施形態では、残響・共鳴装置２１１は、Ｌｃｈ効果音入力データ２０９及びＲｃｈ効果音入力データ２１０に対してそれぞれ用意されるため、ＦＩＲフィルタ演算処理部３０３は、最低でも２つを用意する必要がある。

【００５０】

そこで、本実施形態のＦＩＲフィルタ演算処理部３０３は、以下に説明する構成によって、フィルタ次数を柔軟に変更可能なＦＩＲフィルタ演算処理を実行することができ、また、複数のＦＩＲフィルタ演算処理を、それらの組合せを柔軟に変更しながら同時に実行することを可能とするものである。図１０は、サンプリング周期中に複数系統のＦＩＲフィルタ演算処理を実行可能な本実施形態のＦＩＲフィルタ演算処理部３０３の説明図であ

る。図１０において、ＦＩＲ（１）、ＦＩＲ（２）、・・・、ＦＩＲ（Ｘ－１）、ＦＩＲ（Ｘ）は、それぞれがＦＩＲフィルタ演算処理部を示している。本実施形態では、図３に示されるＬｃｈ及びＲｃｈそれぞれのＦＩＲフィルタ演算処理部３０３を始めとして、特には図示しないその他のＦＩＲフィルタ演算処理部は、１つのフィルタ演算処理装置が時分割処理により実行する個別のＦＩＲフィルタ演算処理機能に対応している。そして、本実施形態では、各サンプリング周期内で、ＦＩＲフィルタ演算処理部ＦＩＲ（１）、ＦＩＲ（２）、・・・、ＦＩＲ（Ｘ－１）、ＦＩＲ（Ｘ）毎に、それぞれのフィルタ次数に対応した時分割長の時分割処理により、図８で機能構成例で説明したフィルタ次数分のＦＩＲフィルタ演算処理が実行される。このとき、制御部として動作する例えば図１のＣＰＵ１０１は、サンプリング周期内で、ＦＩＲフィルタ演算処理部ＦＩＲ（１）、ＦＩＲ（２）、・・・、ＦＩＲ（Ｘ－１）、ＦＩＲ（Ｘ）のそれぞれに、サンプリング周期を細分化した周期を有するクロックでカウントされそれぞれのフィルタ次数分のＦＩＲフィルタ演算処理を実行可能な長さ（クロック数）の連続区間を個別に割り当て、その連続区間で、図１の効果付与部１０５の例えばＤＳＰ（Ｄｉｇｉｔａｌ　Ｓｉｇｎａｌ　Ｐｒｏｃｅｓｓｏｒ：デジタル信号プロセッサ）に、それぞれのフィルタ次数分のＦＩＲフィルタ演算処理を実行させる。

【００５１】

図１１は、図３のＬｃｈ及びＲｃｈの各ＦＩＲフィルタ演算処理部３０３を上記時分割処理により実現するフィルタ演算処理装置のハードウェア構成例を示す図である。

【００５２】

ＦＩＲ係数メモリ１１０１（ＦＩＲ係数記憶部）は、ＦＩＲフィルタ演算処理のフィルタ次数個分のＦＩＲ係数データ組を、フィルタ次数が可変の図１０で説明した１種類以上のＦＩＲフィルタ演算処理部ＦＩＲ（１）、ＦＩＲ（２）、・・・、ＦＩＲ（Ｘ－１）、ＦＩＲ（Ｘ）毎に記憶する。

【００５３】

図１１のＦＩＲ係数メモリ１１０１では、例えば３種類のＦＩＲフィルタ演算処理部ＦＩＲ（１）、ＦＩＲ（２）、ＦＩＲ（３）毎の３組のＦＩＲ係数データ組ｂ０、ｂ１、及びｂ２が記憶される。記憶されるＦＩＲ係数データ組ｂ０、ｂ１、及びｂ２のそれぞれの係数の個数は、それぞれＦＩＲフィルタ演算処理部ＦＩＲ（１）、ＦＩＲ（２）、ＦＩＲ（３）の夫々のフィルタ次数に対応する。

【００５４】

より具体的には、例えばＦＩＲフィルタ演算処理部ＦＩＲ（１）とＦＩＲ（２）が、図３のＬｃｈ効果音入力データ２０９を処理する残響・共鳴装置２１１（Ｌｃｈ）内のＦＩＲフィルタ演算処理部３０３とＲｃｈ効果音入力データ２１０を処理する残響・共鳴装置２１１（Ｒｃｈ）内の各ＦＩＲフィルタ演算処理部３０３であるとする。この場合、ＦＩＲ係数メモリ１１０１に記憶されるＦＩＲ係数データ組ｂ１は、Ｌｃｈ用の残響・共鳴音のインパルス応答データの最初の２Ｎ個のデータ組である。同様に、ＦＩＲ係数メモリ１１０１に記憶されるＦＩＲ係数データ組ｂ２は、Ｒｃｈ用の残響・共鳴音のインパルス応答データの最初の２Ｎ個のデータ組である。

【００５５】

データメモリ１１０２（入力データ記憶部）においては、ＦＩＲフィルタ演算処理部ＦＩＲ（１）、ＦＩＲ（２）、・・・、ＦＩＲ（Ｘ－１）、ＦＩＲ（Ｘ）毎にそれぞれ、（各フィルタ次数－１）アドレス分のリングバッファ形式の記憶領域が確保される。各記憶領域にはそれぞれ、現在の１サンプル前から（フィルタ次数－１）サンプル前までの各ＦＩＲフィルタ演算処理部ＦＩＲ（１）、ＦＩＲ（２）、・・・、ＦＩＲ（Ｘ－１）、ＦＩＲ（Ｘ）に対する入力データ１１１１の組が、遅延データ組として記憶される。

【００５６】

図１１のデータメモリ１１０２では、例えば３種類のＦＩＲフィルタ演算処理部ＦＩＲ（１）、ＦＩＲ（２）、ＦＩＲ（３）毎にそれぞれ、（各フィルタ次数－１）アドレス分の３つのリングバッファ形式の記憶領域が確保され、各記憶領域にはそれぞれ、遅延デー

タ組ｂ０ｗｍ、ｂ１ｗｍ、及びｂ２ｗｍが記憶される。各記憶領域において、サンプリング周期毎に、書込みアドレスが記憶領域の先頭アドレスアドレスから順次インクリメントされながら、各サンプリング周期毎の入力データ１１１１が上記記憶領域の記憶アドレスに順次書き込まれてゆく。書込みアドレスがインクリメントにより上記記憶領域の末尾アドレスを超えると、書込みアドレスが上記記憶領域の先頭アドレスに戻されて、入力データ１１１１の書込みが続行される。このようにして、各記憶領域には、リング状に、現在の１サンプル前から（フィルタ次数－１）サンプル前までの遅延データが書き込まれる。読出し時には、後述するように、サンプリング周期を細分化した周期を有するサンプリングクロックよりも速いクロックに同期して、上記と同様にリング状に読出しアドレスが制御されることにより、各記憶領域から、現在の１サンプル前から（フィルタ次数－１）サンプル前までの遅延データが読み出される。

【００５７】

今例えば、ＦＩＲ（１）が、図３の残響・共鳴装置２１１（Ｌｃｈ）内のＬｃｈ用のＦＩＲフィルタ演算処理部３０３であるとする。この場合、ＦＩＲ（１）に対応する入力データ１１１１は、Ｌｃｈ効果音入力データ２０９である。また、Ｌｃｈ効果音入力データ２０９の現在のサンプリング周期のサンプル値をｂ１ｉｎとする。この場合、データメモリ１１０２のＦＩＲ（１）に対応する記憶領域には、Ｌｃｈ効果音入力データ２０９の現在から１サンプリング周期前のサンプル値ｂ１ｗｍ（１）から（２Ｎ－１）サンプリング周期前のサンプル値ｂ１ｗｍ（２Ｎ－１）までが、遅延データ組ｂ１ｗｍとして記憶される。

【００５８】

同様に、ＦＩＲ（２）が図３の残響・共鳴装置２１１（Ｒｃｈ）内のＲｃｈ用のＦＩＲフィルタ演算処理部３０３であるとする。この場合、ＦＩＲ（２）に対応する入力データ１１１１は、Ｒｃｈ効果音入力データ２１０ある。また、Ｒｃｈ効果音入力データ２１０の現在のサンプリング周期のサンプル値をｂ２ｉｎとする。この場合、データメモリ１１０２のＦＩＲ（２）に対応する記憶領域には、Ｒｃｈ効果音入力データ２１０の現在から１サンプリング周期前のサンプル値ｂ２ｗｍ（１）から（２Ｎ－１）サンプリング周期前のサンプル値ｂ２ｗｍ（２Ｎ－１）までが、遅延データ組ｂ２ｗｍとして記憶される。

【００５９】

次に、図１１において、第１のレジスタ（ｍ０ｒ）１１０３、第１のセレクタ（ＳＥＬ１）１１１０、第２のレジスタ（ｍ１ｒ）１１０４、乗算器１１０５、第３のレジスタ（ｍｒ）１１０６、加算器１１０７、第４のレジスタ（ａｒ）１１０８、及び第２のセレクタ（ＳＥＬ２）１１０９は、１次分のＦＩＲ乗算／累算処理を実行するフィルタ演算部１１００を構成する。このフィルタ演算部１１００は、図８のフィルタ演算部８００の機能を実現する。

【００６０】

フィルタ演算部１１００において、第１のレジスタ（ｍ０ｒ）１１０３は、サンプリング周期を細分化した周期のクロックに同期して、ＦＩＲ係数メモリ１１０１が出力するＦＩＲ係数データを保持する。

【００６１】

フィルタ演算部１１００において、第１のセレクタ（ＳＥＬ１）１１１０は、現在のサンプリング周期の入力データ１１１１又はデータメモリ１１０２が出力する遅延データの何れかを選択する。

【００６２】

フィルタ演算部１１００において、第２のレジスタ（ｍ１ｒ）１１０４は、クロックに同期して、セレクタ（ＳＥＬ１）１１１０が出力するデータを保持する。

【００６３】

フィルタ演算部１１００において、乗算器１１０５は、第１のレジスタ（ｍ０ｒ）１１０３が出力するＦＩＲ係数データと第２のレジスタ（ｍ１ｒ）１１０４が出力するデータとを乗算する。

