

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6641965号  
(P6641965)

(45) 発行日 令和2年2月5日(2020.2.5)

(24) 登録日 令和2年1月8日(2020.1.8)

(51) Int.Cl.

F 1

G 1 O H 1/00 (2006.01)  
G 1 O H 7/00 (2006.01)G 1 O H 1/00 1 O 2 Z  
G 1 O H 7/00

請求項の数 9 (全 42 頁)

(21) 出願番号 特願2015-243506 (P2015-243506)  
 (22) 出願日 平成27年12月14日 (2015.12.14)  
 (65) 公開番号 特開2017-111204 (P2017-111204A)  
 (43) 公開日 平成29年6月22日 (2017.6.22)  
 審査請求日 平成30年12月6日 (2018.12.6)

(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 佐藤 博毅  
 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】音響処理装置、音響処理方法、プログラム、及び電子楽器

## (57) 【特許請求の範囲】

## 【請求項 1】

第1クロックを循環的に第1のカウント値までカウントする第1のカウント処理と、前記第1のカウント処理によるカウントが前記第1のカウント値に達する毎に音響効果が付加された音響信号を出力する音響信号出力処理と、を実行する第1の処理部と、

第2クロックを循環的に第2のカウント値までカウントする第2のカウント処理と、前記第2のカウント処理によるカウントが前記第2のカウント値に達する毎に自動演奏を行なせる自動演奏制御処理と、前記第2のカウント処理によるカウントが前記第2のカウント値に達したタイミングと前記第1のカウント処理によるカウントが前記第1のカウント値に達したタイミングとの差分時間を循環的に検出する検出処理と、前記検出処理により検出される差分時間を小さくするように前記第2のカウント値を変更する補正処理と、を実行する第2の処理部と、

を備えた音響処理装置。

## 【請求項 2】

前記音響処理装置さらに、前記自動演奏のテンポを指定するテンポ指定部を有し、

前記第1のカウント値及び前記第2のカウント値は、前記指定されたテンポに対応して決定される、請求項1に記載の音響処理装置。

## 【請求項 3】

前記音響処理装置はさらに、複数種の自動演奏のテンポ夫々に対応して前記第1及び第2のカウント値を記憶するテーブルを有し、

前記第2の処理部はさらに、指定された自動演奏のテンポに対応する前記第1及び第2のカウント値を前記テーブルから読み出すとともに、前記第1のカウント値を前記第1の処理部に設定するカウント値設定処理を実行する、請求項1または2に記載の音響処理装置。

#### 【請求項4】

前記第1の処理部は、前記第1のカウント処理によりカウントされたカウント値が前記第1のカウント値に達するまでの時間の自然数倍及び所定自然数分の1倍のいずれか一方の関係の時間に同期したタイミングで、供給される音響信号に対して音響効果を付加する処理を実行し、前記音響信号出力処理は、前記タイミングで前記音響効果が付加された音響信号を出力する、請求項1乃至3のいずれかに記載の音響処理装置。

10

#### 【請求項5】

前記第1の処理部は、供給される音響信号に対してエコー効果を付加するディレイ処理を実行し、前記音響信号出力処理は、前記エコー効果が付加された音響信号を出力する、請求項1乃至4のいずれかに記載の音響処理装置。

#### 【請求項6】

前記第1の処理部は、供給される音響信号に対してビブラート効果及びトレモロ効果の少なくとも一方を付加するためのLFOを生成する処理を実行し、前記音響信号出力処理は、前記ビブラート効果及びトレモロ効果の少なくとも一方が付加された音響信号を出力する、請求項1乃至5のいずれかに記載の音響処理装置。

#### 【請求項7】

第1の処理部及び第2の処理部を有する音響処理装置に用いられる音響処理方法であつて、

20

前記第1の処理部に、

第1クロックを循環的に第1のカウント値までカウントする第1のカウント処理と、

前記第1のカウント処理によるカウントが前記第1のカウント値に達する毎に音響効果が付加された音響信号を出力する音響信号出力処理と、

を実行させ、

前記第2の処理部に、

第2クロックを循環的に第2のカウント値までカウントする第2のカウント処理と、

前記第2のカウント処理によるカウントが前記第2のカウント値に達する毎に自動演奏を行なせる自動演奏制御処理と、

30

前記第2のカウント処理によるカウントが前記第2のカウント値に達したタイミングと前記第1のカウント処理によるカウントが前記第1のカウント値に達したタイミングとの差分時間を循環的に検出する検出処理と、

前記検出処理により検出される差分時間を小さくするように前記第2のカウント値を変更する補正処理と、

を実行させる音響処理方法。

#### 【請求項8】

音響処理装置として用いられるコンピュータに、

第1クロックを循環的に第1のカウント値までカウントする第1のカウント処理と、

40

前記第1のカウント処理によるカウントが前記第1のカウント値に達する毎に音響効果が付加された音響信号を出力する音響信号出力処理と、

第2クロックを循環的に第2のカウント値までカウントする第2のカウント処理と、

前記第2のカウント処理によるカウントが前記第2のカウント値に達する毎に自動演奏を行なせる自動演奏制御処理と、

前記第2のカウント処理によるカウントが前記第2のカウント値に達したタイミングと前記第1のカウント処理によるカウントが前記第1のカウント値に達したタイミングとの差分時間を循環的に検出する検出処理と、

前記検出処理により検出される差分時間を小さくするように前記第2のカウント値を変更する補正処理と、

50

を実行させるプログラム。

【請求項 9】

請求項 1 乃至 6 のいずれかに記載の音響処理装置と、  
発生すべき楽音の音高を指定する演奏操作子と、  
前記演奏操作子で指定された音高の楽音波形を音響波形信号として生成して、前記第 1  
の処理部に供給する波形発生器と、  
を備えた電子楽器。

【発明の詳細な説明】

【技術分野】

【0001】

10

本発明は、2つの音響処理プロセッサ間の処理を同期させるための音響処理装置、音響  
処理方法、プログラム、並びに当該音響処理装置を用いた電子楽器に関する。

【背景技術】

【0002】

電子楽器に内蔵され入力信号に対してエコー効果を与えるディレイ機能を搭載した音響  
効果装置において、従来、電子楽器の自動演奏（伴奏、シーケンサ、アルペジオ等）のテ  
ンポの設定に応じて、遅延信号が音楽のリズムと同期するようにディレイタイムを自動的  
に設定するテンポ同期ディレイという技術が知られている（例えば特許文献 1 に記載の技  
術）。更に、楽曲の演奏中であっても、演奏のテンポを変更させるための操作をするとエ  
フェクト処理の内容を変更させるようにした従来技術も知られている（例えば特許文献 2  
に記載の技術）。

20

【0003】

テンポ同期型ディレイ機能において、ディレイタイムを1拍の時間の倍数、例えば1 /  
4拍、1 / 3拍、1 / 2拍、2 / 3拍、1拍、3 / 2拍、2拍、3拍などに設定するこ  
とで、より音楽的に合致したエコー効果を得ることができる。更に、このディレイ機能を利  
用して、同じ演奏を何度も繰り返して演奏するサンプルルーパーという機能も一般的に知  
られている。

【先行技術文献】

【特許文献】

【0004】

30

【特許文献 1】特開平5 - 94180号公報

【特許文献 2】特開2011 - 215363号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

しかしながら、従来のテンポ同期型ディレイ機能におけるディレイタイムは、完全に自  
動演奏のテンポと同期しているわけではなく、可能な範囲でテンポに同期するための近似  
値に設定されているだけであり、厳密には少しずれているのが実情である。ディレイのフ  
ィードバック量が少なくリピート回数が比較的少ないときにはこのずれ量は気にならない  
。しかし、ディレイのフィードバック量を多くしてリピート回数を多くしたり、更にはフ  
ィードバック量を100%にしてサンプルルーパーとして何度も同じ演奏を繰り返し再生  
する場合、繰返し回数分のずれ量が蓄積してしまい、人間の耳でも聞き取れるまで誤差が  
拡大することが起こりうるという問題点があった。

40

そこで、本発明は、自動演奏処理と音響効果処理の処理タイミングのずれを補正可能と  
することを目的とする。

【課題を解決するための手段】

【0006】

態様の一例の音響処理装置は、

第1クロックを循環的に第1のカウント値までカウントする第1のカウント処理と、前  
記第1のカウント処理によるカウントが前記第1のカウント値に達する毎に音響効果が付

50

加された音響信号を出力する音響信号出力処理と、を実行する第1の処理部と、  
第2クロックを循環的に第2のカウント値までカウントする第2のカウント処理と、前記第2のカウント処理によるカウントが前記第2のカウント値に達する毎に自動演奏を進行させる自動演奏制御処理と、前記第2のカウント処理によるカウントが前記第2のカウント値に達したタイミングと前記第1のカウント処理によるカウントが前記第1のカウント値に達したタイミングとの差分時間を循環的に検出する検出処理と、前記検出処理により検出される差分時間を小さくするように前記第2のカウント値を変更する補正処理と、  
を実行する第2の処理部と、

を備える。

【発明の効果】

10

【0007】

本発明によれば、自動演奏処理と音響効果処理の処理タイミングのずれを補正することが可能となる。

【図面の簡単な説明】

【0008】

【図1】本発明による電子鍵盤楽器100の実施形態の外観図である。

【図2】電子鍵盤楽器100の実施形態のハードウェア構成例を示す図である。

【図3】CPU201が内蔵するカウンタ回路の構成例を示す図である。

【図4】DSP206が実行するディレイ処理の説明図である。

【図5】テンポ - カウントテーブル(TEMPO\_COUNT\_TBL)と補助情報テーブルのデータ構成例を示す図である。 20

【図6】DSP206が備えるディレイタイム割込み発生処理を実行する機構及びアドレスポインタを生成する機構の例を示すプロック図である。

【図7】同期拍数テーブル(SYNC\_BEAT\_TBL)と補助情報テーブルのデータ構成例を示す図である。

【図8】本実施形態の動作説明図である。

【図9】CPU ROM202又はCPU RAM203が記憶する各種データのデータ構成例を示す図である。

【図10】CPU201及びDSP206のレジスタのデータ構成例を示す図である。

【図11】CPU201が実行する電子楽器制御処理の全体処理例を示すフローチャートである。 30

【図12】初期化処理の詳細例を示すフローチャートである。

【図13】テンポ設定処理の詳細例を示すフローチャートである。

【図14】ディレイ設定処理の詳細例を示すフローチャートである。

【図15】ディレイホールドモード設定処理(HOLD処理)、ディレイタイム背亭処理(TIME処理)、ディレイフィードバック設定処理(FEEDBACK処理)、及びディレイレベル設定処理(LEVEL処理)の詳細例を示すフローチャートである。

【図16】ディレイテンポシンクロナイズドモード設定処理(SYNC処理)及びディレイ同期拍モードr(BEAT処理)の詳細例を示すフローチャートである。

【図17】自動演奏定常処理の詳細例を示すフローチャートである。 40

【図18】シーケンスクロック割込み処理の例を示すフローチャートである。

【図19】ディレイタイム割込み処理の例を示すフローチャートである。

【図20】シーケンスクロック補正処理の詳細例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】

【0009】

従来、ディレイ処理のディレイタイムを自動演奏のテンポに完全に同期させることができなかったのは、以下の理由による。まず、自動演奏を進行させる自動演奏制御処理は、一般的に、CPU(中央演算処理装置)により実行される。CPUは、ハードウェアタイミングシステムクロックを、ゼロから、指定されたテンポの1拍を所定数分割、例えば480分割した時間に対応するシーケンスクロックカウント上限値まで、循環的にカウ 50

ントするシーケンスロックカウンタを実装する。このシーケンスロックカウンタは、システムクロックのカウント値が、シーケンスロックカウント上限値に達する毎に、シーケンスロック割込みを発生する。C P Uは、このシーケンスロック割込みに同期して自動演奏を進行させる自動演奏制御処理を実行する。これに対して、ディレイ処理は、一般的に、デジタル音声処理を専用のハードウェアと専用のソフトウェアにより実行するD S P（デジタルシグナルプロセッサ）を用いて実現される。このディレイ処理におけるディレイタイムは、D S Pにおいて、指定されたテンポの1拍に対して所定自然数倍又は所定自然数分の1倍の関係を有するように、例えば1/4拍、1/3拍、1/2拍、2/3拍、1拍、3/2拍、2拍、3拍などの値に設定される。このとき、ディレイタイムは、サンプリングクロックの何クロック分にあたるかというサンプル数で与えられる。D S Pは、音響信号を1サンプルずつ記憶するディレイ用リングバッファメモリの書き込みアドレスと読み出しアドレスの差を、上記ディレイタイムのサンプル数に設定する。そして、D S Pは、ディレイ用リングバッファメモリに書き込んだ音響信号を、上記ディレイタイムに対応するサンプル数分だけ遅延させて読み出すことによって、テンポの拍に同期したディレイ効果を実現する。ここで、D S Pにおけるサンプリングクロックと、C P Uにおけるシステムクロックは、それぞれ異なる発振子から生成されているため、従来は、ディレイ処理のディレイタイムを自動演奏のテンポに完全に同期させることができなかつた。この結果、従来例えば、ディレイ処理で出力される音響信号を入力側にフィードバックして音響信号の書き込みと遅延読み出しを繰り返す、いわゆるサンプルルーパー等の処理を実行したような場合に、自動演奏のテンポがディレイ処理後の音響信号のテンポからずれていつてしまい、聴感上違和感を生じてしまっていた。10

#### 【0010】

本実施形態は、以下に説明する動作を実行するディレイタイム割込み発生処理と、シーケンスロック補正処理の機能を備える処理同期部を、例えば第1又は第2の音響処理プロセッサ内に実装することにより、C P Uにおける自動演奏制御処理をD S Pにおけるディレイ処理に同期させることを可能とするものである。20

#### 【0011】

ディレイタイム割込み発生処理は、例えばD S Pにより実行される。まず、D S P内に、サンプリングクロックをゼロからディレイタイムサンプリング数まで循環的にカウントするディレイタイムカウンタが実装される。このディレイタイムカウンタは、サンプリングクロックのカウント値がカウント上限値であるディレイタイムサンプリング数に達する毎に、C P Uに対してディレイタイム割込みを発生させる。30