【００６４】
フィルタ演算部１１００において、第３のレジスタ（ｍｒ）１１０６は、クロックに同期して、乗算器１１０５が出力する乗算結果データを保持する。
【００６５】
フィルタ演算部１１００において、加算器１１０７は、第３のレジスタ（ｍｒ）１１０６が出力する乗算結果データと、後述するセレクタ（ＳＥＬ２）１１０９が出力するデータとを加算する。
【００６６】
フィルタ演算部１１００において、第４のレジスタ（ａｒ）１１０８は、クロックに同期して、加算器１１０７が出力する加算結果データを保持する。
【００６７】
フィルタ演算部１１００において、セレクタ（ＳＥＬ２）１１０９は、値がゼロのデータ又は第４のレジスタ（ａｒ）１１０８が出力する加算結果データの何れかを選択して累算データとして、加算器１１０７にフィードバックする。
【００６８】
図１１の構成において、図１０で説明した各ＦＩＲフィルタ演算処理部ＦＩＲ（ｉ）（１≦ｉ≦Ｘ）の処理では、それに対応して割り当てられたサンプリング周期内の前述した連続区間において、上述のフィルタ演算部１１００が、クロックに同期して、上記ＦＩＲフィルタ演算処理部ＦＩＲ（ｉ）に対応するＦＩＲ係数メモリ１１０１中のＦＩＲ係数データを第１のレジスタ（ｍ０ｒ）１１０３に順次入力させると共に、それぞれＦＩＲフィルタ演算処理部ＦＩＲ（ｉ）に対応する現在の入力データ１１１１又はデータメモリ１１０２から出力される遅延データを第１のセレクタ（ＳＥＬ１）１１１０から第２のレジスタ（ｍ１ｒ）１１０４に順次入力させながら、ＦＩＲ乗算／累算処理をフィルタ次数に対応する回数だけ繰り返し実行し、その実行が完了した時点で第４のレジスタ（ａｒ）１１０８の内容を、畳込み結果データとして出力する。
【００６９】
図１０で説明したＦＩＲフィルタ演算処理部ＦＩＲ（ｉ）（１≦ｉ≦Ｘ）毎に、それに対応して割り当てられたサンプリング周期内の独立した連続区間において時分割処理が実行されることにより、サンプリング周期毎に、上記１つ以上のＦＩＲフィルタ演算処理部ＦＩＲ（ｉ）の演算を時分割で実行することが可能となり、それぞれの畳込み結果データを出力することが可能となる。また、各ＦＩＲフィルタ演算処理部ＦＩＲ（ｉ）においては、それぞれのフィルタ次数に対応したフィルタ係数データ組をＦＩＲ係数メモリ１１０１に記憶させることにより、用途に応じたフィルタ次数に柔軟に対応することが可能である。
【００７０】
図１２は、図１１のハードウェア構成例を有するフィルタ演算処理装置によって実行される１つのＦＩＲフィルタ演算処理部ＦＩＲ（ｉ）の動作例を示すタイミングチャートである。また、図１３は、図１２のタイミングチャートと図８で説明したＦＩＲフィルタ演算処理部の機能構成例との関係を説明する図である。
【００７１】
今、説明の簡単のために、ブロックサイズＮ＝２の場合を例にして、１サンプリング周期内におけるＦＩＲフィルタ演算処理部ＦＩＲ（１）の畳込み結果データの出力処理と、それに続くＦＩＲ（２）の一部処理について説明する。実際のＦＦＴ演算処理装置では、例えば、Ｎ＝１２８やＮ＝２５６であるが、動作原理は同じである。
【００７２】
この場合、ＦＩＲ（１）が、図３の残響・共鳴装置２１１（Ｌｃｈ）内のＬｃｈ用のＦＩＲフィルタ演算処理部３０３であり、ＦＩＲ（２）が、図３の残響・共鳴装置２１１（Ｒｃｈ）内のＲｃｈ用のＦＩＲフィルタ演算処理部３０３である。
【００７３】
そして、図１１のＦＩＲ係数メモリ１１０１には、ＦＩＲ係数データ組ｂ１として、Ｌ

ｃｈ用の残響・共鳴音のインパルス応答データの最初の２Ｎ＝４個のデータ組［ｂ１（０），ｂ１（１），ｂ１（２），ｂ１（３）］が記憶される。これらのデータは、例えば図１３に示されるように、＃０から＃３のフィルタ演算部８００に含まれる＃０から＃３の乗算器８０１の各乗算係数として与えられる。同様に、ＦＩＲ係数メモリ１１０１には、ＦＩＲ係数データ組ｂ１として、Ｒｃｈ用の残響・共鳴音のインパルス応答データの最初の２Ｎ＝４個のデータ組［ｂ２（０），ｂ２（１），ｂ２（２），ｂ２（３）］が記憶される。

【００７４】

また、ＦＩＲ（１）に対応する入力データ１１１１は、図２又は図３のＬｃｈ効果音入力データ２０９である。Ｌｃｈ効果音入力データ２０９の現在のサンプリング周期のサンプル値をｂ１ｉｎとする。更に、図１１のデータメモリ１１０２のＦＩＲ（１）に対応する記憶領域には、Ｌｃｈ効果音入力データ２０９の現在から１サンプリング周期前のサンプル値ｂ１ｗｍ（１）から（２Ｎ－１）＝３サンプリング周期前のサンプル値ｂ１ｗｍ（３）までが、遅延データ組ｂ１ｗｍとして記憶される。図１３においては、ｂ１ｉｎはＬｃｈ効果音入力データ２０９の入力、ｂ１ｗｍ１はｂ１ｉｎを１サンプリング周期分遅延させた遅延処理部８０３（＃０）の出力、ｂ１ｗｍ２はｂ１ｗｍ１を更に１サンプリング周期分遅延させた遅延処理部８０３（＃１）の出力、ｂ１ｗｍ３はｂ１ｗｍ２を更に１サンプリング周期分遅延させた遅延処理部８０３（＃２）の出力に、それぞれ対応する。

【００７５】

同様に、ＦＩＲ（２）に対応する入力データ１１１１は、図２又は図３のＲｃｈ効果音入力データ２１０である。Ｒｃｈ効果音入力データ２１０の現在のサンプリング周期のサンプル値をｂ２ｉｎとする。更に、データメモリ１１０２のＦＩＲ（２）に対応する記憶領域には、Ｒｃｈ効果音入力データ２１０の現在から１サンプリング周期前のサンプル値ｂ２ｗｍ（１）から３サンプリング周期前のサンプル値ｂ２ｗｍ（３）までが、遅延データ組ｂ２ｗｍとして記憶される。

【００７６】

図１２において、Ｌ０（ｍ）、Ｌ１（０）、Ｌ１（１）、Ｌ１（２）、Ｌ１（３）、Ｌ２（０）、Ｌ２（１）、Ｌ２（２）、・・・は、１サンプリング周期を細分化した周期のクロックのタイミングを示している。このクロックの周期は、図１０の複数系統のＦＩＲフィルタ演算処理部ＦＩＲ（１）、ＦＩＲ（２）、・・・、ＦＩＲ（Ｘ－１）、ＦＩＲ（Ｘ）の各最大フィルタ次数の総和をＡとしたとき、「サンプリング周期＝クロックの周期×Ａ」となるのが理想である。しかし、実際にはこのクロックの周期は、図１の効果付加装置１００の性能や、必要とするＦＩＲフィルタ演算処理部の種類やフィルタ次数、及びＦＩＲ係数メモリ１１０１とデータメモリ１１０２の容量等を、図１の電子楽器１００のコストを考慮しながら換算して決定される。いずれにしても、本実施形態では、各ＦＩＲフィルタ演算処理部の種類やフィルタ次数は柔軟に設定することが可能である。

【００７７】

図１２において、クロックタイミングＬ１（０）～Ｌ１（３）は、１サンプリング周期内でＦＩＲフィルタ演算処理部ＦＩＲ（１）（＝図３のＬｃｈ用のＦＩＲフィルタ演算処理部３０３）を時分割処理で実行させるために割り当てられた連続区間であり、ＦＩＲ（１）のフィルタ次数＝２Ｎ＝４に対応するクロック数４の区間長を有する。同様に、クロックタイミングＬ２（０）～Ｌ２（３）は、１サンプリング周期内でＦＩＲフィルタ演算処理部ＦＩＲ（２）（＝図３のＲｃｈ用のＦＩＲフィルタ演算処理部３０３）を時分割処理で実行させるために割り当てられた連続区間であり、ＦＩＲ（２）のフィルタ次数＝２Ｎ＝４に対応するクロック数４の区間長を有する。なお、Ｌ０（ｍ）は、ＦＩＲ（１）の直前に実行されるＦＩＲフィルタ演算処理部ＦＩＲ（０）のフィルタ次数＝ｍ（ｍは自然数）に対応する連続区間の最後のクロックタイミングである。

【００７８】

まず、図１２のクロックタイミングＬ０（ｍ）で、図２のミキサ２０５から出力するＬｃｈ効果音入力データ２０９の値が、現在のサンプリング周期のサンプル値ｂ１ｉｎとな

り（図１２（ａ））、データメモリ１１０２への書込み信号ｗｒｔがハイレベル（ｈｉｇｈ）となる。これにより、データメモリ１１０２内のＦ（１）に対応する記憶領域内の書込みアドレスポインタが示すアドレスに、サンプル値ｂ１ｉｎが書き込まれる。これと共に、ＦＩＲ係数メモリ１１０１に記憶されているＦ（１）に対応するＦＩＲ係数データ組ｂ１内の読出しアドレスポインタが示すアドレスから、１番目のＦＩＲ係数データｂ１（０）が読み出される（図１２（ｆ））。