#### 【0012】

一方、シーケンスロック補正処理は、例えばC P Uにより実行される。この処理において、C P Uは、前述のシーケンスロック割込みがディレイタイムに対応する回数だけ発生する毎の時刻と、D S Pからディレイタイム割込みが発生する毎の時刻との時間差を計測する。そして、C P Uは、その時間差が小さくなるように前述のシーケンスロックカウンタにセットするシーケンスロックカウント上限値を増加又は減少させる。この結果、次回のディレイ処理においては、C P Uにおいてディレイタイム分だけ進行する自動演奏の時間と、D S Pにおけるディレイタイム分のディレイ処理の時間とのずれが無くなることが期待される。40

#### 【0013】

上述の機能を実現する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以下に詳細に説明する。図1は、本発明による電子鍵盤楽器の実施形態の外観図である。本実施形態は、音響処理プロセッサ（C P UとD S P）間の処理同期装置を備えた電子楽器である電子鍵盤楽器100として実施される。電子鍵盤楽器100は、発生すべき楽音の音高を指定する演奏操作子としての複数の鍵からなる鍵盤101と、楽音にディレイ効果を与えるエフェクト機能を指定するための機能選択操作子102と、伴奏やシーケンサ、オートアルペジオなどの自動演奏機能や音色選択操作子としての音色選択を行うための音色選択ボタン103からなるスイッチ・パネルと、ピッチベンドやトレモロ、ビブラート等50

の各種モジュレーション（演奏効果）を付加するベンダ／モジュレーション・ホイール104、及び音色や音色以外の各種設定情報を表示するLCD（Liquid Crystal Display：液晶ディスプレイ）105等を備える。また、電子鍵盤楽器100は、特には図示しないが、演奏により生成された楽音を放音するスピーカを裏面部、側面部、又は背面部等に備える。

#### 【0014】

機能選択操作子102は、図1に示されるように、テンポ（TEMPO）を指定するためのDOWNボタン、UPボタン、ディレイ効果を与えるエフェクト機能（DELAY）を指定するためのHOLDボタン、TIMEノブ、REPEATノブ、LEVELノブ、SYNCボタン、及びBEATノブを備える。これらの詳細については後述する。

10

#### 【0015】

図2は、図1の電子鍵盤楽器100の実施形態のハードウェア構成例を示す図である。図2において、電子鍵盤楽器100は、CPU（中央演算処理装置）201、CPU ROM（リードオンリーメモリ）202、CPU RAM（ランダムアクセスメモリ）203、音源LSI（大規模集積回路）204、キー・スキャナ211、A/Dコンバータ212、及びMIDI I/F（Musical Instrument Digital Interface）214が、それぞれシステムバス216に接続される構成を備える。CPU201にはシステムクロックを供給する発振子209が接続され、音源LSI204にはサンプリングクロックを供給する発振子210が接続される。音源LSI204は、波形発生器205とDSP206とCPU I/F215を内蔵する。波形発生器205には、波形メモリである音源ROM207が接続される。DSP206には、遅延メモリであるDSP RAM208が接続される。本実施形態においては、このDSP206及びDSP RAM208により、第1の処理部206aを構成する。キー・スキャナ211には、図1の鍵盤105と図1の機能選択操作子102及び音色選択ボタン103からなるスイッチ・パネルとが接続される。A/Dコンバータ212には、図1のベンダ／モジュレーション・ホイール104が接続される。LCDコントローラ213には、図1のLCD105が接続される。LCDコントローラ213、MIDI I/F214は、MIDI入力を受け付ける。また、音源LSI205内のDSP206から出力されるデジタル楽音波形データは、D/Aコンバータ217によりアナログ楽音波形信号に変換され、アンプ218で増幅された後に、特には図示しないスピーカ又は出力端子から出力される。

20

#### 【0016】

CPU201は、CPU RAM203をワークメモリとして使用しながらCPU ROM202に記憶された制御プログラムを実行することにより、図1の電子鍵盤楽器100の制御動作を実行する。CPU201は、システムバス216からキー・スキャナ211又はA/Dコンバータ212を介して、図1の鍵盤101、機能選択操作子102、音色選択ボタン103、ベンダ／モジュレーション・ホイール104等の演奏指示を受け取る。または、CPU201は、MIDI I/F214を介して、外部の図示しない機器から、演奏指示を示すMIDI入力を受け取る。CPU201は、これらの演奏指示に基づいて、音源LSI204に対して、楽音の発音／消音の指示、ディレイ効果の付与指示等を出力する。また、CPU201は、CPU ROM202に記憶された自動演奏データに基づいて、自動演奏による楽音の発音／消音の指示等を出力する。

30

本実施形態においては、このCPU201により第1の処理部を構成する。

40

#### 【0017】

音源LSI204に内蔵される波形発生器205は、CPU201からの楽音の発音／消音指示に従って、波形メモリである音源ROM207から楽音波形データを読み出し、DSP206に音響信号として供給する。

#### 【0018】

音源LSI204に内蔵されるDSP206は、波形発生器205から入力する楽音波形データに対して、DSP RAM208を遅延メモリとして使用しながらディレイ効果

50

を付加し、その結果得られる楽音波形データをD/Aコンバータ217に出力する。ディレイ効果の設定は、CPU201から設定される。

#### 【0019】

音源LSI204に内蔵されるCPU I/F215は、CPU201との間の各種データの通信および割込み指示を処理する。CPU I/F215により、CPU201からはDSP206が単なるメモリとして見え、DSP206からもCPU201がメモリとして見えるように制御される。一方がメモリに書き込みを行うと、他方がそれを読むことができる仕組みになっている。

#### 【0020】

発振子209は、CPU201に対して基準クロックであるシステムクロックを供給する。発振子210は、音源LSI204内の波形発生器205及びDSP206に対してサンプリングクロックを生成するための基準クロックを供給する。音源LSI204が内蔵する波形発生器205及びDSP206は、音源LSI204専用の発振子210からサンプリングクロックを生成するための共通の基準クロックが供給されるため、完全に同期して動作する。一方、CPU201は、専用の発振子209からシステムクロックを供給されて動作する。このため、CPU201の動作は基本的には、波形発生器205及びDSP206の動作とは同期しない。しかしながら、本実施形態では、後述する処理同期機能により、CPU201が実行する自動演奏制御処理のテンポは、DSP206が実行するディレイ処理におけるディレイタイムに同期するように制御される。

本実施形態においては、発振子210により第1のクロック発生部を構成し、発振子209により第2のクロック発生部を構成する。

#### 【0021】

キー・スキャナ207は、鍵盤105や機能選択操作子102又は音色選択ボタン103等のスイッチ・パネルの状態を走査して、CPU201に通知するIC(集積回路)である。A/Dコンバータ208は、ベンダ/モジュレーション・ホール104の操作位置を示すアナログ信号をデジタル信号として検出するICである。LCDコントローラ209は、LCD105を制御するICである。

#### 【0022】

図3は、図2のCPU201が内蔵するカウンタ回路の構成例を示す図である。CPU201は、発振子209が発振するシステムクロックを分周して計数を行うシーケンスクロックカウンタ301と、時計として時刻を参照するためのフリーランニングタイマカウンタ302を内蔵する。これらのタイマカウンタは共に、 $1\mu\text{sec}$ (マイクロ秒)単位で値をインクリメントする。シーケンスクロックカウンタ301は、発振子209が発振するシステムクロックを、ゼロから、指定されたテンポの1拍を480分割した時間に対応するシーケンスクロックカウント上限値まで、循環的にカウントする。シーケンスクロックカウンタ301は、システムクロックのカウント値がシーケンスクロックカウント上限値に達する毎に、CPU201に対してシーケンスクロック割込みを発生する。シーケンスクロックカウンタ301は、カウント値がシーケンスクロックカウント上限値に達すると、カウント値を0に戻って係数を循環的に繰り返す。

本実施形態においては、このシーケンスクロックカウンタ301が第2のカウンタに対応し、シーケンスクロックカウント上限値が第2のカウント値に対応する。

#### 【0023】

CPU201は、シーケンスクロックカウンタ301が発生させるシーケンスクロック割込みに同期して、自動演奏を進行させる自動演奏制御処理を実行する。すなわち、自動演奏制御処理は、指定されたテンポの1拍を480分割したシーケンスクロック割込みを基準として動作する。テンポの1拍の間に480回のシーケンスクロック割込みのトリガ信号が発生され、それに併せて自動演奏が進行する。演奏のテンポは、このシーケンスクロック割込みの発生頻度に依存する。自動演奏のテンポは、図1の機能選択操作子102中のTEMPOエリアのDOWNボタン102a又はUPボタン102bによって、上下させることができる。このDOWNボタン102aおよびUPボタン102bにより、本

10

20

30

40

50

実施形態においては、テンポ指定部を構成する。テンポは、最低値 30 BPM (Beat Per Minute) から、1 BPM 刻みで、最高値 300 BPM まで設定できる。これに同期して、シーケンスロック割込みの時間間隔が決定される。

#### 【0024】

フリーランニングタイマカウンタ 302 は、32 ビットのデータ幅を持ち、最大値を超えると 0 に戻って計数を繰り返す時計であり、参照した時点の時刻をカウンタ値で知ることができる。CPU201 は、後述するように、シーケンスロック割込みが音源 LSI 204 内の DSP206 で実行されるディレイ処理において設定されているディレイタイムに対応する回数だけ発生する毎に、フリーランニングタイマカウンタ 302 を参照することにより、各発生時刻を取得して DSP206 におけるディレイ処理との同期制御を実行する。

10

#### 【0025】

次に、音源 LSI 204 内での音響信号の流れについて説明する。図 2 に示されるように、波形発生器 205 から出力される音響信号は、DSP206 に送られて、この DSP206 内でディレイ効果を付加される。波形発生器 205 から出力される音響信号のうち、ユーザ（演奏者）による鍵盤 101 の演奏に基づいて波形発生器 205 で生成される音響信号に対してディレイ効果が付加される。一方、CPU201 による自動演奏制御処理に基づいて波形発生器 205 で生成される音響信号については、ディレイ効果が付加されずに DSP206 の出力段でディレイ効果が付加された音響信号と混合されて出力されてもよいし、ディレイ効果が付加されてもよい。音源 LSI 204 内部での音響信号の受け渡し、及び DSP206 から出力される音響信号はいずれも、同じサンプリングクロックに基づいて例えば 44.1 KHz (キロヘルツ) というサンプリング周波数でサンプリングされ、処理される。このサンプリングクロックは、音源 LSI 204 に接続される発振子 210 が発生させる基準クロックを音源 LSI 204 内部で分周することにより生成されており、発振子 210 の発振周波数とサンプリング周波数は完全に比例関係にある。発振子 210 の発振周波数に変動が起これば、同じ比率でサンプリング周波数に影響が及ぶ。

20

#### 【0026】

音源 LSI 204 内の波形発生器 205 は、一般的な波形読み出し方式による楽音波形の生成機能を備える。具体的には、波形発生器 205 は、波形メモリとして機能する音源 ROM207 から、CPU201 から予め指示された種別の楽音波形データを、CPU201 から逐次指定されるノートオン指示で指定された音高に対応する読み出し速度で補間を行いながら読み出すことにより、音響信号を生成する。

30

#### 【0027】

図 4 (a) は、音源 LSI 204 内の DSP206 が実行するディレイ処理の機能ブロック図である。まず、波形発生器 205 から入力された音響信号 407 が分岐させられ、一方はダイレクトに出力側の加算器 406 へ送られ、もう一方はアンプ 402 及び加算器 405 から遅延装置 401 に送られ、遅延処理が実行される。遅延装置 401 の出力は、アンプ 403 を介して加算器 406 で、原音である上記ダイレクトの音響信号 407 と混合される。このときの原音に対する遅延音のレベルは、ディレイ入力音量調整用のアンプ 402 とディレイ出力音量調整用のアンプ 403 によって調整することができる。遅延装置 401 の出力はフィードバック量調整用のアンプ 404 によって設定された量だけ、加算器 405 を介して遅延装置 401 の入力側にフィードバックすることができる。アンプ 404 における増幅率を 1.0 倍に近づけるほど、音響信号のリピート回数が多くなる。

40

#### 【0028】

また、このディレイ処理にはディレイホールドモードという特殊モードが用意されている。ユーザが図 1 の機能選択操作子 102 内の HOLD ボタンを押下してその LED を点灯させることにより、ディレイホールドモードが設定される。この状態になると、フィードバック量調整用のアンプ 404 のゲインは 1.0 倍となり、ユーザが再び HOLD ボタンを押下してその LED を消灯させてこの状態を解除するまで、遅延装置 401 が出力す

50

る遅延された音響信号を減衰させずに永遠に繰り返し出力することができる。このモードがオンされた瞬間に、通常1.0倍に設定されているディレイ入力音量調整用のアンプ402の増幅率は0倍に設定され、それ以降入力された音響信号407にディレイ効果がかからない状態になる。その結果、このモードがオンされる直前まで遅延装置401に入力していた音響信号が遅延装置401で繰り返し遅延させられる音響信号と、入力された音響信号407を加算器406でそのまま混合した音響信号が出力される。この機能は、いわゆるサンプルルーパーと呼ばれる。

#### 【0029】

図4(b)は、図4(a)の遅延装置401の説明図である。遅延装置401は、DSP206がDSPRAM208にアクセスする機能として実現される。DSPRAM208は、ディレイ用リングバッファメモリとして動作し、概念的にはリングバッファと呼ばれる形式で管理され、バッファエリアの最終アドレス414のメモリ領域は先頭アドレス413のメモリ領域と仮想的に連続しているように、最終アドレス414の次のアドレスにおいて、アドレスを指定するライトポインタ410及びリードポインタ411の値の折り返しが行われる。このディレイ用リングバッファメモリ上のライトポインタ410の値が示す書込みアドレスとリードポインタ411の値が示す読出しあдресを一定の間隔で設定しておき、サンプリングクロックの一周期ごとに、各ポインタの値を1アドレスずつインクリメントし、加算器405から入力された音響信号の値の書込みと、出力側に送る音響信号の値の読出しが行われる。また、図4(c)に示されるように、ディレイ用リングバッファメモリにおいて、ライトポインタ410又はリードポインタ411が最終アドレス414を指し示している場合、次に指し示すのは先頭アドレス413となる。このアドレス移動は、機器が動作している限り無限に繰り返される。  
10