【００７９】

次に、図１２のクロックタイミングＬ１（０）で、データメモリ１１０２への書込み信号ｗｒｔがローレベル（ｌｏｗ）になる（図１２（ｂ））。これと共に、図１１の第１のセレクタ（ＳＥＬ１）１１１０が入力側ｉｎを選択することにより（図１２（ｃ））、クロックタイミングＬ０（ｍ）で入力していた現在のサンプリング周期のサンプルｂ１ｉｎが、クロックタイミングＬ１（０）の立ち上がりで第２のレジスタ（ｍ１ｒ）１１０４にラッチされ、その出力がｂ１ｉｎとなる（図１２（ａ）→（ｅ））。また、クロックタイミングＬ０（ｍ）でＦＩＲ係数メモリ１１０１から出力されていた１番目のＦＩＲ係数データｂ１（０）が、クロックタイミングＬ１（０）の立ち上がりで第１のレジスタ（ｍ０ｒ）１１０３にラッチされ、その出力がｂ１（０）になる（図１２（ｆ）→（ｇ））。なお、ＦＩＲ係数メモリ１１０１において、クロックタイミングＬ１（０）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、ＦＩＲ係数メモリ１１０１からは、ＦＩＲ係数データ組ｂ１内の２番目のＦＩＲ係数データｂ１（１）が出力される（図１２（ｆ））。また、データメモリ１１０２において、クロックタイミングＬ１（０）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、データメモリ１１０２からは、遅延データ組ｂ１ｗｍ内の１番目（＝現在の１サンプリング周期前）の遅延データｂ１ｗｍ１が出力される（図１２（ｄ））。そして、クロックタイミングＬ１（０）において、第２のレジスタ（ｍ１ｒ）１１０４の出力ｂ１ｉｎと、第１のレジスタ（ｍ０ｒ）１１０３の出力ｂ１（０）とが乗算器１１０５に入力することにより、乗算結果データが「ｂ１ｉｎ＊ｂ１（０）」（「＊」は乗算を示す）となる（図１２（ｅ）、（ｇ）→（ｈ））。クロックタイミングＬ１（０）での上述の演算処理が、図１３の乗算処理部８０１（＃０）の動作に対応する。

【００８０】

次に、図１２のクロックタイミングＬ１（１）で、データメモリ１１０２への書込み信号ｗｒｔがローレベル（ｌｏｗ）になる（図１２（ｂ））。これと共に、図１１の第１のセレクタ（ＳＥＬ１）１１１０がメモリ側ｗｍを選択することにより（図１２（ｃ））、クロックタイミングＬ１（０）でデータメモリ１１０２から出力されていた現在の１サンプリング周期前の遅延データｂ１ｗｍ１が、クロックタイミングＬ１（１）の立ち上がりで第２のレジスタ（ｍ１ｒ）１１０４にラッチされ、その出力がｂ１ｗｍ１となる（図１２（ｄ）→（ｅ））。また、クロックタイミングＬ１（０）でＦＩＲ係数メモリ１１０１から出力されていた２番目のＦＩＲ係数データｂ１（１）が、クロックタイミングＬ１（１）の立ち上がりで第１のレジスタ（ｍ０ｒ）１１０３にラッチされ、その出力がｂ１（１）になる（図１２（ｆ）→（ｇ））。なお、ＦＩＲ係数メモリ１１０１において、クロックタイミングＬ１（１）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、ＦＩＲ係数メモリ１１０１からは、ＦＩＲ係数データ組ｂ１内の３番目のＦＩＲ係数データｂ１（２）が出力される（図１２（ｆ））。また、データメモリ１１０２において、クロックタイミングＬ１（１）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、データメモリ１１０２からは、遅延データ組ｂ１ｗｍ内の２番目（＝現在の２サンプリング周期前）の遅延データｂ１ｗｍ２が出力される（図１２（ｄ））。更に、クロックタイミングＬ１（１）の立ち上がりで、クロックタイミングＬ１（０）で乗算器１１０５から出力されていた乗算結果データ「ｂ１ｉｎ＊ｂ１（０）」が第３のレジスタ（ｍｒ）１１０６にラッチされ、その値が出力ａ０に現れる（図１２（ｈ）→（ｉ））。そして、クロックタイミングＬ１（１）において、第２のレジスタ（ｍ１ｒ）１１０４の出力ｂ１ｗｍ１と、第１のレジスタ（ｍ０ｒ）１１０３の出力ｂ１（

１）とが乗算器１１０５に入力することにより、乗算結果データが「ｂ１ｗｍ１＊ｂ１（１）」となる（図１２（ｅ）、（ｇ）→（ｈ））。クロックタイミングＬ１（１）でのここまでの演算処理が、図１３の乗算処理部８０１（＃１）の動作に対応する。これに加えて、クロックタイミングＬ１（１）で、第２のセレクタ（ＳＥＬ２）１１０９が値０のデータを選択し、その値が出力ａ１に現れる（図１２（ｋ））。そして、クロックタイミングＬ１（１）において、上記第２のセレクタ（ＳＥＬ２）１１０９の出力ａ１と第３のレジスタ（ｍｒ）１１０６の出力ａ０とが加算器１１０７に入力することにより、加算結果データが「０＋ｂ１ｉｎ＊ｂ１（０）」となる（図１２（ｋ）、（ｉ）→（ｊ））。クロックタイミングＬ１（１）のこれらの演算処理が、図１３の累算処理部８０２（＃０）の動作に対応する。

【００８１】

次に、図１２のクロックタイミングＬ１（２）で、データメモリ１１０２への書込み信号ｗｒｔがローレベル（ｌｏｗ）になる（図１２（ｂ））。これと共に、図１１の第１のセレクタ（ＳＥＬ１）１１１０がメモリ側ｗｍを選択することにより（図１２（ｃ））、クロックタイミングＬ１（１）でデータメモリ１１０２から出力されていた現在の２サンプリング周期前の遅延データｂ１ｗｍ２が、クロックタイミングＬ１（２）の立ち上がりで第２のレジスタ（ｍ１ｒ）１１０４にラッチされ、その出力がｂ１ｗｍ２となる（図１２（ｄ）→（ｅ））。また、クロックタイミングＬ１（１）でＦＩＲ係数メモリ１１０１から出力されていた３番目のＦＩＲ係数データｂ１（２）が、クロックタイミングＬ１（２）の立ち上がりで第１のレジスタ（ｍ０ｒ）１１０３にラッチされ、その出力がｂ１（２）になる（図１２（ｆ）→（ｇ））。なお、ＦＩＲ係数メモリ１１０１において、クロックタイミングＬ１（２）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、ＦＩＲ係数メモリ１１０１からは、ＦＩＲ係数データ組ｂ１内の４番目のＦＩＲ係数データｂ１（３）が出力される（図１２（ｆ））。また、データメモリ１１０２において、クロックタイミングＬ１（２）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、データメモリ１１０２からは、遅延データ組ｂ１ｗｍ内の３番目（＝現在の３サンプリング周期前）の遅延データｂ１ｗｍ３が出力される（図１２（ｄ））。更に、クロックタイミングＬ１（２）の立ち上がりで、クロックタイミングＬ１（１）で乗算器１１０５から出力されていた乗算結果データ「ｂ１ｗｍ１＊ｂ１（１）」が第３のレジスタ（ｍｒ）１１０６にラッチされ、その値が出力ａ０に現れる（図１２（ｈ）→（ｉ））。そして、クロックタイミングＬ１（２）において、第２のレジスタ（ｍ１ｒ）１１０４の出力ｂ１ｗｍ２と、第１のレジスタ（ｍ０ｒ）１１０３の出力ｂ１（２）とが乗算器１１０５に入力することにより、乗算結果データが「ｂ１ｗｍ２＊ｂ１（２）」となる（図１２（ｅ）、（ｇ）→（ｈ））。クロックタイミングＬ１（２）でのここまでの演算処理が、図１３の乗算処理部８０１（＃２）の動作に対応する。これに加えて、クロックタイミングＬ１（２）の立ち上がりにおいて、クロックタイミングＬ１（１）で加算器１１０７が出力していた加算結果データ「０＋ｂ１ｉｎ＊ｂ１（０）」が、第４のレジスタ（ａｒ）１１０８にラッチされ、その値が出力ａｒに現れる（図１２（ｊ）→（ｌ））。そして、クロックタイミングＬ１（２）で、第２のセレクタ（ＳＥＬ２）１１０９が第４のレジスタ（ａｒ）１１０８の出力ａｒを選択し、その値が出力ａ１に現れる（図１２（ｋ））。この結果、クロックタイミングＬ１（２）において、上記第２のセレクタ（ＳＥＬ２）１１０９の出力ａ１と第３のレジスタ（ｍｒ）１１０６の出力ａ０とが加算器１１０７に入力することにより、加算結果データが「０＋ｂ１ｉｎ＊ｂ１（０）＋ｂ１ｗｍ１＊ｂ１（１）」となる（図１２（ｌ）、（ｉ）→（ｊ））。クロックタイミングＬ１（２）のこれらの演算処理が、図１３の累算処理部８０２（＃１）の動作に対応する。

【００８２】

次に、図１２のクロックタイミングＬ１（３）で、図１１の第１のセレクタ（ＳＥＬ１）１１１０がメモリ側ｗｍを選択することにより（図１２（ｃ））、クロックタイミングＬ１（２）でデータメモリ１１０２から出力されていた現在の３サンプリング周期前の遅

延データｂ１ｗｍ３が、クロックタイミングＬ１（３）の立ち上がりで第２のレジスタ（ｍ１ｒ）１１０４にラッチされ、その出力がｂ１ｗｍ３となる（図１２（ｄ）→（ｅ））。また、クロックタイミングＬ１（２）でＦＩＲ係数メモリ１１０１から出力されていた４番目のＦＩＲ係数データｂ１（３）が、クロックタイミングＬ１（３）の立ち上がりで第１のレジスタ（ｍ０ｒ）１１０３にラッチされ、その出力がｂ１（３）になる（図１２（ｆ）→（ｇ））。更に、クロックタイミングＬ１（３）の立ち上がりで、クロックタイミングＬ１（２）で乗算器１１０５から出力されていた乗算結果データ「ｂ１ｗｍ２＊ｂ１（２）」が第３のレジスタ（ｍｒ）１１０６にラッチされ、その値が出力ａ０に現れる（図１２（ｈ）→（ｉ））。そして、クロックタイミングＬ１（３）において、第２のレジスタ（ｍ１ｒ）１１０４の出力ｂ１ｗｍ３と、第１のレジスタ（ｍ０ｒ）１１０３の出力ｂ１（３）とが乗算器１１０５に入力することにより、乗算結果データが「ｂ１ｗｍ３＊ｂ１（３）」となる（図１２（ｅ）、（ｇ）→（ｈ））。クロックタイミングＬ１（３）でのここまでの演算処理が、図１３の乗算処理部８０１（＃３）の動作に対応する。これに加えて、クロックタイミングＬ１（３）の立ち上がりにおいて、クロックタイミングＬ１（２）で加算器１１０７が出力していた加算結果データ「０＋ｂ１ｉｎ＊ｂ１（０）＋ｂ１ｗｍ１＊ｂ１（１）」が、第４のレジスタ（ａｒ）１１０８にラッチされ、その値が出力ａｒに現れる（図１２（ｊ）→（ｌ））。そして、クロックタイミングＬ１（３）で、第２のセレクタ（ＳＥＬ２）１１０９が第４のレジスタ（ａｒ）１１０８の出力ａｒを選択し、その値が出力ａ１に現れる（図１２（ｋ））。この結果、クロックタイミングＬ１（３）において、上記第２のセレクタ（ＳＥＬ２）１１０９の出力ａ１と第３のレジスタ（ｍｒ）１１０６の出力ａ０とが加算器１１０７に入力することにより、加算結果データが「０＋ｂ１ｉｎ＊ｂ１（０）＋ｂ１ｗｍ１＊ｂ１（１）＋ｂ１ｗｍ２＊ｂ１（２）」となる（図１２（ｌ）、（ｉ）→（ｊ））。クロックタイミングＬ１（３）のこれらの演算処理が、図１３の累算処理部８０２（＃２）の動作に対応する。一方、ＦＩＲ係数メモリ１１０１において、クロックタイミングＬ１（３）の立ち上がりで読出しアドレスポインタがインクリメントされた結果、ＦＩＲ係数メモリ１１０１においては、次のＦＩＲフィルタ演算処理部ＦＩＲ（２）のためのＦＩＲ係数データ組ｂ２内の１番目のＦＩＲ係数データｂ２（０）が出力される（図１２（ｆ））。また、クロックタイミングＬ１（３）において、図２のミキサ２０６から出力するＲｃｈ効果音入力データ２１０の値が、現在のサンプリング周期のサンプル値ｂ２ｉｎとなり（図１２（ａ））、データメモリ１１０２への書込み信号ｗｒｔがハイレベル（ｈｉｇｈ）となる。これにより、データメモリ１１０２内のＦ（２）に対応する記憶領域内の書込みアドレスポインタが示すアドレスに、サンプル値ｂ２ｉｎが書き込まれる。

【００８３】

次に、図１２のクロックタイミングＬ２（０）で、データメモリ１１０２への書込み信号ｗｒｔがローレベル（ｌｏｗ）になる（図１２（ｂ））。これと共に、図１１の第１のセレクタ（ＳＥＬ１）１１１０が入力側ｉｎを選択することにより（図１２（ｃ））、クロックタイミングＬ１（３）で入力していた現在のサンプリング周期のサンプルｂ２ｉｎが、クロックタイミングＬ２（０）の立ち上がりで第２のレジスタ（ｍ１ｒ）１１０４にラッチされ、その出力がｂ２ｉｎとなる（図１２（ａ）→（ｅ））。また、クロックタイミングＬ１（３）でＦＩＲ係数メモリ１１０１から出力されていた１番目のＦＩＲ係数データｂ２（０）が、クロックタイミングＬ２（０）の立ち上がりで第１のレジスタ（ｍ０ｒ）１１０３にラッチされ、その出力がｂ２（０）になる（図１２（ｆ）→（ｇ））。なお、ＦＩＲ係数メモリ１１０１において、クロックタイミングＬ２（０）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、ＦＩＲ係数メモリ１１０１からは、ＦＩＲ係数データ組ｂ２内の２番目のＦＩＲ係数データｂ２（１）が出力される（図１２（ｆ））。また、データメモリ１１０２において、クロックタイミングＬ２（０）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、データメモリ１１０２からは、遅延データ組ｂ２ｗｍ内の１番目（＝現在の１サンプリング周期前）の遅延データｂ２ｗｍ１が出力される（図１２（ｄ））。更に、クロックタイミングＬ

２（０）の立ち上がりで、クロックタイミングＬ１（３）で乗算器１１０５から出力されていた乗算結果データ「ｂ１ｗｍ３＊ｂ１（３）」が第３のレジスタ（ｍｒ）１１０６にラッチされ、その値が出力ａ０に現れる（図１２（ｈ）→（ｉ））。そして、クロックタイミングＬ２（０）において、第２のレジスタ（ｍ１ｒ）１１０４の出力ｂ２ｉｎと、第１のレジスタ（ｍ０ｒ）１１０３の出力ｂ２（０）とが乗算器１１０５に入力することにより、乗算結果データが「ｂ２ｉｎ＊ｂ２（０）」となる（図１２（ｅ）、（ｇ）→（ｈ））。これにより、ＦＩＲ（１）の演算処理とオーバーラップしてＦＩＲ（２）での演算処理が開始される。これに加えて、クロックタイミングＬ２（０）の立ち上がりにおいて、クロックタイミングＬ１（３）で加算器１１０７が出力していた加算結果データ「０＋ｂ１ｉｎ＊ｂ１（０）＋ｂ１ｗｍ１＊ｂ１（１）＋ｂ１ｗｍ２＊ｂ１（２）」が、第４のレジスタ（ａｒ）１１０８にラッチされ、その値が出力ａｒに現れる（図１２（ｊ）→（ｌ））。そして、クロックタイミングＬ２（０）で、第２のセレクタ（ＳＥＬ２）１１０９が第４のレジスタ（ａｒ）１１０８の出力ａｒを選択し、その値が出力ａ１に現れる（図１２（ｋ））。この結果、クロックタイミングＬ２（０）において、上記第２のセレクタ（ＳＥＬ２）１１０９の出力ａ１と第３のレジスタ（ｍｒ）１１０６の出力ａ０とが加算器１１０７に入力することにより、加算結果データが「０＋ｂ１ｉｎ＊ｂ１（０）＋ｂ１ｗｍ１＊ｂ１（１）＋ｂ１ｗｍ２＊ｂ１（２）＋ｂ１ｗｍ３＊ｂ１（３）」となる（図１２（ｌ）、（ｉ）→（ｊ））。クロックタイミングＬ２（０）のこれらの演算処理が、図１３の累算処理部８０２（＃３）の動作に対応する。

【００８４】

続いて、図１２のクロックタイミングＬ２（１）で、データメモリ１１０２への書込み信号ｗｒｔがローレベル（ｌｏｗ）になる（図１２（ｂ））。これと共に、図１１の第１のセレクタ（ＳＥＬ１）１１１０がメモリ側ｗｍを選択することにより（図１２（ｃ））、クロックタイミングＬ２（０）でデータメモリ１１０２から出力されていた現在の１サンプリング周期前の遅延データｂ２ｗｍ１が、クロックタイミングＬ２（１）の立ち上がりで第２のレジスタ（ｍ１ｒ）１１０４にラッチされ、その出力がｂ２ｗｍ１となる（図１２（ｄ）→（ｅ））。また、クロックタイミングＬ２（０）でＦＩＲ係数メモリ１１０１から出力されていた２番目のＦＩＲ係数データｂ２（１）が、クロックタイミングＬ２（１）の立ち上がりで第１のレジスタ（ｍ０ｒ）１１０３にラッチされ、その出力がｂ２（１）になる（図１２（ｆ）→（ｇ））。なお、ＦＩＲ係数メモリ１１０１において、クロックタイミングＬ２（１）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、ＦＩＲ係数メモリ１１０１からは、ＦＩＲ係数データ組ｂ２内の３番目のＦＩＲ係数データｂ２（２）が出力される（図１２（ｆ））。また、データメモリ１１０２において、クロックタイミングＬ２（１）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、データメモリ１１０２からは、遅延データ組ｂ２ｗｍ内の２番目（＝現在の２サンプリング周期前）の遅延データｂ２ｗｍ２が出力される（図１２（ｄ））。更に、クロックタイミングＬ２（１）の立ち上がりで、クロックタイミングＬ２（０）で乗算器１１０５から出力されていた乗算結果データ「ｂ２ｉｎ＊ｂ２（０）」が第３のレジスタ（ｍｒ）１１０６にラッチされ、その値が出力ａ０に現れる（図１２（ｈ）→（ｉ））。そして、クロックタイミングＬ２（１）において、第２のレジスタ（ｍ１ｒ）１１０４の出力ｂ２ｗｍ１と、第１のレジスタ（ｍ０ｒ）１１０３の出力ｂ２（１）とが乗算器１１０５に入力することにより、乗算結果データが「ｂ２ｗｍ１＊ｂ２（１）」となる（図１２（ｅ）、（ｇ）→（ｈ））。これに加えて、クロックタイミングＬ２（１）で、第２のセレクタ（ＳＥＬ２）１１０９が値０のデータを選択し、その値が出力ａ１に現れる（図１２（ｋ））。そして、クロックタイミングＬ２（１）において、上記第２のセレクタ（ＳＥＬ２）１１０９の出力ａ１と第３のレジスタ（ｍｒ）１１０６の出力ａ０とが加算器１１０７に入力することにより、加算結果データが「０＋ｂ２ｉｎ＊ｂ２（０）」となる（図１２（ｋ）、（ｉ）→（ｊ））。更に、クロックタイミングＬ２（１）の立ち上がりにおいて、クロックタイミングＬ２（０）で加算器１１０７が出力していた加算結果データ「０＋ｂ１ｉｎ＊ｂ１（０）＋ｂ１ｗｍ１＊ｂ１（１）＋ｂ１ｗ

ｍ２＊ｂ１（２）＋ｂ１ｗｍ３＊ｂ１（３）」が、第４のレジスタ（ａｒ）１１０８にラッチされ、その値が出力ａｒに現れる（図１２（ｊ）→（ｌ））。ＦＩＲ（１）に対応する連続区間の最後のクロックタイミングＬ１（３）から２クロック後のクロックタイミングＬ２（１）のタイミングで、第４のレジスタ（ａｒ）１１０８に現れた出力データ「０＋ｂ１ｉｎ＊ｂ１（０）＋ｂ１ｗｍ１＊ｂ１（１）＋ｂ１ｗｍ２＊ｂ１（２）＋ｂ１ｗｍ３＊ｂ１（３）」が、ＦＩＲ（１）の畳込み結果データ＝図３のＬｃｈ効果音出力データ２１１として図３の加算部３０５を介して出力される。この出力は、図１３の累算処理部８０２（＃３）から出力される畳込み結果データに対応する。