#### 【0030】

この結果、図4(b)又は(c)に示されるように、上記書込みが行われてから読出しが行われるまでディレイ用リングバッファメモリ上の各アドレスに一時記憶されている音響信号の各16ビットの波高値が、遅延中の音響信号412となる。また、上記ライトポインタ410とリードポインタ411の相対的なアドレス差が、遅延量を示すディレイタイムとなる。1アドレス離れるごとに、下記(1)式で示される遅延が発生する。  
20

#### 【0031】

$$(1 \div 44.1 \text{ KHz}) \text{ sec} \quad \text{約} 22.7 \mu\text{sec} \quad \dots (1)$$

30

#### 【0032】

本実施例では、例えば、 $3000000 \times 2$ バイト( $3000000$ ワード)のディレイ用リングバッファメモリを有し、1サンプルは1ワード(2バイト=16ビット)なので、最大で $22.7 \mu\text{sec} \times 3000000$  約6.8secのディレイタイムが実現できる。例えば、1secのディレイタイムを発生させたい場合には、ライトポインタ410はリードポインタ411よりも、 $44.1 \text{ KHz} \times 1000 \times 1 \text{ sec} = 44100$ アドレス少ない値に設定しておけば良い。図4(b)又は(c)に示される設定例では、ディレイ用リングバッファメモリの約1/4の記憶領域が遅延中の音響信号412となっているため、 $6.8 \text{ sec} \div 4 = \text{約} 1.7 \text{ sec}$ のディレイタイムが発生するよう設定されている。従って、ライトポインタ410とリードポインタ411の差は、 $44100 \times 1.7 \text{ sec} = 74970$ アドレスとなる。  
40

#### 【0033】

上述のディレイタイムを含むディレイ効果指定は、図1の機能選択操作子102のDELAYエリアに示される操作子群によって、下記のように行うことができる。

#### 【0034】

・SYNCボタン(LEDインジケータ付き)：この操作子により、ディレイタイムをテンポに同期させるモードであるディレイテンポシンクロナイズドモードを使用するか否かが切り替えられる。ユーザは、このモードをオンした場合には、後述するBEATノブにより、ディレイタイムをテンポに同期させる設定を行うことができるようになる。なお、このモードがオフの場合には、ディレイタイムは後述するTIMEノブにより自由な値  
50

に設定することができる。

・**HOLD**ボタン( LED インジケータ付き)：この操作子により、前述のディレイホールドモードを使用するか否かが切り替えられる。

・**B E A T**ノブ：ディレイテンポシンクロナイズドモードがオンされたときに、この操作子により、ディレイタイムがテンポの何拍に同期するかを設定する。ノブの位置に応じて、設定値 0 ( 1 / 4 拍 ) 、設定値 1 ( 1 / 3 拍 ) 、設定値 2 ( 1 / 2 拍 ) 、設定値 3 ( 2 / 3 拍 ) 、設定値 4 ( 1 拍 ) 、設定値 5 ( 3 / 2 拍 ) 、設定値 6 ( 2 拍 ) 、設定値 7 ( 3 拍 ) の 8 種類から選択できる。ディレイテンポシンクロナイズドモードがオフのときには、**B E A T**ノブの操作は無視される。

・**T I M E**ノブ：ディレイテンポシンクロナイズドモードがオンされないときに、この操作子により、ディレイタイムが直接指定される。例えば、 0 ~ 1 2 3 1 6 ( 2 s e c ) の間で調整できる。ディレイテンポシンクロナイズドモードがオンのときには、**T I M E**ノブの操作は無視される。10

・**R E P E A T**ノブ：ディレイホールドモードがオンされていないときに、この操作子により、ディレイのフィードバック量が調整される。ここで指定された値により、図 4 ( a ) のフィードバック量調整用のアンプ 4 0 4 のゲインが決定される。ディレイホールドモードがオンのときには、上記フィードバック量は強制的に 1 0 0 % に設定される。

・**L E V E L**ノブ：この操作子により、ディレイ信号のレベルが調整される。ここで指定された値により、図 4 ( a ) のディレイ出力音量調整用のアンプ 4 0 3 のゲインが決定される。20

#### 【 0 0 3 5 】

上記**B E A T**、**T I M E**、**R E P E A T**、**L E V E L**の 4 つのノブは、割り当てられたパラメータの値を、設定位置(角度)に応じて最大値と最小値の間で設定することができる。例えば中央の位置に操作すれば最大値と最小値の中央の値に設定される。

#### 【 0 0 3 6 】

次に、本実施形態において C P U 2 0 1 が実行する自動演奏制御処理を音源 L S I 2 0 4 内の D S P 2 0 6 が実行するディレイ処理に同期させるための制御動作について説明する。前述したように、C P U 2 0 1 が実行する自動演奏制御処理のタイミングは、C P U 2 0 1 内のシーケンスロックカウンタ 3 0 1 が、発振子 2 0 9 からのシステムクロックを、指定されたテンポ( B P M )の 1 拍を 4 8 0 分割した時間に対応するシーケンスロックカウント上限値まで循環的にカウントする毎に発生するシーケンスロック割込みに基いて、制御される。ここで、ディレイテンポシンクロナイズドモード、すなわちディレイタイムを指定されたテンポに同期させるモードが指定されたときには、C P U 2 0 1 が、D S P 2 0 6 が実現する遅延装置 4 0 1 ( 図 4 ( a ) 参照 ) に設定されるサンプリングクロックに同期したディレイタイムと、上記シーケンスロック割込みに同期したディレイタイムに対応する時間とのずれを逐次監視して、そのずれを補正するようにシーケンスロック割込みの発生タイミングを調整しながら動作する。音源 L S I 2 0 4 が発生するサンプリングクロックをそのまま自動演奏制御処理の基準クロックとして使用することも理論的には可能である。しかし、現実的には 4 4 . 1 K H z というサンプリングレート、すなわち 2 2 . 7  $\mu$  s e c という単位で 1 拍を 4 8 0 分の 1 に分割したシーケンスロック割込みに変化をつけるには精度が荒すぎるのと、C P U 2 0 1 が生成した 1  $\mu$  s e c 単位でされたシステムクロックとの差が広がって認識可能なレベルになるのは数 1 0 拍も演奏してからの現象であることを考えると、ディレイ処理と自動演奏制御処理とを別々のクロックで動作させて、隨時補正するのが現実的であると考えられる。そこで、本実施形態では、ディレイテンポシンクロナイズドモードにおいて指定されたテンポに対して自然数倍又は自然数分の 1 倍の関係を有するように設定されたディレイタイムについて、D S P 2 0 6 上でサンプリングクロックを単位として計測した時間と、C P U 2 0 1 上でシーケンスロック割込みの回数として計測した時間とを、C P U 2 0 1 が比較する。そして、C P U 2 0 1 で計測された時間のほうが、遅れている場合はシーケンスロックカウンタ 3 0 1 のカウントが上限に達するシーケンスロックカウント上限値を減少させてシーケ304050

ンスクロック割込みのタイミングを少し速め、逆に進んでいる場合はシーケンスクロックカウント上限値を増加させてシーケンスクロック割込みのタイミングを遅くすることにより、徐々に近づくような調整操作を自動的に行う。

#### 【0037】

ディレイ時間と音楽の周期が一致するだけでなく、音楽とディレイ効果によって発音する1つ1つの楽音の位相が合わなければ音楽的には成立しないが、本実施形態での上記調整操作により、ディレイの周期が合うだけでなく、位相も一致させることができる。

#### 【0038】

図5(a)は、CPU ROM202が記憶するテンポ - カウントテーブル(以下「TEMPO\_COUNT\_TBL」とも表記する)のデータ構成例を示す図であり、また、図5(b)は、TEMPO\_COUNT\_TBLの説明に使用する補助情報テーブルのデータ構成例を示す図である。なお、補助情報テーブルは、本実施形態においてTEMPO\_COUNT\_TBLを説明する便宜上図示したものであり、実装されるものではない。図5(a)に例示されるTEMPO\_COUNT\_TBLは、TEMPO、DELAY\_COUNT、SEQ\_CLOCK\_COUNTの各項目を有する。TEMPO\_COUNT\_TBLの各行は、30~300(BPM: Beat Per Minute、時間1分あたりの拍数)まで1刻みで指定されるTEMPO項目値の各々について、DELAY\_COUNT項目値とSEQ\_CLOCK\_COUNT項目値が設定されるエントリを示している。なお、以下の説明では、TEMPO項目、DELAY\_COUNT項目、又はSEQ\_CLOCK\_COUNT項目の各値そのものを、TEMPO、DELAY\_COUNT、又はSEQ\_CLOCK\_COUNTと表記する場合もある。

10

#### 【0039】

DELAY\_COUNTは、DSP206(図2)内の遅延装置401(図4(a)参照)に設定されるディレイタイムを指定されたテンポ値(TEMPO)の1拍に同期させたときに、そのディレイタイムを音源LSI204に接続される図2の発振子210に基づいて生成されるサンプリングクロックでカウントしたときの、サンプリングクロック数を示している。例えばサンプリングクロックの周波数を44.1KHzとすれば、TEMPO\_COUNT\_TBLのTEMPO毎のエントリに設定されるDELAY\_COUNTは、下記(2)式により計算される値である。

20

#### 【0040】

$$\text{DELAY\_COUNT} = \\ (60 / \text{TEMPO}) / \{ (1 / (44.1 \times 1000)) \} \quad \dots (2)$$

30

#### 【0041】

SEQ\_CLOCK\_COUNTは、CPU201に接続される発振子209からのシステムクロックによりカウントアップされるCPU201内のシーケンスクロックカウンタ301(図3)が発生する、指定されたテンポ値(TEMPO)の1拍を480分割するシーケンスクロック割込みの発生に要する、システムクロック数を示している。システムクロックの1周期を例えば1μsecとすれば、SEQ\_CLOCK\_COUNTは、1シーケンスクロックをCPU201のシステムクロックでカウントしたときの所要時間(μsec)となる。TEMPO\_COUNT\_TBLのTEMPO毎のエントリに設定されるSEQ\_CLOCK\_COUNT(μsec)は、下記(3)式により計算される値である。

40

#### 【0042】

$$\text{SEQ\_CLOCK\_COUNT} = \\ \{ (60 / \text{TEMPO}) / 480 \} \times 1000000 \quad \dots (3)$$

#### 【0043】

次に、図5(b)の補助情報1は、DELAY\_COUNTが示すサンプリングクロック数分をサンプリングするのに要する所要時間、すなわちDSP206におけるサンプリングクロックを基準としたディレイタイムの時間(msc)を示す。この補助情報1は、図5(a)に例示されるTEMPO\_COUNT\_TBLのDELAY\_COUNT値

50

を用いて、下記(4)式により計算される。

【0044】

補足情報1 =

$$\{1 / (44.1 \times 1000) \times \text{DELAY\_COUNT} \times 1000\} \quad \dots \quad (4)$$

【0045】

次に、図5(b)の補助情報2は、上記補助情報1として算出されるDELAY\_COUNTのサンプリング所要時間と、CPU201内のシーケンスロック割込みで1拍分をカウントしたときの所要時間の時間差(sec)を示す。すなわち、ディレイタイムを指定されたテンポ値TEMPOの1拍に同期させた場合における、DSP206におけるサンプリングクロックを基準としたときの所要時間と、CPU201におけるシーケンスロック割込みを基準としたときの所要時間との時間差である。この補助情報2は、上記補助情報1(sec)と、図5(a)に例示されるTEMPO\_COUNT\_TBのSEQ\_CLOCK\_COUNT値(μsec)とを用いて、下記(5)式により計算される。  
10

【0046】

補足情報2 = 補足情報1 -

$$(SEQ_CLOCK_COUNT / 1000) \times 480 \quad \dots \quad (5)$$

【0047】

更に、図5(b)の補助情報3は、指定されたテンポ値TEMPOの1拍に同期するディレイタイムについて算出された上記時間差の、4分の4拍子で32小節分の自動演奏楽曲が演奏された場合における累積値を示す。この補助情報3は、上記補助情報2を用いて、下記(6)式により計算される。  
20

【0048】

補足情報3 = 補足情報2 × 4 × 32    ... (6)

【0049】

図5(b)において、補助情報2から知見されるように、指定されたテンポ値TEMPOの1拍に同期するディレイタイムの1ディレイ程度では、DSP206のサンプリングクロックに基づく制御と、CPU201でのシーケンスロック割込みに基づく制御とでは、大きな時間差は生じない。しかし、例えばディレイホールドモードにして前述したサンプルルーパーが実施されるような場合においては、ディレイされた音響信号が何小節にもわたって繰返し使用される。このような場合には、補助情報3から知見されるように、人間の耳で認識できるほどの大きな時間差が生じることがわかる。また、これは発振子の精度は無視した理論値であり、実際には上記の誤差発振の精度(±0.1% ~ ±0.001%)程度の差を加えて考える必要がある。  
30

【0050】

上記(5)式及び(6)式の関係より、図5(b)の補助情報2又は3においてプラス値が発生した場合には、CPU201におけるシーケンスロックに基づくディレイタイムのカウント値SEQ\_CLOCK\_COUNTのほうが、DSP206におけるサンプリングクロックに基づくディレイタイムのカウント値DELAY\_COUNTよりも大きいことを示している。従って、この場合にはSEQ\_CLOCK\_COUNTの値を上記プラス値の分だけ減少させれば、次のディレイ処理時に、DSP206のサンプリングクロックに基づく制御とCPU201でのシーケンスロック割込みに基づく制御との間の時間差が0に近づくことが期待できる。  
40

【0051】

一方、補助情報2又は3において、マイナス値が発生した場合には、CPU201におけるシーケンスロックに基づくディレイタイムのカウント値SEQ\_CLOCK\_COUNTのほうが、DSP206におけるサンプリングクロックに基づくディレイタイムのカウント値DELAY\_COUNTよりも小さいことを示している。従って、この場合にはSEQ\_CLOCK\_COUNTの値を上記マイナス値だけ増加させれば、次のディレイ処理時に、DSP206のサンプリングクロックに基づく制御とCPU201でのシ  
50

一ケンスクロック割込みに基づく制御との間の時間差が 0 に近づくことが期待できる。

#### 【0052】

以上の制御処理を実現するために、本実施形態はまず、DSP206 内に、CPU201 に対してディレイタイム割込みを発生する機構と、DSP206 に接続される DSPRAM208 にアクセスするためのアドレスポインタを生成する機構とを備える。図6は、DSP206 が備えるディレイタイム割込み発生処理を実行する機構及びディレイ処理のためのアドレスポインタを生成する機構の例を示すブロック図である。図6において、DSP206 は、ディレイタイムカウンタ601と、ディレイタイムサンプリング数レジスタ602と、符号反転器603と、加算器604と、ライトポインタ生成部605と、アドレス折返し部606と、データアクセス部607とを備える。

10

本実施形態においては、このディレイタイムカウンタ601が第1のカウンタを構成している。

さらに、DSP206 内の符号反転器603と、加算器604と、ライトポインタ生成部605と、アドレス折返し部606と、データアクセス部607と、このDSP206 に接続される DSPRAM208 とにより、音響効果回路610を構成する。

#### 【0053】

ユーザが図1の機能選択操作子102のDELAYエリアのSYNCボタンを押下してLEDインジケータを点灯させると、ディレイタイムをテンポに同期させるディレイテンポシンクロナ化ドモードが設定される。このモードの場合、CPU201はまず、CPUROM202に記憶されている図5(a)に示されているTEMPO\_COUNT\_TBLのユーザによって指定されたテンポ値がTEMPO項目に設定されているエンタリから、DELAY\_COUNT項目値、すなわち、ディレイタイムを指定されたテンポ値(TEMPO)の1拍に同期させたときのサンプリングクロック数を取得する。ただし、このDELAY\_COUNTは、指定されたテンポの1拍と同期させた場合の基準値であるので、CPU201は実際には、機能選択操作子102のDELAYエリアのBEATノブで設定された同期拍数の分周比率である、1/4拍、1/3拍、1/2拍、2/3拍、1拍、3/2拍、2拍、3拍のいずれかをDELAY\_COUNTに乗算することにより、指定されたテンポの指定された同期拍数のディレイタイムに対応するディレイタイムサンプリング数608を算出する。例えば、同期拍数として1/2拍が指定されれば、CPU201は、DELAY\_COUNTに1/2を乗算する。

20

#### 【0054】

CPU201は、以上のようにして算出したディレイタイムサンプリング数608を、システムバス216、CPU\_I/F215、及び音源LSI204を介して、DSP206内のディレイタイムサンプリング数レジスタ602にセットする。

#### 【0055】

ディレイタイムサンプリング数レジスタ602にセットされたディレイタイムサンプリング数608は、ディレイタイムカウンタ601にカウント上限値としてセットされる。ディレイタイムカウンタ601は、サンプリングクロックをゼロからディレイタイムサンプリング数608まで循環的にカウントするカウンタであり、サンプリングクロックのカウント値がこの上限値に達する毎にCPU201に対してディレイタイム割込み609を発生させて、同期拍数の値に設定されたディレイタイムが一周期経過したことを知らせ、その後カウント値を0にリセットしてカウント動作を繰り返す。

40

本実施形態においては、このディレイタイムカウンタ601のカウント上限値が第1のカウンタ値に相当する。

#### 【0056】

ライトポインタ生成部605は、図4(b)及び(c)で説明したライトポインタ410を生成する部分である。ライトポインタ生成部605は、サンプリングクロックに従って図4の先頭アドレス413から最終アドレス414まで+1ずつ値が増加するアドレス値を生成し、このアドレス値が最終アドレス414に達すると次にこのアドレス値を先頭アドレス413に戻し、以後同様の動作を繰り返す。このようにして生成されたライトボ

50

インタ410のアドレス値が、DSP206に接続されるDSP RAM208（図2参照）に、書込みアドレスとして供給される。

#### 【0057】

ディレイタイムサンプリング数レジスタ602にセットされたディレイタイムサンプリング数608は、符号反転器603でマイナス値に変換された後、加算器604に入力する。加算器604は、ライトポインタ生成部605が生成したライトポインタ410の値にディレイタイムサンプリング数608のマイナス値を加算、すなわち、ライトポインタ410の値からディレイタイムサンプリング数608の値を減算することによって、図4(b)及び(c)で説明したリードポインタ411を生成する。アドレス折返し部606は、リードポインタ411の値が図4(c)の先頭アドレス413の値未満となった場合に、リードポインタ411の値に(最終アドレス414の値 - 先頭アドレス413の値 + 1)を加算することにより、最終アドレス414に近い側のアドレスに折り返す。このようにして最終的に生成されたリードポインタ411のアドレス値が、DSP206に接続されるDSP RAM208に、読み出しがアドレスとして供給される。10

#### 【0058】

データアクセス部607は、サンプリングクロック毎に、ライトポインタ生成部605からDSP RAM208に供給される書込みアドレスに波形発生部205から供給される音響波形信号及びDSP RAM208から読み出された音響波形信号の少なくとも一方を書き込み、アドレス折返し部606からDSP RAM208に供給される読み出しアドレスから音響波形信号を読み出してD/Aコンバータ217（図2参照）に出力する（図4(b)又は(c)参照）。これにより、図4(a)の遅延装置401の機能が実現される。このように、DSP206は、波形発生器205（図2参照）から入力される音響信号に対して、上述の遅延装置401の動作を含む図4(a)のブロック図で示される処理を実行し、加算器406から出力される音響信号を、出力408としてD/Aコンバータ217に出力する。20

#### 【0059】

上述のDSP206の構成に加えて、本実施形態では、CPU201内に、シーケンスクロック補正処理を実行する機構を備える。前述したディレイテンポシンクロナイズモードの場合、CPU201はまず、CPU ROM202に記憶されている図5(a)に例示されているTEMPO\_COUNT\_TBLのユーザによって指定されたテンポ値がTEMPO項目に設定されているエントリから、SEQ\_CLOCK\_COUNT項目値、すなわち、指定されたテンポ(TEMPO)の1拍を480分割するシーケンスクロック割込みの発生に要するシステムクロック数(μsec)を取得し、CPU201内のシーケンスクロックカウンタ301（図3参照）にシーケンスクロックカウント上限値としてセットする。前述したように、シーケンスクロックカウンタ301は、発振子209が発振するシステムクロックを、ゼロからシーケンスクロックカウント上限値まで循環的にカウントする。シーケンスクロックカウンタ301は、システムクロックのカウント値がシーケンスクロックカウント上限値に達する毎に、CPU201に対してシーケンスクロック割込みを発生させて、指定テンポの1拍を480分割したシーケンスクロック周期が経過したことを知らせ、その後カウント値を0にリセットしてカウント動作を繰り返す。30

#### 【0060】

CPU201は、機能選択操作子102のDELAYエリアのBEATノブで設定された設定値に対応する同期拍数に応じた回数のシーケンスクロック割込みをシーケンスクロックカウンタ301から受けると、その時点の時刻をフリーランニングタイマカウンタ302（図3）のカウンタ値として読み取って記録する。CPU201は、この回数を、CPU ROM202が記憶する同期拍数テーブルから読みとる。図7(a)は、この同期拍数テーブル（以下「SYNC\_BEAT\_TBL」とも表記する）のデータ構成例を示す図である。また、図7(b)は、SYNC\_BEAT\_TBLの説明上使用する補助情報テーブルのデータ構成例を示す図である。図5(b)の補助情報テーブルと同様に、図7(b)の補助情報テーブルも、本実施形態においてSYNC\_BEAT\_TBLを説明4050

する便宜上図示したものであり、実装されるものではない。ユーザが、B E A T ノブで、8種類の同期拍数の設定値0、1、2、3、4、5、6、7のうちのいずれか1つを設定すると、C P U 2 0 1は、その設定値を設定値項目に有するS Y N C \_ B E A T \_ T B Lのエントリを参照し、分子(N U M E R A T O R)項目の値と分母(D E N O M I N A T O R)項目の値を取得する。なお、以下の説明では、N U M E R A T O R項目、D E N O M I N A T O R項目の各値そのものを、N U M E R A T O R、D E N O M I N A T O Rと表記する場合もある。C P U 2 0 1は、これらの値に基づいて、「N U M E R A T O R / D E N O M I N A T O R」の値として、同期拍数を決定する。図7(b)の補助情報テーブルには、設定値毎の同期拍数の値を示してある。このようにして、設定値0、1、2、3、4、5、6、7のそれぞれに対応して、1/4拍、1/3拍、1/2拍、2/3拍、1拍、3/2拍、2拍、3拍という同期拍数が算出されることになる。また、図7(b)の補助情報テーブルには、各同期拍数に対して、シーケンスロック割込みによってカウントされるべき回数(以下「同期シーケンスロック数」と表記)を示してある。例えば、設定値=4の場合には、N U M E R A T O R=1、D E N O M I N A T O R=1であるから、同期拍数=1/1=1である。そして、シーケンスロック割込みは1拍を480分割したものであるから、同期拍数が1の場合の同期シーケンスロック数は480となる。また、設定値=0の場合は、N U M E R A T O R=1、D E N O M I N A T O R=4であるから、同期拍数=1/4であり、同期シーケンスロック数は、 $480 \times 1/4 = 120$ となる。具体的には、C P U 2 0 1は、B E A T ノブで指定された設定値に対応するエントリのN U M E R A T O R項目値とD E N O M I N A T O R項目値を取得し、下記(7)式に従って同期拍数bを計算し、更に、下記(8)式に従って同期シーケンスロック数s、すなわち、フリーランニングタイマカウンタ302から時刻を読みとるべきシーケンスロック割込みの回数を決定する。

## 【0061】

b = N U M E R A T O R / D E N O M I N A T O R .. (7)

## 【0062】

s = 480 × b .. (8)

## 【0063】

一方、C P U 2 0 1は、D S P 2 0 6内のディレイタイムカウンタ601から、ディレイタイム割込み609を受けると、その時点の時刻もフリーランニングタイマカウンタ302(図3)のカウンタ値として読み取って記録する。前述したように、ディレイタイム割込み609は、D S P 2 0 6におけるディレイ処理において同期拍数に設定されたディレイタイムがサンプリングクロックによって一周期分カウントされる毎に、D S P 2 0 6からC P U 2 0 1に通知される。

## 【0064】

以上のようにして、C P U 2 0 1内で同期拍数に応じた回数(=s回)分のシーケンスロック割込みが受け取られた時点の時刻(以下このC P U 2 0 1側の時刻を「L A S T \_ B E A T \_ T I M E」と表記)と、D S P 2 0 6から同期拍数に対応するディレイタイムに対するサンプリングクロックによる一周期分のカウントにより発生するディレイタイム割込み609が受け取られた時点の時刻(以下このD S P 2 0 6側の時刻を「L A S T \_ D E L A Y \_ T I M E」と表記)が揃うと、C P U 2 0 1は、両方の時刻の時間差dを、下記(9)式により計算する。

## 【0065】

d = L A S T \_ B E A T \_ T I M E - L A S T \_ D E L A Y \_ T I M E  
.. (9)

## 【0066】

C P U 2 0 1は、この時間差が小さくなるように、図3のシーケンスロックカウンタ301にセットするシーケンスロックカウント上限値を増加又は減少させる。具体的には、C P U 2 0 1は、時間差がプラス値である、すなわち、L A S T \_ B E A T \_ T I M EのほうがL A S T \_ D E L A Y \_ T I M Eよりも後の時刻の場合には、C P U 2 0 1内

のシーケンスロックカウンタ 301においてカウント上限値に達してシーケンスロック割込みが発生するまでの時間がかかりすぎているため、シーケンスロックカウント上限値から上記時間差のプラス値を 1 同期拍数 (= 1 ディレイタイム) 分に換算した値を減算して、シーケンスロックカウント上限値を小さくさせる。一方、時間差がマイナス値である、すなわち、LAST\_BEAT\_TIME のほうが LAST\_DELAY\_TIME よりも先の時刻の場合には、CPU201 内のシーケンスロックカウンタ 301においてカウント上限値に達してシーケンスロック割込みが発生するまでの時間が早すぎるため、シーケンスロックカウント上限値から上記時間差のマイナス値を 1 同期拍数 (= 1 ディレイタイム) 分に換算した値を減算して、シーケンスロックカウント上限値を大きくする。つまり、CPU201 は、上記(7)式によって計算される同期拍数 b と、上記(9)式によって計算される時間差 d とを用いて、下記(10)式によりシーケンスロックカウント上限値の補正值 c を計算する。

10

## 【0067】

$$c = d / (480 \times b) \quad \cdots (10)$$

## 【0068】

CPU201 は、このようにして算出した補正值 c をシーケンスロックカウント上限値から減算して、シーケンスロックカウンタ 301 にセットする。この結果、次回のディレイ処理においては、CPU201 においてシーケンスロック割込みに従ってディレイタイム分だけ進行する自動演奏の時間と、DSP206 におけるディレイタイム分のディレイ処理の時間とのずれが無くなることが期待される。以上の本実施形態による一連の制御処理を、ディレイ同期処理（処理同期手段）と呼ぶ。