【００８５】

次に、図１２のクロックタイミングＬ２（２）で、データメモリ１１０２への書込み信号ｗｒｔがローレベル（ｌｏｗ）になる（図１２（ｂ））。これと共に、図１１の第１のセレクタ（ＳＥＬ１）１１１０がメモリ側ｗｍを選択することにより（図１２（ｃ））、クロックタイミングＬ２（１）でデータメモリ１１０２から出力されていた現在の２サンプリング周期前の遅延データｂ２ｗｍ２が、クロックタイミングＬ２（２）の立ち上がりで第２のレジスタ（ｍ１ｒ）１１０４にラッチされ、その出力がｂ２ｗｍ２となる（図１２（ｄ）→（ｅ））。また、クロックタイミングＬ２（１）でＦＩＲ係数メモリ１１０１から出力されていた３番目のＦＩＲ係数データｂ２（２）が、クロックタイミングＬ２（２）の立ち上がりで第１のレジスタ（ｍ０ｒ）１１０３にラッチされ、その出力がｂ２（２）になる（図１２（ｆ）→（ｇ））。なお、ＦＩＲ係数メモリ１１０１において、クロックタイミングＬ２（２）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、ＦＩＲ係数メモリ１１０１からは、ＦＩＲ係数データ組ｂ２内の４番目のＦＩＲ係数データｂ２（３）が出力される（図１２（ｆ））。また、データメモリ１１０２において、クロックタイミングＬ２（２）の立ち上がりで読出しアドレスポインタがインクリメントされることにより、データメモリ１１０２からは、遅延データ組ｂ２ｗｍ内の３番目（＝現在の３サンプリング周期前）の遅延データｂ２ｗｍ３が出力される（図１２（ｄ））。更に、クロックタイミングＬ２（２）の立ち上がりで、クロックタイミングＬ２（１）で乗算器１１０５から出力されていた乗算結果データ「ｂ２ｗｍ１＊ｂ２（１）」が第３のレジスタ（ｍｒ）１１０６にラッチされ、その値が出力ａ０に現れる（図１２（ｈ）→（ｉ））。そして、クロックタイミングＬ２（２）において、第２のレジスタ（ｍ１ｒ）１１０４の出力ｂ２ｗｍ２と、第１のレジスタ（ｍ０ｒ）１１０３の出力ｂ２（２）とが乗算器１１０５に入力することにより、乗算結果データが「ｂ２ｗｍ２＊ｂ２（２）」となる（図１２（ｅ）、（ｇ）→（ｈ））。これに加えて、クロックタイミングＬ２（２）の立ち上がりにおいて、クロックタイミングＬ２（１）で加算器１１０７が出力していた加算結果データ「０＋ｂ２ｉｎ＊ｂ２（０）」が、第４のレジスタ（ａｒ）１１０８にラッチされ、その値が出力ａｒに現れる（図１２（ｊ）→（ｌ））。そして、クロックタイミングＬ２（２）で、第２のセレクタ（ＳＥＬ２）１１０９が第４のレジスタ（ａｒ）１１０８の出力ａｒを選択し、その値が出力ａ１に現れる（図１２（ｋ））。この結果、クロックタイミングＬ２（２）において、上記第２のセレクタ（ＳＥＬ２）１１０９の出力ａ１と第３のレジスタ（ｍｒ）１１０６の出力ａ０とが加算器１１０７に入力することにより、加算結果データが「０＋ｂ２ｉｎ＊ｂ２（０）＋ｂ２ｗｍ１＊ｂ２（１）」となる（図１２（ｌ）、（ｉ）→（ｊ））。

【００８６】

以上のようにして、データの矛盾を生じることなく、ＦＩＲフィルタ演算処理部ＦＩＲ（１）での演算処理及び出力に続いて、ＦＩＲフィルタ演算処理部ＦＩＲ（２）での演算処理及び出力を、サンプリング周期を細分化した周期のクロックに同期した時分割処理により実行することが可能となり、それぞれ個別にフィルタ次数が設定された複数のＦＩＲフィルタ演算処理をサンプリング周期毎に遅延無く実行することが可能となる。これにより、例えばインパルス応答データが短い場合などにおいて、ブロックサイズを小さくし、ＦＩＲ資源を別のフィルタリング処理に用いることが可能となる。また、インパルス応答長が長い場合に、図３のＣＯＮＶ演算処理部３０４で用いられる畳込み用の係数データサ

イズも大きくなるため、例えば図２の効果付与部１０５であるＤＳＰ内に実装される図３のＲＡＭ３０６を共通に用いる場合、前述したＣＯＮＶ演算処理部３０４でのブロックサイズＮが固定ならば、メモリ帯域に応じたブロックサイズ調整を行うことができないが、本実施形態の構成により、ブロックサイズが調整可能となるので、効果付与部１０５内での処理を最適化することが可能となる。

【００８７】

図３のＦＩＲフィルタ演算処理部３０３のためにはフィルタ次数＝２Ｎの実係数データを、また図３のＣＯＮＶ演算処理部３０４のためには上記２Ｎ個以降の実係数データをブロック毎にＦＦＴ演算した結果を、それぞれ図３のＲＡＭ３０６に記憶しておけばよいが、ブロックサイズＮが可変となる場合、この全てを記憶しておくとメモリ消費が増える。そこで、まずインパルス応答データが係数データとして図３のＲＡＭ３０６メモリに記憶され、先のリバーブ時間や機器のシステム条件に応じてブロックサイズＮが決まれば、ＲＡＭ３０６に記憶されているインパルス応答データの始めの２Ｎ個がＦＩＲフィルタ演算処理部３０３に供給され、上記２Ｎ個以降のデータをＦＦＴ演算で変換したものがＲＡＭ３０６に展開されてＣＯＮＶ演算処理部３０４に供給されればよい。ＦＩＲフィルタ演算処理部３０３が入力データのＮサンプル分を処理している間に、上記ＲＡＭ３０６への展開が完了すれば、ＣＯＮＶ演算処理部３０４での処理に影響は生じない。

【００８８】

また、予め最適となる設定を想定して各ブロックサイズが決定され、このブロックサイズの情報とＣＯＮＶ演算処理部３０４用のＦＦＴ変換データがＲＡＭ３０６に配置され、畳込み処理の実行時にブロックサイズＮが決まった場合にこの決定値がＲＡＭ３０６に記憶されたブロックサイズ情報と比較されることにより、ブロックサイズが異なる場合のみ係数のＦＦＴ変換処理が行われるようにしてもよい。

【００８９】

図１４は、上述したようにブロックサイズを動的に変更可能な図２の残響・共鳴装置２１１を実現するために図１のＣＰＵ１０１が実行する全体動作の制御処理例を示すメインフローチャートである。この制御処理は、ＣＰＵ１０１がＲＯＭ１０２からＲＡＭ１０３にロードさせた制御処理プログラムを実行する動作である。

【００９０】

図１の電子楽器１００において、操作子１０８の電源スイッチがオンされた後、ＣＰＵ１０１は、図１４のメインフローチャートで示される制御処理の実行を開始する。ＣＰＵ１０１はまず、図１のＲＡＭ１０３の記憶内容、音源（ＴＧ）１０４の状態、効果付加装置１００の状態等の初期化を行う（ステップＳ１４０１）。その後、ＣＰＵ１０１は、電源スイッチがオフされるまで、ステップＳ１４０２からＳ１４０７の一連の処理を繰り返し実行される。

【００９１】

上記繰返し処理において、ＣＰＵ１０１はまず、スイッチ処理を実行する（ステップＳ１４０２）。ここでは、ＣＰＵ１０１は、図１の操作子１０８の操作状態を検出する。

【００９２】

次に、ＣＰＵ１０１は、押鍵検出処理を実行する（ステップＳ１４０３）。ここでは、ＣＰＵ１０１は、図１の鍵盤１０６の押鍵状態を検出する。

【００９３】

次に、ＣＰＵ１０１は、ペダル検出処理を実行する（ステップＳ１４０４）。ここでは、ＣＰＵ１０１は、図１のペダル１０７の操作状態を検出する。

【００９４】

次に、ＣＰＵ１０１は、残響・共鳴更新処理を実行する（ステップＳ１４０５）。ここでは、ＣＰＵ１０１は、ステップＳ１４０２での残響・共鳴効果付与のための操作子１０８の操作状態の検出結果と、ステップＳ１４０４でのペダル１０７の操作状態の検出結果とに基づいて、効果付与部１０５に、音源（ＴＧ）１０４が生成する図２のＬｃｈ効果音入力データ２０９及びＲｃｈ効果音入力データ２１０に対して、図３の残響・共鳴装置２

１１（Ｌｃｈ）及び残響・共鳴装置２１１（Ｒｃｈ）による残響・共鳴効果の付与を実行させる。

【００９５】

次に、ＣＰＵ１０１は、その他の処理を実行する（ステップＳ１４０６）。ここでは、ＣＰＵ１０１は、例えば楽音エンベロープの制御処理等を実行する。

【００９６】

そして、ＣＰＵ１０１は、発音処理を実行する（ステップＳ１４０７）。ここでは、ＣＰＵ１０１は、ステップＳ１４０３での押鍵検出処理での鍵盤１０６の押鍵（又は離鍵）の状態に基づいて、音源（ＴＧ）１０４に対して発音指示を行う。

【００９７】

図１５は、図１４のステップＳ１４０５の残響・共鳴更新処理の詳細処理例を示すフローチャートである。

【００９８】

まず、ＣＰＵ１０１は、ステップＳ１４０２での残響・共鳴効果付与のための操作子１０８の操作状態の検出結果に基づいて、図１のＲＯＭ１０２を参照することにより、効果のタイプ情報を取得する（ステップＳ１５０１）。

【００９９】

次に、ＣＰＵ１０１は、操作子１０８によって効果のタイプが指定されたか否かを判定する（ステップＳ１５０２）。

【０１００】

ステップＳ１５０２の判定がＹＥＳならば、ＣＰＵ１０１は、図３の残響・共鳴装置２１１（Ｌｃｈ、Ｒｃｈ）における前述したブロックサイズＢＬ（＝Ｎ）の更新処理を実行する（ステップＳ１５０３）。ステップＳ１５０２の判定がＮＯならば、ＣＰＵ１０１は、ステップＳ１５０３の処理はスキップする。その後、ＣＰＵ１０１は、図１５のフローチャートで示される残響・共鳴更新処理を終了して、図１４のメインフローチャートの繰返し処理に戻る。