20

## 【0069】

なお、上記(10)式で計算される補正值 c は、整数値であるので、少数以下の切り捨て処理をした場合は、補正值 c は時間差 d が同期シーケンスロック数に収まっている限り 0 となり、何の補正も行われないことになる。しかしながら、本実施形態で問題としているのは 10 msec、20 msec といった認知可能な蓄積されたずれであり、数百 μsec といった微小なずれは補正する意味がほとんど無いので、問題となることはない。例えば、同期拍数が 1/2 の場合、同期シーケンスロック数は 240 回であるが、シーケンスロックカウンタ 301 のカウンタ上限値であるこの同期シーケンスロック数を 1 増やすとシーケンスロック割込みの間隔は 1 μsec 長くなり、次のディレイとの同期タイミングまでには、この 240 倍の 240 μsec を補正することになる。240 μsec 未満のズレの場合は何も行わないが、音楽的には認知不能な領域であるので問題ない。図 8 は、本実施形態の動作説明図である。図 8(a) は、指定テンポに対して所定の同期拍数として「一拍」がディレイタイムとして設定されたサンプルルーパーのディレイ処理が DSP206 で実行されることにより、同じ音響信号波形が繰返し再生される様子を示している。図 8(B) は、「0」から「一拍」に対応して設定されたディレイタイムサンプリング数 608 を、第 1 のカウント値、つまり上限カウント値までのカウントを繰り返すディレイタイムカウンタ 601 のカウント状態を表わす。このディレイタイムカウンタ 601 のカウント値が上限カウント値に到達するタイミングに応答して音響信号波形が繰返し再生されるとともに、図 8(C) に示すように、割り込み信号 609 が発生する。

30

これに対して、CPU201 では、指定テンポに対して生成されるシーケンスロックに従って自動演奏制御処理が実行される。より具体的には、シーケンスロックカウンタ 301 が、シーケンスロックを「0」から「一拍」に対応して設定されたシーケンスロックのカウント上限値までカウントを繰り返すことにより、この自動演奏の拍のタイミングが得られる。しかしながら、このシーケンスロックと上述のサンプリングクロックとは周期も異なり、かつ同期もしていないため、本実施形態のディレイ同期処理が実行されない場合には、音響信号波形が繰り返されるに従い、シーケンスロックカウンタ 301 のカウントが第 2 のカウント値、つまりカウント上限値まで到達するタイミングが、図 8(D) に示されるように、音響信号波形の繰返しタイミング（図 8 中の破線のタイミン

40

50

グ)に対して徐々にずれていってしまう。

このため、このタイミングを自動演奏の拍タイミングとして自動演奏を進行させると、自動演奏の拍タイミングと音響信号波形の繰り返しタイミングとのずれが大きくなってしまう(図8(E))。

一方、本実施形態のディレイ同期処理が実行された場合、割り込み信号609の発生タイミングとシーケンスロックカウンタ301がカウント上限値までカウントするタイミングとの差分時間を求め、この差分時間が少なくなるように、シーケンスロックカウンタ301がカウント上限値を増加あるいは減少させている(図8(F))。

これにより、シーケンスロックカウンタ301のカウントがカウント上限値まで到達するタイミングである自動演奏の拍タイミングは、図8(G)に示されるように、音響信号波形の繰返しタイミングに対して常に同期させることができる。このようにして、本実施形態では、CPU201の負荷が軽いにもかかわらず、DSP206でのディレイ処理に対する長期的なずれを回避することできるのである。10

#### 【0070】

図2のCPU201がCPU ROM202に記憶されたディレイ同期プログラムに従って、以上の基本動作を実現するディレイ同期処理を含む電子楽器制御処理について、以下に詳細に説明する。

#### 【0071】

図9(a)は、CPU ROM202が記憶する、前述した図5(a)に例示したTEMPO\_COUNT\_TBL(テンポ・カウントテーブル)と、前述した図7(a)に例示したSYNC\_BEAT\_TBL(同期拍数テーブル)の各データ構成例(定数値)を示す図である。図9(b)は、CPU RAM203に記憶される、CPU201による主にディレイ同期処理において使用される主要変数の一覧を示す図である。図10(a)は、前述したCPU201内のフリーランニングタイマカウンタ302の値を示すレジスタCPU\_FREE\_TIMERと、CPU201内のシーケンスロックカウンタ301に設定されるシーケンスロックカウント上限値を示すレジスタCPU\_TIMER\_COUNTを示す図である。図10(b)は、CPU201がDSP206と通信する場合のDSP206側のレジスタの一覧を示す図である。これらのレジスタは、CPU\_I/F215(図2参照)により、CPU201側からはメモリとしてアクセスすることができる。以上の各データ構成の詳細については、後述する。2030

#### 【0072】

図11は、本実施形態において、CPU201が実行する電子楽器制御処理の全体処理例を示すフローチャートである。

#### 【0073】

電源投入後、CPU201は、まず初期化処理を実行する(ステップS1101)。この初期化処理の詳細については、図12のフローチャートの説明で後述する。

#### 【0074】

初期化処理の後、CPU201は、ステップS1102からステップS1110までの各処理を順番に繰り返す無限ループに入る。CPU201はまず、ユーザーインターフェース処理(以下「ユーザI/F処理」と表記)を実行し、図1の鍵盤101、機能選択操作子102、音色選択ボタン103、ベンダ/モジュレーション・ホイール104に対するユーザによる操作を検知する(ステップS1102)。40

#### 【0075】

次にCPU201は、ステップS1102でのユーザI/F処理の結果、ユーザが図1の機能選択操作子102のTEMPOエリアのDOWNボタン又はUPボタンを操作することにより、テンポ設定イベントが発生したか否かを判定する(ステップS1103)。この判定結果が肯定的(以下「Yes」と表記)の場合には、CPU201は、テンポ設定処理を実行する(ステップS1104)。このとき、CPU201は、ユーザが機能選択操作子102のTEMPOエリアのDOWNボタン又はUPボタンを押下することにより、キー・スキヤナ211を介して取得したテンポ変化量を、CPU RAM203上の50

変数 D にセットする。UP ボタンが押下された場合には D = + 1 がセットされ、DOWN ボタンが押下された場合は D = - 1 がセットされる。テンポ設定処理の詳細については、図 13 のフローチャートの説明で後述する。ステップ S 1103 の判定結果が否定的（以下「No」と表記）の場合には、CPU201 は、ステップ S 1104 のテンポ設定処理は実行せずにスキップする。

#### 【0076】

次に CPU201 は、ステップ S 1102 でのユーザ I/F 処理の結果、ユーザが機能選択操作子 102 の DISPLAY エリアの各ボタン又はノブを操作することにより、ディレイ設定イベントが発生したか否かを判定する（ステップ S 1105）。この判定結果が Yes の場合には、CPU201 は、ディレイ設定処理を実行する（ステップ S 1106）。このとき、CPU201 は、ユーザが機能選択操作子 102 の DISPLAY エリアの各ボタン又はノブを操作することにより、キー・スキャナ 211 を介して取得した変更されたボタン又はノブに対応する操作種別とその値をそれぞれ、CPU RAM203 上の変数 p 及び変数 v にセットする。ディレイ設定処理の詳細については、図 14～図 16 のフローチャートの説明で後述する。ステップ S 1106 の判定結果が No の場合には、CPU201 は、ステップ S 1106 のディレイ設定変更処理は実行せずにスキップする。

#### 【0077】

次に CPU201 は、ステップ S 1102 でのユーザ I/F 処理の結果、ユーザが鍵盤 101 を操作することにより演奏イベントが発生したか否か、又は MIDI I/F 214 を介して押鍵又は離鍵の MIDI 入力が受信されたか否かを判定する（ステップ S 1107）。この判定結果が Yes の場合には、CPU201 は、押鍵 / 離鍵処理を実行する（ステップ S 1108）。CPU201 は、例えばユーザが鍵盤 101 を操作することによりキー・スキャナ 211 を介して取得した音高情報及びベロシティ情報に基づいて、又は MIDI I/F 214 を介して取得したノートオンイベントの MIDI データの音高情報及びベロシティ情報に基づいて、音源 LSI204 内の波形発生器 205 に対して、発音指示であるノートオンイベントや、消音指示であるノートオフイベントを発行する。この処理は、一般的な処理であるため、その詳細は省略する。ステップ S 1107 の判定結果が No の場合には、CPU201 は、ステップ S 1108 のディレイ設定変更処理は実行せずにスキップする。

#### 【0078】

次に CPU201 は、自動演奏定常処理を実行する（ステップ S 1109）。自動演奏定常処理は、前述したシーケンスロック割込みに基づいて、自動演奏を進行させる処理である。自動演奏定常処理の詳細については、図 17 のフローチャートの説明で後述する。

#### 【0079】

最後に CPU201 は、音源定常処理を実行する（ステップ S 1110）。音源定常処理では、例えば図 1 の音色選択ボタン 103 が押された場合に対応する音色変更の処理や、図 1 のベンダ / モジュレーション・ホール 104 が操作された場合に対応するベロシティ変更やピッチ変更の処理等の、音源 LSI204 に対して指示を行う処理が実行される。この処理は、一般的な処理であるため、その詳細は省略する。

#### 【0080】

図 12 は、図 11 のステップ S 1101 の初期化処理の詳細例を示すフローチャートである。

#### 【0081】

図 12において、CPU201 はまず、CPU I/F 215 を介して、ディレイ処理を実行する音源 LSI204 内の DSP206 のレジスタ DSP\_DELAY\_INIT（図 10(b) 参照）に、値 1 を書き込む（ステップ S 1201）。DSP206 は、レジスタ DSP\_DELAY\_INIT の値が 1 になると、DSP206 に接続される DSP RAM208（遅延メモリ）の内容や、ライトポインタ生成部 605 で生成されるポインタ値等を初期化し、レジスタ DSP\_DELAY\_INIT の値を 0 に戻す。

10

20

30

40

50

## 【0082】

次にCPU201は、CPU RAM203の変数領域の各変数値（図9（b）の一覧を参照）を0に初期化する（ステップS1202）。この結果、自動演奏を行っている状態か否かを示す変数SEQ\_RUN（図9（b）参照）の値が0に初期化されることにより、自動演奏は停止状態にされる。また、ディレイホールドモードか否かを示す変数DELAY\_HOLD（図9（b）参照）の値が0に初期化されることにより、ディレイホールドモードはオフにされる。更に、ディレイテンポシンクロナイズドモードか否かを示す変数DELAY\_SYNC（図9（b）参照）の値が0に初期化されることにより、ディレイテンポシンクロナイズドモードはオフにされる。また自動演奏を司るシーケンスクロック割込みの割込み回数を示すシーケンスカウンタ変数SEQ\_CLOCK（図9（b）参照）の値が0に初期化される。なお、テンポを示す変数TEMPO（図9（b）参照）については、初期値として120（BPM）が設定される。10

## 【0083】

その後、CPU201は、テンポ設定処理を実行する。このとき、CPU201は、テンポ変化量=0をCPU RAM203上の変数Dにセットする。テンポ設定処理の詳細については、図13のフローチャートの説明で後述するが、この処理において、CPU201は、CPU RAM203の変数TEMPO（図9（b）参照）に設定されたテンポ初期値120（BPM）によって、CPU ROM202に記憶されるTEMPO\_COUNT\_TBL（図9（a））を参照することにより、SEQ\_CLOCK\_COUNT項目の値=1042を取得し（図5（a）参照）、その値をCPU201のレジスタCPU\_TIMER\_COUNT（図10（a）参照）にセットする（後述する図13のステップS1306参照）。このレジスタCPU\_TIMER\_COUNTの値として、CPU201内のシーケンスクロックカウンタ301にシーケンスクロックカウント上限値=1042が初期設定される。20

## 【0084】

次にCPU201は、図1の機能選択操作子102のDELAYエリアのTIMEノブの操作位置を、図2のキー・スキャナ211を介して読み込み、ディレイタイムを示すCPU RAM203上の変数DELAY\_TIME（図9（b）参照）に格納する（ステップS1204）。ディレイタイムの値域は、0～FFFFFH（「H」は16進数を表す）の範囲であり、この範囲が0～2000msの時間範囲に対応する。従って、DELAY\_TIMEをmsで表現するためには、下記（11）式の変換を行えばよい。30

## 【0085】

$$\text{ディレイタイム(ms)} = \text{DELAY\_TIME} \times 2000 / \text{FFFFFH} \quad \dots (11)$$

## 【0086】

次にCPU201は、図1の機能選択操作子102のDELAYエリアのREPEATノブの操作位置を、図2のキー・スキャナ211を介して読み込み、ディレイのフィードバック量を示すCPU RAM203上の変数DELAY\_FEEDBACK（図9（b）参照）に格納する（ステップS1205）。

## 【0087】

次にCPU201は、図1の機能選択操作子102のDELAYエリアのLEVELノブの操作位置を、図2のキー・スキャナ211を介して読み込み、ディレイ音のレベルを示すCPU RAM203上の変数DELAY\_LEVEL（図9（b）参照）に格納する（ステップS1206）。

## 【0088】

次にCPU201は、CPU RAM203上のパラメータ変数pに、ディレイホールドモードを示す文字列‘DELAY\_HOLD’を設定し、値変数vにディレイホールドモードでないことを示す値0を設定して、ディレイ設定処理を実行する（ステップS1207）。p = ‘DELAY\_HOLD’によりディレイ設定処理の詳細を示す図14のフローチャートにおいて後述するステップS1402が実行されることにより図15（a）4050

のフローチャートで示されるディレイホールドモード設定処理( B E A T 処理)が実行される。これにより、ディレイホールドモードでない状態に対応して、ディレイ入力音量調整用のアンプ 4 0 2 ( 図 4 ( a ) 参照) のゲインを示す D S P 2 0 6 のレジスタ D S P \_ D E L A Y \_ I N P U T ( 図 1 0 ( b ) 参照) に、値 F F F F H ( 倍率 1 . 0 ) が初期設定される。また、フィードバック量調整用のアンプ 4 0 4 ( 図 4 ( a ) 参照) のゲインを示す D S P 2 0 6 のレジスタ D S P \_ D E L A Y \_ F E E D B A C K に、図 1 2 のステップ S 1 2 0 5 で C P U R A M 2 0 3 上の変数 D E L A Y \_ F E E D B A C K に設定された R E P E A T ノブの初期操作値が初期設定される( 後述する図 1 5 ( a ) のステップ S 1 5 0 3 参照)。