【０１０１】

図１６は、図１５のステップＳ１５０３のブロック更新処理の詳細処理例（その１）を示すフローチャートである。

【０１０２】

ＣＰＵ１０１はまず、図１５のステップＳ１５０１で取得した効果のタイプを畳込テーブル番号として、ＲＯＭ１０２に記憶されている畳込テーブルを参照し、インパルス応答データのインパルス応答（ＩＲ：Ｉｍｐｕｌｓｅ　Ｒｅｓｐｏｎｓｅ）長を取得する（ステップＳ１６０１）。

【０１０３】

次に、ＣＰＵ１０１は、インパルス応答長（ＩＲ長）が、或る短い閾値よりも短いか否かを判定する（ステップＳ１６０２）。

【０１０４】

ステップＳ１６０２の判定がＹＥＳならば、ＣＰＵ１０１は、ブロックサイズＢＬを所定の短い値に設定する（ステップＳ１６０３）。

【０１０５】

ステップＳ１６０２の判定がＮＯならば、ＣＰＵ１０１は、インパルス応答長が、或る中程度の閾値よりも短いか否かを判定する（ステップＳ１６０４）。

【０１０６】

ステップＳ１６０４の判定がＹＥＳならば、ＣＰＵ１０１は、ブロックサイズＢＬを所定の中程度の値に設定する（ステップＳ１６０５）。

【０１０７】

ステップＳ１６０４の判定がＮＯならば、ＣＰＵ１０１は、ブロックサイズＢＬを所定の長い値に設定する（ステップＳ１６０５）。

【０１０８】

ステップＳ１６０３，Ｓ１６０５、又はＳ１６０６の処理の後、ＣＰＵ１０１は、ＲＯＭ１０２（又はＲＯＭ１０２からロードされ効果付加装置１００のＤＳＰ内の図３のＲＡＭ３０６）に記憶されているインパルス応答データの始めの２＊ＢＬ個をＦＩＲフィルタ演算処理部３０３を構成する図１１のＦＩＲ係数メモリ１１０１に記憶させて更新する（ステップＳ１６０７）。

【０１０９】

続いて、ＣＰＵ１０１は、上記インパルス応答データの２＊ＢＬ個以降のデータをブロックＢＬ毎にＦＦＴ演算で変換し、図３のＲＡＭ３０６に展開させてＣＯＮＶ演算処理部３０４に供給する（ステップＳ１６０８）。その後、ＣＰＵ１０１は、図１６のフローチャートで示されるブロック更新処理を終了し、図１５のフローチャートの処理に戻る。

【０１１０】

図１７は、図１５のステップＳ１５０３のブロック更新処理の詳細処理例（その２）を示すフローチャートである。

【０１１１】

ＣＰＵ１０１はまず、システム状態（メモリの使用状態等）から、例えばＲＯＭ１０２内のテーブルを参照する等により、図３のＣＯＮＶ演算処理部３０４に設定可能なブロックサイズＢ１を取得する（ステップＳ１７０１）。

【０１１２】

次に、ＣＰＵ１０１は、図１５のステップＳ１５０１で取得した効果のタイプを畳込テーブル番号として、ＲＯＭ１０２に記憶されている畳込テーブルを参照し、ブロックサイズＢ２を取得する（ステップＳ１７０２）。

【０１１３】

ＣＰＵ１０１は、ステップＳ１７０１で取得したシステム状態に基づくブロックサイズＢ１が、ステップＳ１７０２で取得した効果のタイプに基づくブロックサイズＢ２と等しいか否かを判定する（ステップＳ１７０３）。

【０１１４】

ステップＳ１７０３の判定がＹＥＳならば、図３のＦＩＲフィルタ演算処理部３０３及びＣＯＮＶ演算処理部３０４の係数データの更新は行わずに、同じデータのセット又は参照先の設定のみを行う（ステップＳ１７０４）。

【０１１５】

ステップＳ１７０３の判定がＮＯならば、図１６のステップＳ１６０７及びＳ１６０８の場合と同様にして、係数の更新を行う（ステップＳ１７０５）。

【０１１６】

ステップＳ１７０４又はＳ１７０５の処理の後、ＣＰＵ１０１は、図１６のフローチャートで示されるブロック更新処理を終了し、図１５のフローチャートの処理に戻る。

【０１１７】

以上説明したようにして、本実施形態によれば、フィルタ次数を柔軟に変更可能で、同時に複数種類のＦＩＲフィルタ演算処理を実行可能なフィルタ演算処理装置を実現することが可能となる。

【０１１８】

また、本実施形態によるＦＩＲフィルタ演算処理とＣＯＮＶ演算処理とを組み合わせて畳込み演算を行う場合において、発音設定やシステムの状態等に応じてブロックサイズを柔軟に変更しながら、ブロック分の遅延を生じさせることなく、楽音信号にインパルス応答を付与することができるため、電子楽器などの用途において、応答性と再現性の高い効果を付与することが可能となる。

【０１１９】

以上の実施形態に関して、更に以下の付記を開示する。

（付記１）

ＦＩＲフィルタ演算処理に用いる複数のＦＩＲ係数データが記憶されたＦＩＲ係数記憶部と、

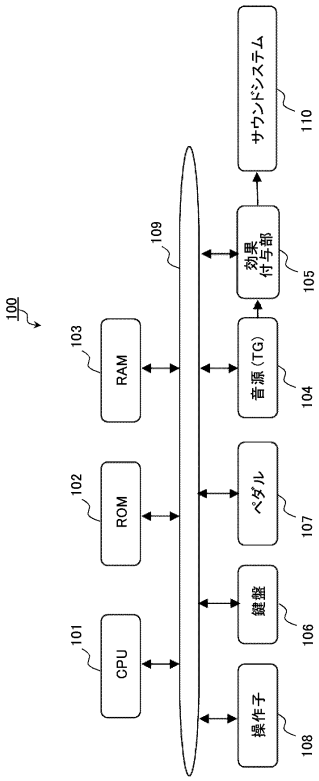
前記ＦＩＲフィルタ演算処理の対象となる複数の入力データを記憶する入力データ記憶部と、
前記ＦＩＲ係数記憶部に記憶された前記複数のＦＩＲ係数データと前記入力データ記憶部に記憶された前記複数の入力データとに基づいて、指定されたフィルタ次数に対応する回数だけ、各ＦＩＲ係数データと各入力データとを乗算しながら累算していくＦＩＲフィルタ演算処理を実行させる制御部と、
を有することを特徴とするフィルタ演算処理装置。
（付記２）
前記制御部は、異なる複数のフィルタ次数を選択して指定することが可能であり、
前記ＦＩＲ係数記憶部は、前記制御部が指定可能なフィルタ次数個分のＦＩＲ係数データを記憶し、
前記入力データ記憶部は、前記制御部が指定可能なフィルタ次数に対応するサンプル数の入力データを記憶する、
ことを特徴とする付記１に記載のフィルタ演算処理装置。
（付記３）
前記ＦＩＲ係数記憶部は、ＦＩＲフィルタ演算処理のフィルタ次数個分のＦＩＲ係数データ組を、前記フィルタ次数が可変の１種類以上の前記ＦＩＲフィルタ演算処理毎に記憶し、
前記入力データ記憶部は、現在の１サンプル前から（前記フィルタ次数－１）サンプル前までの前記ＦＩＲフィルタ演算処理に対する入力データ組を遅延データ組として、前記ＦＩＲフィルタ演算処理毎に記憶し、
１次分の前記ＦＩＲフィルタ演算処理を実行するフィルタ演算部を更に備え、
前記制御部は、各サンプリング周期内で、前記ＦＩＲフィルタ演算処理毎に、前記フィルタ次数に対応した時分割長の時分割処理により、前記フィルタ演算部に、前記ＦＩＲ係数記憶部のＦＩＲ係数データと現在の入力データ又は前記入力データ記憶部の遅延データとを順次入力させながら１次分の前記ＦＩＲフィルタ演算処理を繰り返し実行させて出力データを出力させる、
ことを特徴とする付記１又は２に記載のフィルタ演算処理装置。
（付記４）
前記制御部は、前記サンプリング周期内で、当該周期を細分化した周期を有するクロックでカウントされ前記フィルタ次数を有する前記ＦＩＲフィルタ演算処理を実行可能な長さの連続区間を個別に割り当て、当該連続区間で、前記フィルタ演算部に、前記クロックに同期させて、前記ＦＩＲフィルタ演算処理に対応する前記ＦＩＲ係数記憶部のＦＩＲ係数データと現在の入力データ又は前記入力データ記憶部の遅延データとを順次入力させながら、前記ＦＩＲ乗算／累算処理を繰り返し実行させ、前記ＦＩＲフィルタ演算処理に対応する前記サンプリング周期の出力データを出力させる制御を、前記ＦＩＲフィルタ演算処理毎に実行する、ことを特徴とする付記３に記載のフィルタ演算処理装置。
（付記５）
前記フィルタ演算部は、
前記クロックに同期して、前記ＦＩＲ係数記憶部が出力するＦＩＲ係数データを保持する第１のレジスタと、
現在のサンプリング周期の入力データ又は前記入力データ記憶部が出力する遅延データの何れかを選択する第１のセレクタと、
前記クロックに同期して、前記第１のセレクタが出力するデータを保持する第２のレジスタと、
前記第１のレジスタが出力する前記ＦＩＲ係数データと前記第２のレジスタが出力するデータとを乗算する乗算器と、
前記クロックに同期して、前記乗算器が出力する乗算結果データを保持する第３のレジスタと、
前記第３のレジスタが出力する前記乗算結果データと現在までの累算データとを加算す