## 【 0 0 8 9 】

次に C P U 2 0 1 は、 C P U R A M 2 0 3 上のパラメータ変数 p に、ディレイテンポシンクロナイズドモードを示す文字列 ‘ D E L A Y \_ S Y N C ’ を設定し、値変数 v にディレイテンポシンクロナイズドモードでないことを示す値 0 を設定して、ディレイ設定処理を実行する( ステップ S 1 2 0 8 )。 p = ‘ D E L A Y \_ S Y N C ’ によりディレイ設定処理の詳細を示す図 1 4 のフローチャートにおいて後述するステップ S 1 4 0 6 が実行されることにより図 1 6 ( a ) のフローチャートで示されるディレイテンポシンクロナイズドモード設定処理( S Y N C 処理)が実行される。これにより、ディレイシンクロナイズドモードでない状態に対応して、前述したディレイタイムサンプリング数レジスタ 6 0 2 ( 図 6 参照) に対応する D S P 2 0 6 のレジスタ D S P \_ D E L A Y \_ S A M P L E ( 図 1 0 ( b ) 参照) に、図 1 2 のステップ S 1 2 0 4 で C P U R A M 2 0 3 上の変数 D E L A Y \_ T I M E に設定された T I M E ノブの初期操作値に対応する時間値を D S P 2 0 6 の 4 4 . 1 K H z のサンプリングクロックでサンプリングしたときのサンプル数が初期設定される( 前述した( 1 1 ) 式と後述する図 1 6 ( a ) のステップ S 1 6 0 4 参照)。

。

## 【 0 0 9 0 】

次に C P U 2 0 1 は、 C P U R A M 2 0 3 上のパラメータ変数 p に、ディレイタイムを示す文字列 ‘ D E L A Y \_ T I M E ’ を設定し、値変数 v に C P U R A M 2 0 3 上の変数 D E L A Y \_ T I M E が示す値を設定して、ディレイ設定処理を実行する( ステップ S 1 2 0 9 )。 p = ‘ D E L A Y \_ T I M E ’ によりディレイ設定処理の詳細を示す図 1 4 のフローチャートにおいて後述するステップ S 1 4 0 3 が実行されることにより図 1 5 ( b ) のフローチャートで示されるディレイタイム設定処理( T I M E 処理)が実行される。これにより、前述したディレイタイムサンプリング数レジスタ 6 0 2 ( 図 6 参照) に対応する D S P 2 0 6 のレジスタ D S P \_ D E L A Y \_ S A M P L E ( 図 1 0 ( b ) 参照) に、 C P U R A M 2 0 3 上の変数 D E L A Y \_ T I M E に得られている T I M E ノブの操作値に対応する時間値を D S P 2 0 6 の 4 4 . 1 K H z のサンプリングクロックでサンプリングしたときのサンプル数が初期設定される( 前述した( 1 1 ) 式と後述する図 1 5 ( b ) のステップ S 1 5 1 1 及び S 1 5 1 3 参照)。

## 【 0 0 9 1 】

次に C P U 2 0 1 は、 C P U R A M 2 0 3 上のパラメータ変数 p に、ディレイフィードバックを示す文字列 ‘ D E L A Y \_ F E E D B A C K ’ を設定し、値変数 v に C P U R A M 2 0 3 上の変数 D E L A Y \_ F E E D B A C K が示す値を設定して、ディレイ設定処理を実行する( ステップ S 1 2 1 0 )。 p = ‘ D E L A Y \_ F E E D B A C K ’ によりディレイ設定処理の詳細を示す図 1 4 のフローチャートにおいて後述するステップ S 1 4 0 4 が実行されることにより図 1 5 ( c ) のフローチャートで示されるディレイフィードバック設定処理( F E E D B A C K 処理)が実行される。これにより、フィードバック量調整用のアンプ 4 0 4 ( 図 4 ( a ) 参照) のゲインを示す D S P 2 0 6 のレジスタ D S P \_ D E L A Y \_ F E E D B A C K ( 図 1 0 ( b ) 参照) に、図 1 2 のステップ S 1 2 0 5 で C P U R A M 2 0 3 上の変数 D E L A Y \_ F E E D B A C K に設定された R E P E A T ノブの初期操作値に対応するフィードバック量が初期設定される。

## 【 0 0 9 2 】

10

20

30

40

50

更にCPU201は、CPU RAM203上のパラメータ変数pに、ディレイレベルを示す文字列‘DELAY\_LEVEL’を設定し、値変数vにCPU RAM203上の変数DELAY\_LEVELが示す値を設定して、ディレイ設定処理を実行する（ステップS1211）。p = ‘DELAY\_LEVEL’によりディレイ設定処理の詳細を示す図14のフローチャートにおいて後述するステップS1405が実行されることにより図15（d）のフローチャートで示されるディレイレベル設定処理（LEVEL処理）が実行される。これにより、ディレイ出力音量調整用のアンプ403（図4（a）参照）のゲインを示すDSP206のレジスタDSP\_DELAY\_OUTPUT（図10（b）参照）に、図12のステップS1206でCPU RAM203上の変数DELAY\_LEVELに設定されたLEVELノブの初期操作値に対応するレベル量が初期設定される。

10

#### 【0093】

最後に、CPU201は、本実施形態のディレイ同期処理以外に関連するCPU RAM203上の変数や音源LSI204上のレジスタ等を初期化する、その他の初期化処理を実行する（ステップS1212）。その後、CPU201は、図12のフローチャートで示される図11のステップS1101の初期化処理を終了する。

#### 【0094】

図13は、図11のステップS1104又はステップS1101の初期化処理内の図12のステップS1203で実行されるテンポ設定処理の詳細例を示すフローチャートである。

20

#### 【0095】

図13において、CPU201は、CPU RAM203上の変数TEMPOに格納されているテンポ値を、CPU RAM203上の変数Dで引き渡されたテンポ操作量の値だけ増減（）させることにより、テンポ値を変更する（ステップS1301）。ユーザが図1の機能選択操作子102のTEMPOエリアのUPキーを押下した場合には、D = +1が引き渡されることにより変数TEMPOの値が+1される。ユーザがTEMPOエリアのDOWNキーを押下した場合には、D = -1が引き渡されることにより変数TEMPOの値が-1される。

#### 【0096】

次にCPU201は、更新された変数TEMPOの値が最低値30を下回ったか否かを判定し（ステップS1302）、ステップS1302の判定がYesならば、変数TEMPOの値を最低値30に設定する（ステップS1303）。ステップS1302の判定がNoならば、CPU201は、ステップS1303は実行せずにスキップする。

30

#### 【0097】

次にCPU201は、更新された変数TEMPOの値が最高値300を上回ったか否かを判定し（ステップS1304）、ステップS1304の判定がYesならば、変数TEMPOの値を最高値300に設定する（ステップS1305）。ステップS1304の判定がNoならば、CPU201は、ステップS1305は実行せずにスキップする。

#### 【0098】

続いてCPU201は、ステップS1301からS1305で更新された変数TEMPOの値によって、CPU ROM202に記憶されるTEMPO\_COUNT\_TBL（図9（a））を参照することにより、TEMPO値に対応するエントリのSEQ\_CLOCK\_COUNT項目の値を取得し（図5（a）参照）、その値をCPU201のレジスタCPU\_TIMER\_COUNT（図10（a）参照）にセットする（ステップS1306）。このレジスタCPU\_TIMER\_COUNTの更新値として、CPU201内のシーケンスロックカウンタ301にシーケンスロックカウント上限値が更新される。

40

#### 【0099】

その後、CPU201は、CPU RAM203上の変数DELAY\_SYNCの値が1であるか否か、すなわち、ディレイホールドモードが現在設定されているか否かを判定

50

する（ステップS1307）。

【0100】

ステップS1307の判定がYesならば、CPU201はまず、ステップS1301からS1305で更新された変数TEMPOの値によって、CPUROM202に記憶されるTEMPO\_COUNT\_TBL（図9（a））を参照することにより、TEMPO値に対応するエントリのDELAY\_COUNT項目値、すなわち、ディレイタイムを更新テンポ値の1拍に同期させたときのサンプリングクロック数をTEMPO\_COUNT\_TBL(TEMPO).DELAY\_COUNTとして取得し、CPURAM203上の変数aに格納する。次に、CPU201は、ユーザの操作によって図1の機能選択操作子102のDELAYエリアのBEATノブによって選択され、CPURAM203上の変数DELAY\_SYNC\_BEATに格納されている設定値によって、CPUROM202に記憶されているSYNC\_BEAT\_TBL（図9（a））を参照することにより、DELAY\_SYNC\_BEAT値に対応するエントリのNUMERATOR項目値とDENOMINATOR項目値をそれぞれSYNC\_BEAT\_TBL(DELAY\_SYNC\_BEAT).NUMERATOR及びSYNC\_BEAT\_TBL(DELAY\_SYNC\_BEAT).DENOMINATORとして取得し、それぞれCPURAM203上の変数n及びdに格納する（以上、ステップS1308）。

【0101】

そして、CPU201は、下記（12）式により、前述したディレイタイムカウンタ601のカウンタ上限値であるディレイタイムサンプリング数608（図6）を算出し、前述したディレイタイムサンプリング数レジスタ602（図6参照）に対応するDSP206のレジスタDSP\_DELAY\_SAMPLE（図10（b）参照）に格納する（ステップS1309）。

【0102】

$$DSP\_DELAY\_SAMPLE = a \times n / d \quad \dots (12)$$

【0103】

ステップS1308とS1309により、指定されたテンポの指定された同期拍数のディレイタイムに対応するディレイタイムサンプリング数608が算出されて、DSP206のレジスタDSP\_DELAY\_SAMPLEに格納される。その後、CPU201は、図13のフローチャートで示されるテンポ設定処理を終了する。

【0104】

ステップS1307の判定がNoならば、CPU201は、ステップS1308とS1309は実行せずにスキップし、図13のフローチャートで示されるテンポ設定処理を終了する。

【0105】

図14は、図11のステップS1106又はステップS1101の初期化処理内の図12のステップS1207からS1211で実行されるディレイ設定処理の詳細例を示すフローチャートである。

【0106】

図14において、CPU201はまず、CPURAM203上の変数pで引き渡された設定処理種別、すなわちユーザによる図1の機能選択操作子102のDELAYエリアでの操作種別を判別する（ステップS1401）。

【0107】

CPU201は、変数p = 'DELAY\_HOLD'、すなわちHOLDボタンが押下されたならばステップS1402のディレイホールドモード設定処理（HOLD処理）を実行する。このとき、HOLDボタンのLEDがオフの状態で更にHOLDボタンが押下された場合には、変数vにディレイホールドモードがオフからオンになったことを示す値1が引き渡される。逆に、HOLDボタンのLEDがオンの状態で更にHOLDボタンが押下された場合には、変数vにディレイホールドモードがオンからオフになったことを示す値1が引き渡される。

10

20

30

40

50

**【0108】**

CPU201は、変数p = 'DELAY\_TIME'、すなわちTIMEノブが操作されたならばステップS1403のディレイタイム設定処理(TIME処理)を実行する。このとき、変数vには、TIMEノブの操作位置に対応する0～FFFFFHの範囲の値が引き渡される。

**【0109】**

CPU201は、変数p = 'DELAY\_FEEDBACK'、すなわちREPEATノブが操作されたならばステップS1404のディレイフィードバック設定処理(FEEDBACK処理)を実行する。このとき、変数vには、REPEATノブの操作位置に対応する0～FFFFFHの範囲の値が引き渡される。

10

**【0110】**

CPU201は、変数p = 'DELAY\_LEVEL'、すなわちLEVELノブが操作されたならばステップS1405のディレイレベル設定処理(LEVEL処理)を実行する。このとき、変数vには、LEVELノブの操作位置に対応する0～FFFFFHの範囲の値が引き渡される。

**【0111】**

CPU201は、p = 'DELAY\_SYNC'、すなわちSYNCボタンが操作されたならばステップS1406のディレイテンポシンクロナイズドモード設定処理(SYNC処理)を実行する。このとき、SYNCボタンのLEDがオフの状態で更にSYNCボタンが押下された場合には、変数vにディレイテンポシンクロナイズドモードがオフからオンになったことを示す値1が引き渡される。逆に、SYNCボタンのLEDがオンの状態で更にSYNCボタンが押下された場合には、変数vにディレイテンポシンクロナイズドモードがオンからオフになったことを示す値1が引き渡される。

20

**【0112】**

CPU201は、p = 'DELAY\_BEAT'、すなわちBEATノブが操作されたならばステップS1407のディレイテンポ同期拍数設定処理(BEAT処理)を実行する。このとき、変数vには、BEATノブの操作位置に対応する0、1、2、3、4、5、6、7の何れかの設定値が引き渡される。

**【0113】**

各設定処理が終了した後、CPU201は、図14のフローチャートで示されるディレイ設定処理を終了する。

30

**【0114】**

図15(a)は、図14のステップS1402のディレイホールドモード設定処理(HOLD処理)の詳細例を示すフローチャートである。この処理は、ユーザが図1の機能選択操作子102のDELAYエリアのHOLDボタンを押下した場合に実行される。ユーザは、HOLDボタンを押下してそのLEDを点灯又は消灯させることにより、前述のディレイホールドモードのオン/オフを切り替えることができる。このとき、前述したように、HOLDボタンのLEDが消灯されている状態で更にHOLDボタンが押下された場合には、変数vにディレイホールドモードがオフからオンになったことを示す値1が引き渡される。逆に、HOLDボタンのLEDが点灯されている状態で更にHOLDボタンが押下された場合には、変数vにディレイホールドモードがオンからオフになったことを示す値1が引き渡される。