る加算器と、
前記クロックに同期して、前記加算器が出力する加算結果データを保持する第４のレジスタと、
値がゼロのデータ又は前記第４のレジスタが出力する加算結果データの何れかを選択し、当該選択されたデータを前記累算データとして前記加算器にフィードバックする第２のセレクタと、
を含むことを特徴とする付記３又は４に記載のフィルタ演算処理装置。
（付記６）
音響効果音のインパルス応答データを原音に畳み込むことにより前記原音に音響効果を付与する効果付与装置であって、
付記１乃至３の何れかに記載のフィルタ演算処理装置の前記ＦＩＲフィルタ演算処理により、前記インパルス応答データの前半データ部分を前記サンプリング周期単位の時間域処理により前記原音に畳み込む時間域畳込み部と、
前記インパルス応答データの後半データ部分を高速フーリエ変換演算を用いたブロック単位の周波数域処理により前記原音に畳み込む周波数域畳込み部と、
を有することを特徴とする効果付与装置。
（付記７）
前記インパルス応答データの前半データ部分と後半データ部分とを任意に設定する、
ことを特徴とする付記６に記載の効果付与装置。
（付記８）
前記ブロックのブロックサイズを決定し、前記ＦＩＲフィルタ演算処理における前記フィルタ次数及び前記ＦＦＴ演算処理装置のＦＦＴ点数を前記ブロックサイズの２倍に設定する設定部を更に有する、
ことを特徴とする付記６又は７に記載の効果付与装置。
（付記９）
前記設定部は、前記インパルス応答データの時間長に応じて前記ブロックサイズを決定する、ことを特徴とする付記８に記載の効果付与装置。
（付記１０）
前記設定部は、前記原音の発音設定に応じて前記ブロックサイズを決定する、ことを特徴とする付記８に記載の効果付与装置。
（付記１１）
前記設定部は、
前記インパルス応答データの時間域の係数データを記憶し、
前記ブロックサイズを決定した後、前記インパルス応答データの係数データから前記ＦＩＲフィルタ演算処理で用いる前記ブロックサイズの２倍のフィルタ次数の前記ＦＩＲ係数データを除いた残りの係数データを、高速フーリエ変換演算により周波数域の畳込み用係数データに変換し、メモリ展開する、
ことを特徴とする付記８乃至１０の何れかに記載の効果付与装置。
（付記１２）
前記設定部は、
前記周波数域の畳込み用係数データに、前記決定したブロックサイズを示すデータを付与して記憶し、
その後、前記ブロックサイズを再度決定した場合には、前記再度決定したブロックサイズが、前記記憶していたブロックサイズと異なる場合のみ、前記周波数域の畳込み用係数データを高速フーリエ変換により更新する、
ことを特徴とする付記１１に記載の効果付与装置。
（付記１３）
ＦＩＲフィルタ演算処理に用いる複数のＦＩＲ係数データが記憶されたＦＩＲ係数記憶部から順次ＦＩＲ係数データを読み出し、
前記ＦＩＲフィルタ演算処理の対象となる複数の入力データを記憶する入力データ記憶

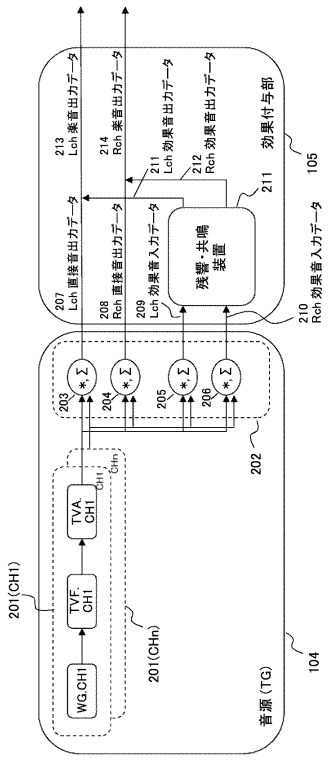
部から順次入力データを読み出し、
前記ＦＩＲフィルタ演算処理におけるフィルタ次数を指定し、
順次読み出されるＦＩＲ係数データと、順次読み出される入力データに対して、指定されたフィルタ次数に対応する回数だけ、各ＦＩＲ係数データと各入力データとを乗算しながら累算していくＦＩＲフィルタ演算処理を繰り返し実行させる、
ことを特徴とするフィルタ演算方法。
（付記１４）
音響効果音のインパルス応答データを原音に畳み込むことにより前記原音に音響効果を付与する場合に、
前記ＦＩＲフィルタ演算処理により、前記インパルス応答データの前半データ部分を前記サンプリング周期単位の時間域処理により前記原音に畳み込み、
前記インパルス応答データの後半データ部分を高速フーリエ変換演算を用いたブロック単位の周波数域処理により前記原音に畳み込む、
ことを特徴とする付記１３に記載のフィルタ演算方法。
（付記１５）
前記インパルス応答データの前半データ部分と後半データ部分とを任意に設定する、
ことを特徴とする付記１３又は１４に記載のフィルタ演算方法。
【符号の説明】
【０１２０】
１００　電子楽器
１０１　ＣＰＵ
１０２　ＲＯＭ
１０３、３０６　ＲＡＭ
１０４　音源（ＴＧ）
１０５　効果付与部
１０６　鍵盤
１０７　ペダル
１０８　操作子
１０９　システムバス
１１０　サウンドシステム
２０１　楽音生成部
２０２　ミキシング部
２０３、２０４、２０５、２０６　ミキサ
２０７　Ｌｃｈ直接音出力データ
２０８　Ｒｃｈ直接音出力データ
２０９　Ｌｃｈ効果音入力データ
２１０　Ｒｃｈ効果音入力データ
２１１　Ｌｃｈ効果音出力データ
２１２　Ｒｃｈ効果音出力データ
２１３　Ｌｃｈ楽音出力データ
２１４　Ｒｃｈ楽音出力データ
３０３　ＦＩＲフィルタ演算処理部
３０４　ＣＯＮＶ演算処理部
３０５　加算部
８００、１１００　フィルタ演算部
８０１　乗算処理部
８０２　累算処理部
８０３　遅延処理部
１１０１　ＦＩＲ係数メモリ
１１０２　データメモリ

１１０３　第１のレジスタ（ｍ０ｒ）
１１０４　第２のレジスタ（ｍ１ｒ）
１１０５　乗算器
１１０６　第３のレジスタ（ｍｒ）
１１０７　加算器
１１０８　第４のレジスタ（ａｒ）
１１０９　第２のセレクタ（ＳＥＬ２）
１１１０　第１のセレクタ（ＳＥＬ１）
１１１１　入力データ

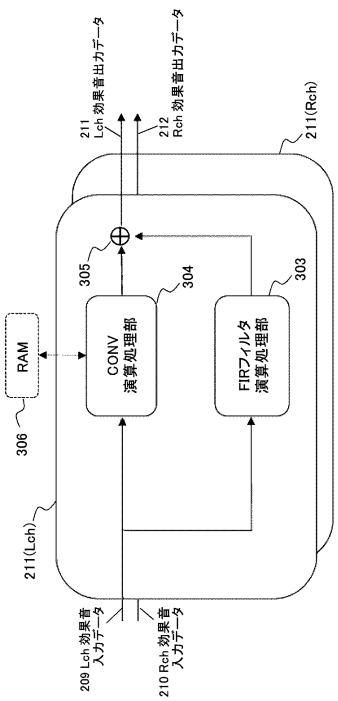
【図１】



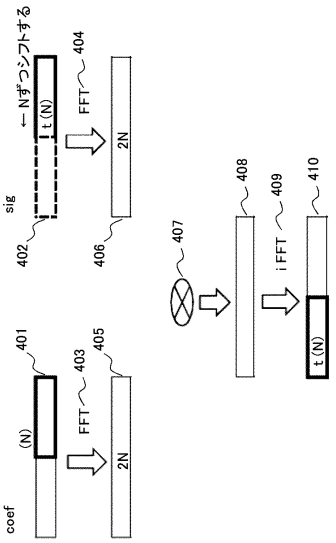
【図２】



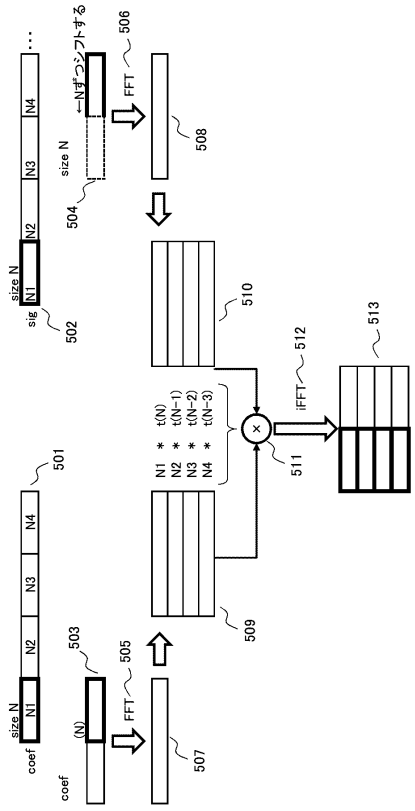
【図３】


211 Lch 効果音出力データ
212 Rch 効果音出力データ
211(Rch)
305
304
303
RAM
306
CONV 演算処理部
FIRフィルタ 演算処理部
211(Lch)
209 Lch 効果音 入力データ
210 Rch 効果音 入力データ

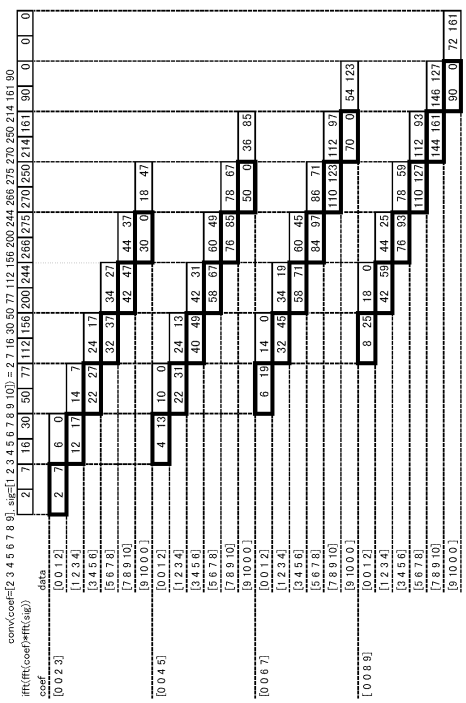
【図４】


←Nずつシフトする
t (N)
FFT
404
sig
402
2N
406
407
408
409
i FFT
410
t (N)
401
(N)
FFT
403
405
2N
coef

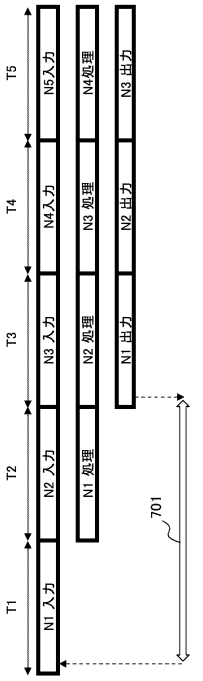
【図５】


size N
N1
N2
N3
N4
coef
(N)
501
503
505
FFT
507
509
N1 * t(N)
N2 * t(N-1)
N3 * t(N-2)
N4 * t(N-3)
511
iFFT
512
513
510
508
506
←Nずつシフトする
FFT
504
size N
N1
N2
N3
N4
sig
502