40

**【0115】**

CPU201はまず、変数vで引き渡された値を、ディレイホールドモードか否かを示す変数DELAY\_HOLD(図9(b)参照)にセットする(ステップS1501)。

**【0116】**

次にCPU201は、ステップS1501で変数DELAY\_HOLDにセットされた値を判定する(ステップS1502)。

**【0117】**

CPU201は、ステップS1502で、変数DELAY\_HOLDにセットされた値

50

が0である、すなわち、ディレイホールドモードがオフされたと判定した場合には、ディレイ入力音量調整用のアンプ402（図4（a）参照）のゲインを示すDSP206のレジスタDSP\_DELAY\_INPUT（図10（b）参照）に、値FFF FH（倍率1.0）を設定する。これにより、ディレイ入力音量調整用のアンプ402の増幅率が1.0倍に設定され、それ以降入力された音響信号407にディレイ効果がかかる状態になる。また、CPU201は、フィードバック量調整用のアンプ404（図4（a）参照）のゲインを示すDSP206のレジスタDSP\_DELAY\_FEEDBACK（図10（b）参照）に、CPU\_RAM203上の変数DELAY\_FEEDBACKに設定されているフィードバック量を設定する（以上、ステップS1503）。これにより、図1の機能選択操作子102のDELAYエリアのREPEATノブによるフィードバック調整が有効となる。その後、CPU201は、図15（a）のフローチャートで示される図14のステップS1402のディレイホールドモード設定処理（HOLD処理）を終了する。  
10

#### 【0118】

CPU201は、ステップS1502で、変数DELAY\_HOLDにセットされた値が1である、すなわちディレイホールドモードがオンされたと判定した場合には、ディレイ入力音量調整用のアンプ402（図4（a）参照）のゲインを示すDSP206のレジスタDSP\_DELAY\_INPUT（図10（b）参照）に、値0を設定する。つまり、図4（a）において、ディレイホールドモード時には、遅延装置401に新たな音響信号の入力407は流入しないように設定される。また、CPU201は、フィードバック量調整用のアンプ404（図4（a）参照）のゲインを示すDSP206のレジスタDSP\_DELAY\_FEEDBACK（図10（b）参照）に、値FFF FH（倍率1.0）を設定する。すなわち、図4（a）において、ディレイホールドモード時には、遅延装置401から出力される音響信号が全て遅延装置401の入力側にフィードバックされる。これにより、前述したサンプルルーパーの機能が実現される。その後、CPU201は、図15（a）のフローチャートで示される図14のステップS1402のディレイホールドモード設定処理（HOLD処理）を終了する。  
20

#### 【0119】

図15（b）は、図14のステップS1403のディレイタイム設定処理（TIME処理）の詳細例を示すフローチャートである。この処理は、ユーザが図1の機能選択操作子102のDELAYエリアのTIMEノブを操作した場合に実行される。ユーザは、DELAYエリアのSYNCボタンのLEDを消灯させてディレイテンポシンクロナイズドモードをオフしているときに、TIMEノブの操作により、ディレイタイムを直接、例えば、0～2secの間で指定することができる。このとき、前述したように、変数vには、TIMEノブの操作位置に対応する0～FFF FHの範囲の値が引き渡される。  
30

#### 【0120】

CPU201はまず、前述した（11）式により、変数vで引き渡された変数DELAY\_TIMEの値を、16進値から msec 値に変換して、新たに変数vに格納する（ステップS1511）。

#### 【0121】

次にCPU201は、CPU\_RAM203上の変数DELAY\_SYNCの値を判別する（ステップS1512）。

#### 【0122】

CPU201は、ステップS1512で、変数DELAY\_SYNCにセットされた値が0である、すなわち、ディレイテンポシンクロナイズドモードがオフであると判定した場合には、下記（13）式を実行する（ステップS1513）。

#### 【0123】

$$\begin{aligned} \text{DSP_DELAY_SAMPLE} = \\ (\text{v} / 1000) \times 44100 \quad \dots (13) \end{aligned}$$

#### 【0124】

10

20

30

40

50

これにより、前述したディレイタイムサンプリング数レジスタ602(図6参照)に対応するDSP206のレジスタDSP\_DELAY\_SAMPLE(図10(b)参照)に、CPU RAM203上の変数DELAY\_TIMEに得られているTIMEノブの操作値に対応する時間値をDSP206の44.1KHzのサンプリングクロックでサンプリングしたときのサンプル数が格納される。その後、CPU201は、図15(b)のフローチャートで示される図14のステップS1403のディレイタイム設定処理(TIME処理)を終了する。

#### 【0125】

CPU201は、ステップS1512で、変数DELAY\_SYNCにセットされた値が1である、すなわち、ディレイテンポシンクロナイズドモードがオンであると判定した場合には、前述したようにディレイタイムはテンポに同期して決定されることになり、TIMEノブの操作は無視されるため、そのまま図15(b)のフローチャートで示される図14のステップS1403のディレイタイム設定処理(TIME処理)を終了する。10

#### 【0126】

図15(c)は、図14のステップS1404のディレイフィードバック設定処理(FEEDBACK処理)の詳細例を示すフローチャートである。この処理は、ユーザが図1の機能選択操作子102のDELAYエリアのREPEATノブを操作した場合に実行される。ユーザは、DELAYエリアのHOLDボタンのLEDを消灯させていてディレイホールドモードをオフしているときに、REPEATノブの操作により、ディレイのフィードバック量を調整することができる。ここで指定された値により、図4(a)のフィードバック量調整用のアンプ404のゲインが決定される。このとき、前述したように、変数vには、REPEATノブの操作位置に対応する0~FFFHの範囲の値が引き渡される。20

#### 【0127】

CPU201はまず、変数vで引き渡された値を変数DELAY\_FEEDBACKに格納する(ステップS1521)。

#### 【0128】

次にCPU201は、CPU RAM203上の変数DELAY\_HOLDの値を判別する(ステップS1522)。

#### 【0129】

CPU201は、ステップS1522で、変数DELAY\_HOLDにセットされた値が0である、すなわち、ディレイホールドモードがオフであると判定した場合には、DSP206上のフィードバック量調整用のアンプ404(図4(a)参照)のゲインを示すDSP206のレジスタDSP\_DELAY\_FEEDBACK(図10(b)参照)に、変数DELAY\_FEEDBACKに設定されているフィードバック量を設定する(以上、ステップS1523)。その後、CPU201は、図15(c)のフローチャートで示される図14のステップS1404のディレイフィードバック設定処理(FEEDBACK処理)を終了する。30

#### 【0130】

CPU201は、ステップS1522で、変数DELAY\_HOLDにセットされた値が1である、すなわち、ディレイホールドモードがオンであると判定した場合には、DSP206のレジスタDSP\_DELAY\_FEEDBACKには、図15(a)のディレイホールドモード設定処理(HOLD処理)のステップS1504で最大値FFFHが設定されるため、CPU201は、ステップS1523の処理は実行せずにスキップし、図15(c)のフローチャートで示される図14のステップS1404のディレイフィードバック設定処理(FEEDBACK処理)を終了する。40

#### 【0131】

図15(d)は、図14のステップS1405のディレイレベル設定処理(LEVEL処理)の詳細例を示すフローチャートである。この処理は、ユーザが図1の機能選択操作子102のDELAYエリアのLEVELノブを操作した場合に実行される。ユーザは、50

LEVELノブの操作により、ディレイ信号のレベルを調整することができる。ここで指定された値により、図4(a)のディレイ出力音量調整用のアンプ403のゲインが決定される。このとき、前述したように、変数vには、LEVELノブの操作位置に対応する0~FFFFHの範囲の値が引き渡される。

#### 【0132】

CPU201はまず、変数vで引き渡された値を変数DELAY\_LEVELに格納する(ステップS1531)。

#### 【0133】

次にCPU201は、ディレイ出力音量調整用のアンプ403(図4(a)参照)のゲインを示すDSP206のレジスタDSP\_DELAY\_OUTPUT(図10(b)参照)に、変数DELAY\_LEVELに設定されたLEVELノブの操作値に対応するレベル量を設定する(ステップS1532)。その後、CPU201は、図15(c)のフローチャートで示される図14のステップS1405のディレイレベル設定処理(LEVEL処理)を終了する。

10

#### 【0134】

図16(a)は、図14のステップS1406のディレイテンポシンクロナイズドモード設定処理(SYNC処理)の詳細例を示すフローチャートである。この処理は、ユーザが図1の機能選択操作子102のDELAYエリアのSYNCボタンを押下した場合に実行される。ユーザは、SYNCボタンを押下してそのLEDを点灯又は消灯させることにより、前述のディレイテンポシンクロナイズドモードのオン/オフを切り替えることができる。このとき、前述したように、SYNCボタンのLEDが消灯されている状態で更にSYNCボタンが押下された場合には、変数vにディレイテンポシンクロナイズドモードがオフからオンになったことを示す値1が引き渡される。逆に、SYNCボタンのLEDが点灯されている状態で更にSYNCボタンが押下された場合には、変数vにディレイテンポシンクロナイズドモードがオンからオフになったことを示す値1が引き渡される。

20

#### 【0135】

CPU201はまず、変数vで引き渡された値を、ディレイテンポシンクロナイズドモードか否かを示す変数DELAY\_SYNC(図9(b)参照)にセットする(ステップS1601)。

#### 【0136】

30

次にCPU201は、ステップS1601で変数DELAY\_SYNCにセットされた値を判定する(ステップS1602)。

#### 【0137】

CPU201は、ステップS1602で、変数DELAY\_SYNCにセットされた値が1である、すなわち、ディレイテンポシンクロナイズドモードがオンされたと判定した場合には、以下の動作を実行する(ステップS1603)。CPU201はまず、変数TEMPOに設定されている現在のテンポ値によって、CPU\_ROM202に記憶されるTEMPO\_COUNT\_TBL(図9(a))を参照することにより、TEMPO値に対応するエントリのDELAY\_COUNT項目値、すなわち、ディレイタイムを更新テンポ値の1拍に同期させたときのサンプリングクロック数を、TEMPO\_COUNT\_TBL(TEMPO).DELAY\_COUNTとして取得する。また、CPU201は、ユーザの操作によって図1の機能選択操作子102のDELAYエリアのBEATノブによって選択され、CPU\_RAM203上の変数DELAY\_SYNC\_BEATに格納されている設定値によって、CPU\_ROM202に記憶されているSYNC\_BEAT\_TBL(図9(a))を参照することにより、DELAY\_SYNC\_BEAT値に対応するエントリのNUMERATOR項目値とDENOMINATOR項目値をそれぞれ、SYNC\_BEAT\_TBL(DELAY\_SYNC\_BEAT).NUMERATOR及びSYNC\_BEAT\_TBL(DELAY\_SYNC\_BEAT).DENOMINATORとして取得する。そして、CPU201は、下記(14)式により、前述したディレイタイムカウンタ601のカウンタ上限値であるディレイタイムサンプリング数

40

50

608(図6)を算出し、前述したディレイタイムサンプリング数レジスタ602(図6参照)に対応するDSP206のレジスタDSP\_DELAY\_SAMPLE(図10(b)参照)に格納する。

#### 【0138】

```
DSP_DELAY_SAMPLE = TEMPO_COUNT_TBL(TEMPO).DELAY_COUNT × SYNC_BEAT_TBL(DELAY_SYNC_BEAT).NUMERATOR / SYNC_BEAT_TBL(DELAY_SYNC_BEAT).DENOMINATOR . . . (14)
```

#### 【0139】

これにより、ディレイタイムを指定されたテンポ値(TEMPO)の1拍に同期させたときのサンプリングクロック数DELAY\_COUNTに、機能選択操作子102のDELAYエリアのBEATノブで設定された同期拍数の分周比率NUMERATOR/DENOMINATORを乗算して算出される、指定されたテンポの指定された同期拍数のディレイタイムに対応するディレイタイムサンプリング数が、DSP206でのディレイ処理におけるディレイタイムとして設定される。その後、CPU201は、図16(a)のフローチャートで示される図14のステップS1406のディレイテンポシンクロナイズドモード設定処理(SYNC処理)を終了する。10

#### 【0140】

CPU201は、ステップS1602で、変数DELAY\_SYNCにセットされた値が0である、すなわち、ディレイテンポシンクロナイズドモードがオフされたと判定した場合には、ユーザによる図1の機能選択操作子102のDELAYエリアのTIMEノブの操作が有効となるため、CPU201はまず、前述した(11)式の計算を含む下記(15)式を計算する(ステップS1604)。20

#### 【0141】

```
DSP_DELAY_SAMPLE =
{ (DELAY_TIME × 2000 / FFFFH) / 1000 } × 44100
. . . (15)
```

#### 【0142】

これにより、前述したディレイタイムサンプリング数レジスタ602(図6参照)に対応するDSP206のレジスタDSP\_DELAY\_SAMPLEに、CPU RAM203上の変数DELAY\_TIMEに得られているTIMEノブの操作値を16進値からmsec値に変換した上でその時間値をDSP206の44.1KHzのサンプリングクロックでサンプリングしたときのサンプル数が格納される。すなわち、TIMEノブの操作量に応じて、DSP206でのディレイ処理におけるディレイタイムが直接決定されることになる。その後、CPU201は、図16(a)のフローチャートで示される図14のステップS1406のディレイテンポシンクロナイズドモード設定処理(SYNC処理)を終了する。30

#### 【0143】

図16(b)は、図14のステップS1407のディレイテンポ同期拍数設定処理(BEAT処理)の詳細例を示すフローチャートである。この処理は、ユーザが図1の機能選択操作子102のDELAYエリアのBEATノブを操作した場合に実行される。ユーザは、BEATノブの操作により、ディレイテンポシンクロナイズドモードにおける同期拍数を指定することができる。このとき、前述したように、変数vには、BEATノブの操作位置に対応する0、1、2、3、4、5、6、7の何れかの設定値が引き渡される。40