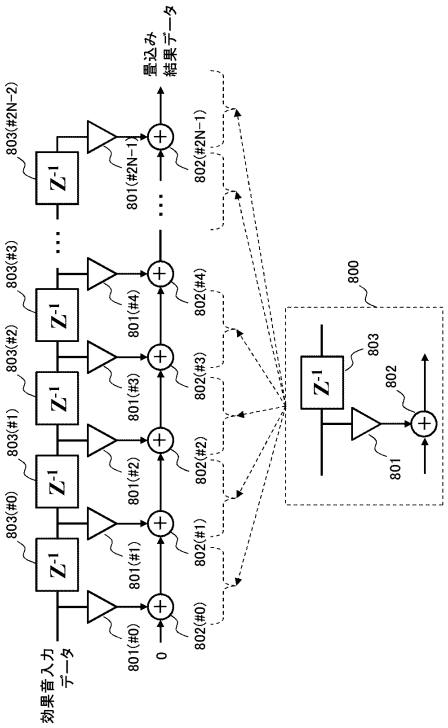
【図６】



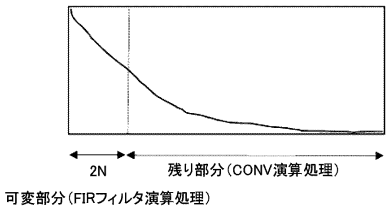
【図７】



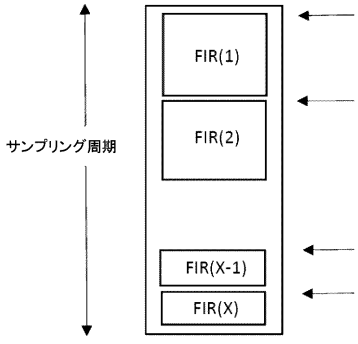
【図８】



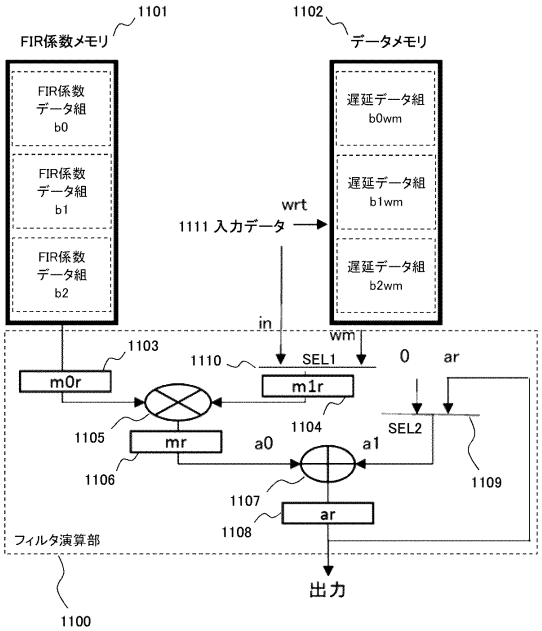
【図９】



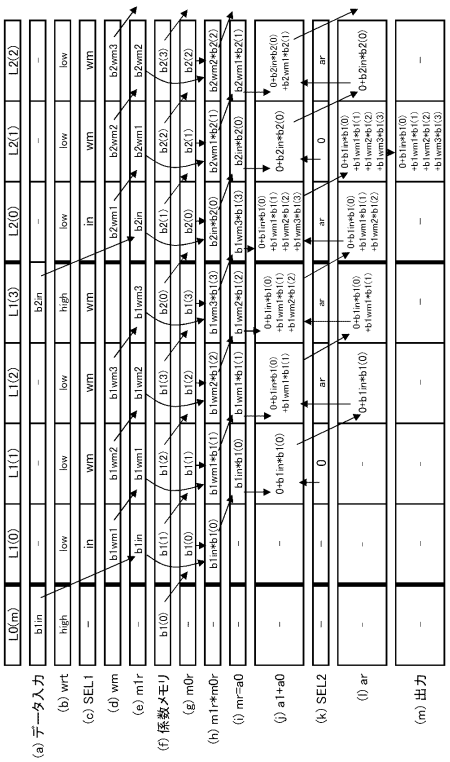
【図１０】



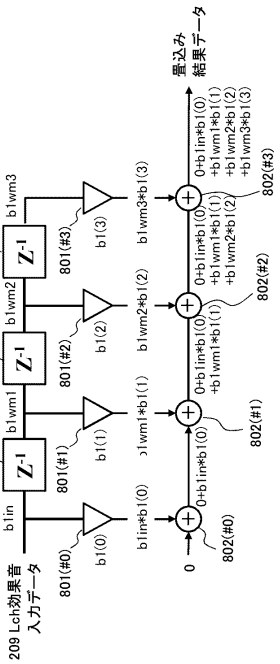
【図１１】



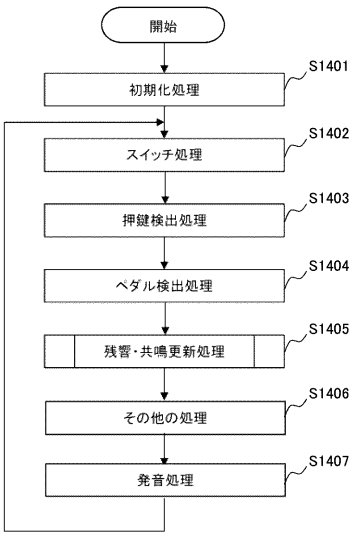
【図１２】



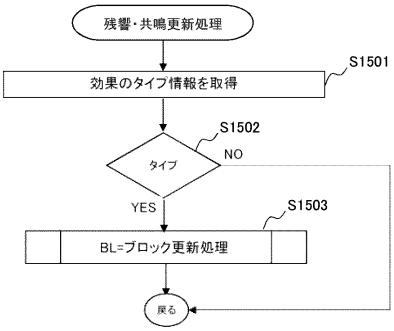
【図１３】



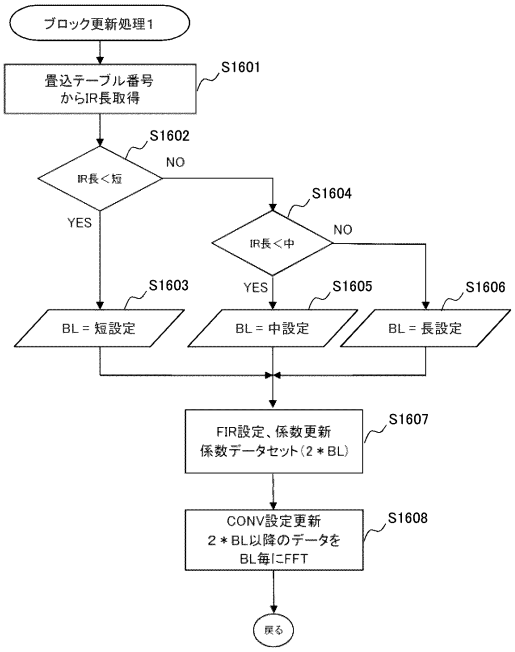
【図１４】



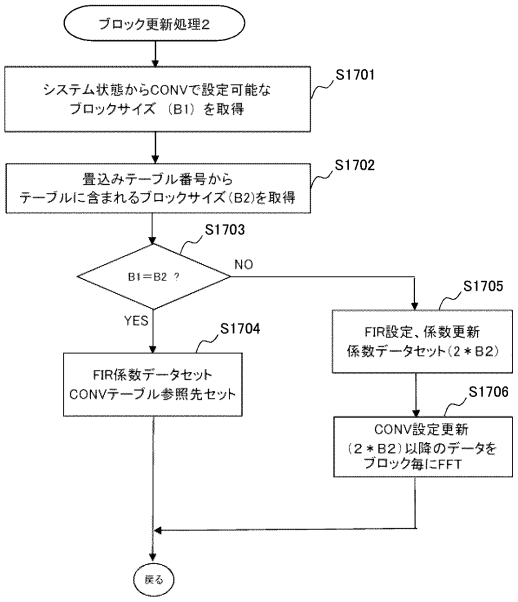
【図１５】



【図１６】


ブロック更新処理1
畳込テーブル番号
からIR長取得
S1601
S1602
IR長＜短
NO
YES
S1604
IR長＜中
NO
YES
S1603
BL = 短設定
S1605
BL = 中設定
S1606
BL = 長設定
FIR設定、係数更新
係数データセット（2＊BL）
S1607
CONV設定更新
2＊BL以降のデータを
BL毎にFFT
S1608
戻る

【図１７】


ブロック更新処理2
システム状態からCONVで設定可能な
ブロックサイズ （B1）を取得
S1701
畳込みテーブル番号から
テーブルに含まれるブロックサイズ（B2)を取得
S1702
S1703
B1＝B2 ？
NO
YES
S1704
FIR係数データセット
CONVテーブル参照先セット
S1705
FIR設定、係数更新
係数データセット（2＊B2）
S1706
CONV設定更新
（2＊B2）以降のデータを
ブロック毎にFFT
戻る

フロントページの続き

(56)参考文献　特開昭６０－１７７７１３（ＪＰ，Ａ）
特開平０９－１８５３７３（ＪＰ，Ａ）
特開２００８－１９７２８４（ＪＰ，Ａ）
特開昭６４－０６８７９９（ＪＰ，Ａ）
特開２０００－２６７６８２（ＪＰ，Ａ）
米国特許第０５８１４７５０（ＵＳ，Ａ）
山口 晶大，体験版シミュレータや評価キットで楽しみながら学ぶ これならわかる！ＤＳＰ入門，トランジスタ技術，日本，CQ出版株式会社，２００２年１０月　１日，第39巻，第10号，第147-186ページ
WILLIAM G. GARDNER, Efficient Convolution without Input-Output Delay, J. Audio Eng. Soc., Vol.43, No.3, １９９５年

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ１０Ｈ　　１／００－　７／１２
Ｇ１０Ｋ　１５／００－１５／０４
Ｈ０３Ｈ　１７／０６
Ｈ０３Ｈ　１７／０８