#### 【0144】

CPU201はまず、変数vで引き渡された値を、同期拍数を格納する変数DELAY\_SYNC\_BEAT(図9(b)参照)にセットする(ステップS1611)。

#### 【0145】

次にCPU201は、変数DELAY\_SYNCにセットされている値を判定する(ステップS1612)。50

## 【0146】

CPU201は、ステップS1612で、変数DELAY\_SYNCにセットされた値が1である、すなわち、ディレイテンポシンクロナイズドモードがオンされていると判定した場合には、ディレイテンポシンクロナイズドモード設定処理(SYNC処理)において実行される図16(a)のステップS1603と同様の処理を実行する(ステップS1613)。これにより、ディレイタイムを指定されたテンポ値(TEMPO)の1拍に同期させたときのサンプリングクロック数DELAY\_COUNTに、機能選択操作子102のDELAYエリアのBEATノブで設定された同期拍数の分周比率NUMERATOR/DENOMINATORを乗算して算出される、指定されたテンポの指定された同期拍数のディレイタイムに対応するディレイタイムサンプリング数が、DSP206でのディレイ処理におけるディレイタイムとして、前述したディレイタイムサンプリング数レジスタ602(図6参照)に対応するDSP206のレジスタDSP\_DELAY\_SAMPLE(図10(b)参照)に格納される。その後、CPU201は、図16(b)のフローチャートで示される図14のステップS1407のディレイテンポ同期拍数設定処理(BEAT処理)を終了する。10

## 【0147】

CPU201は、ステップS1612で、変数DELAY\_SYNCにセットされた値が0である、すなわち、ディレイテンポシンクロナイズドモードがオフされていると判定した場合には、BEATノブの操作は無視されるため、そのまま図16(b)のフローチャートで示される図14のステップS1407のディレイテンポ同期拍数設定処理(BEAT処理)を終了する。20

## 【0148】

図17は、図11のステップS1109の自動演奏定常処理の詳細例を示すフローチャートである。ここでは、CPU201内のシーケンスロックカウンタ301(図3)からのシーケンスロック割込みに基づいて、自動演奏の進行制御が実施される。

## 【0149】

CPU201はまず、ユーザが図1の機能選択操作子102内の特に図示しない自動演奏指定スイッチを操作することにより、自動演奏が指定されているか否かを判定する。この指定の有無は、CPU\_RAM203上の変数SEQ\_RUNの値により判定できる。変数SEQ\_RUNの値が1でなければ、自動演奏制御処理は実施されないため、CPU201は、図17のフローチャートで示される図11のステップS1109の自動演奏定常処理をそのまま終了する。30

## 【0150】

変数SEQ\_RUNの値が1の場合、CPU201はまず、自動演奏を司るシーケンスロック割込みの通算の割込み回数を示す通算シーケンスロック割込みカウント変数SEQ\_CLOCK(図9(b)参照)の値を、CPU\_RAM203上の変数sに格納する(ステップS1702)。

## 【0151】

次にCPU201は、下記(16)式に示されるように、ステップS1702で変数sにセットされた今回の通算シーケンスロック割込み回数の値から、今回以前で最後に図11のステップS1109の自動演奏定常処理が実行されたときのシーケンスカウンタ値を格納する変数LAST\_SEQ\_CLOCKの値を減算した値を、CPU\_RAM203上の変数dに格納する(ステップS1703)。この結果、変数dには、今回以前で最後に図11のステップS1109の自動演奏定常処理が実行されたときから今回までの間に発生したシーケンスロック割込みの回数が格納される。40

## 【0152】

$$d = s - LAST\_SEQ\_CLOCK \quad \dots \quad (16)$$

## 【0153】

次にCPU201は、下記(17)式に示されるように、次回実行される図11のステップS1109の自動演奏定常処理のために、ステップS1702で変数sにセットされ50

た今回のシーケンスロック割込み回数の値を、変数 LAST\_SEQ\_CLOCK に格納しておく。

**【0154】**

LAST\_SEQ\_CLOCK = s . . (17)

**【0155】**

その後、CPU201は、下記ステップS1705からS1707の一連の処理の繰返しにより、変数dにセットされた前回から今回の自動演奏定常処理までの間に発生したシーケンスロック割込みの回数分だけ、自動演奏を進行させる処理を実行する。

**【0156】**

この繰返し処理において、CPU201はまず、変数dの値が0か否か、すなわち、変数dにセットされた前回から今回の自動演奏定常処理までの間にシーケンスロック割込みが発生していないか否かを判定する（ステップS1705）。 10

**【0157】**

変数dにセットされた前回から今回の自動演奏定常処理までの間にシーケンスロック割込みが発生しておらず変数dの値が0（ステップS1705の判定がYes）である場合には、CPU201は、図17のフローチャートで示される図11のステップS1109の自動演奏定常処理をそのまま終了する。

**【0158】**

変数dにセットされた前回から今回の自動演奏定常処理までの間にシーケンスロック割込みが発生しており変数dの値が0でない（ステップS1705の判定がNo）である場合には、CPU201は、自動演奏処理を1シーケンスロック割込みに対応する時間分だけ進める自動演奏制御処理を実行する（ステップS1706）。自動演奏制御処理は一般的な技術であるため、その詳細は省略する。 20

**【0159】**

次にCPU201は、変数dにセットされたシーケンスロック割込みの回数を-1する。その後、CPU201は、ステップS1705の判定処理に戻って変数dの値が0になったか否かを判定し、その判定がNoである間、上記ステップS1706とS1707の処理を繰り返し実行する。やがて、ステップS1705の判定がYesになると、CPU201は、図17のフローチャートで示される図11のステップS1109の自動演奏定常処理を終了する。 30

**【0160】**

以上の自動演奏定常処理により、CPU201は、前回から今回の自動演奏定常処理までの間に発生したシーケンスロック割込みの回数分だけ、自動演奏を進行させることができる。

**【0161】**

図18は、CPU201内のシーケンスロックカウント301のカウント値がシーケンスロックカウント上限値に達してシーケンスロック割込みが発生したときにCPU201が実行する、シーケンスロック割込み処理の例を示すフローチャートである。この処理は、CPU201が、図11～図17のフローチャートで例示される電子楽器制御処理を中断し、CPU ROM202に記憶されたシーケンスロック割込み処理プログラムを実行する動作である。 40

**【0162】**

CPU201はまず、自動演奏開始後のシーケンスロック割込みの通算の割込み回数を示す通算シーケンスロック割込みカウント変数SEQ\_CLOCK（図9（b）参照）の値を、+1インクリメントする（ステップS1801）。

**【0163】**

次にCPU201は、ディレイタイム毎にシーケンスロック割込みの回数をカウントするためのディレイタイム毎シーケンスロック割込みカウント変数SYNC\_SEQ\_CLOCK（図9（b）参照）の値を、+1インクリメントする（ステップS1802）。なお、このカウント変数の値は、ディレイタイムと同期したことが検出される後述する 50

ステップS1805の判定がY e sとなるタイミングで、0にリセットされる（後述するステップS1806）。

#### 【0164】

次にCPU201は、変数DELAY\_SYNCにセットされている値を判定する（ステップS1803）。

#### 【0165】

CPU201は、ステップS1612で、変数DELAY\_SYNCにセットされた値が1ではない、すなわち、ディレイテンポシンクロナイズドモードがオフされると判定した場合には、そのまま図18のフローチャートで示されるシーケンスロック割込み処理を終了し、図11～図17のフローチャートで例示される電子楽器制御処理を再開する。この場合、シーケンスロック割込みに基づいてステップS1801で通算シーケンスロック割込みカウント変数SEQ\_CLOCKの値がインクリメントされるため、前述した自動演奏定常処理（図11のステップS1109）により自動演奏が進行させられる。10

#### 【0166】

一方、CPU201は、ステップS1612で、変数DELAY\_SYNCにセットされた値が1である、すなわち、ディレイテンポシンクロナイズドモードがオンされると判定した場合には、以下の動作を実行する（ステップS1804）。CPU201は、ユーザの操作によって図1の機能選択操作子102のDELAYエリアのBEATノブによって選択され、CPU RAM203上の変数DELAY\_SYNC\_BEATに格納されている設定値によって、CPU ROM202に記憶されているSYNC\_BEAT\_TBL（図9（a））を参照することにより、DELAY\_SYNC\_BEAT値に対応するエントリのNUMERATOR項目値とDENOMINATOR項目値をそれぞれ、SYNC\_BEAT\_TBL(DELAY\_SYNC\_BEAT).NUMERATOR及びSYNC\_BEAT\_TBL(DELAY\_SYNC\_BEAT).DENOMINATORとして取得する。そして、CPU201は、下記（18）式により、ユーザが図1の機能選択操作子102のDELAY領域にあるエリアのBEATノブで設定した設定値に対応する同期拍数に応じたシーケンスロック割込み回数の設定値を算出し、CPU RAM203上の変数sに格納する。20

#### 【0167】

$$s = 480 \times SYNC_BEAT_TBL(DELAY_SYNC_BEAT).NUMERATOR / SYNC_BEAT_TBL(DELAY_SYNC_BEAT).DENOMINATOR \quad \dots (18)$$

#### 【0168】

次にCPU201は、シーケンスロック割込みの発生毎にステップS1802でインクリメントしたディレイタイム毎シーケンスロック割込みカウント変数SEQ\_CLOCKの値が、上記ステップS1804で算出し変数sに格納された同期拍数に応じたシーケンスロック割込み回数の設定値に一致したか否かを判定する（ステップS1805）。

#### 【0169】

ステップS1805の判定がNoならば、シーケンスロック割込みの回数がまだディレイタイムの設定値を経過していないため、CPU201は、そのまま図18のフローチャートで示されるシーケンスロック割込み処理を終了し、図11～図17のフローチャートで例示される電子楽器制御処理を再開する。40

#### 【0170】

ステップS1805の判定がYesになると、シーケンスロック割込みの回数がディレイタイムの設定値を経過したことになるため、CPU201はまず、ディレイタイム毎シーケンスロック割込みカウント変数SEQ\_CLOCKの値を次のディレイタイムの処理のために0にクリアする（ステップS1806）。

#### 【0171】

50

次にCPU201は、前述したCPU201内のフリーランニングタイマカウンタ302（図3参照）の値を示すCPU201無いレジスタCPU\_FREE\_TIMER（図10（a）参照）の値を、CPU RAM203上の変数LAST\_BEAT\_TIMERに格納する（ステップS1807）。この変数により、CPU201が最後にシーケンスクロック割込みがディレイタイムと一致する回数の同期拍をカウントしたと判断した時の時刻値が記憶される。

#### 【0172】

次にCPU201は、CPU RAM203上の変数SYNC\_STATの値が1であるか否かを判定する（ステップS1808）。ここで、変数SYNC\_STATは、今回のシーケンスクロック割込みに対応するDSP206からのディレイタイム割込みが先に発生し終わっているか否かを示している。もし、今回のシーケンスクロック割込みよりも先にそれに対応するディレイタイム割込みが発生していれば、後述するディレイタイム割込み処理では、その発生時刻が変数LAST\_DELAY\_TIMEに記録される処理のみが実行された後に（後述する図19のステップS1902参照）、シーケンスクロック補正処理は実行されずに変数SYNC\_STATの値が1にセットされてから、そのディレイタイム割込み処理がそのまま終了して（後述する図19のステップS1903 S1906参照）、シーケンスクロック補正処理はそれに続いて発生するシーケンスクロック割込み処理にゆだねられる。一方、もし、今回のシーケンスクロック割込みのほうがそれに対応するディレイタイム割込みよりも先に発生すれば、変数SYNC\_STATの値は前回のシーケンスクロック補正処理の後に0にリセットされたままとなっている（後述する図18のステップS1810又は図19のステップS1905参照）。

#### 【0173】

従って、変数SYNC\_STATの値が1（ステップS1808の判定がYes）の場合には、ディレイタイム割込みのほうが先に発生しておりその発生時刻が変数LAST\_DELAY\_TIMEに揃っている状態であるため、CPU201は、シーケンスクロック補正処理を実行する（ステップS1809）。この処理の詳細は、図20のフローチャートを用いて後述する。

#### 【0174】

シーケンスクロック補正処理が終了すると、CPU201は、変数SYNC\_STATの値を0にリセットする（ステップS1810）。その後、CPU201は、図18のフローチャートで示される今回のシーケンスクロック割込み処理を終了し、図11～図17のフローチャートで例示される電子楽器制御処理を再開する。

#### 【0175】

一方、変数SYNC\_STATの値が0（ステップS1808の判定がNo）の場合には、今回のシーケンスクロック割込み処理に対応するディレイタイム割込みがまだ発生していない。この場合には、CPU201は、変数SYNC\_STATに値1を格納し（ステップS1811）、その後そのまま、図18のフローチャートで示されるシーケンスクロック割込み処理を終了し、図11～図17のフローチャートで例示される電子楽器制御処理を再開する。この場合には、続けて今回のシーケンスクロック割込みに対応するディレイタイム割込みが発生したタイミングで、変数SYNC\_STAT = 1が判定されて、シーケンスクロック補正処理が実行されることになる（後述する図19のステップS1901 S1902 S1903の判定がYes S1905）。なお、この場合に、今回のシーケンスクロック割込みに対応するディレイタイム割込みの発生の前に次回のシーケンスクロック割込みが再び発生して図18のフローチャートが実行されたとしても、変数SYNC\_SEQ\_CLOCKの値はステップS1806で0にリセットされている。そのため、ステップS1801 S1802 S1803の判定がYes S1804と実行された後に、ステップS1805の判定がNoとなる。これにより、今回のシーケンスクロック割込みに対応するディレイタイム割込みが発生する前に、次回のシーケンスクロック割込みに基づいてステップS1809のシーケンスクロック補正処理が実行されてしまうことはない。

10

20

30

40

50

**【0176】**

図19は、DSP206内のディレイタイムカウンタ601のカウント値がディレイタイムサンプリング数608（図6参照）に達してディレイタイム割込みが発生したときにCPU201が実行する、ディレイタイム割込み処理の例を示すフローチャートである。この処理は、CPU201が、図11～図17のフローチャートで例示される電子楽器制御処理を中断し、CPU ROM202に記憶されたディレイタイム割込み処理プログラムを実行する動作である。

**【0177】**

CPU201はまず、変数DELAY\_SYNCにセットされている値を判定する（ステップS1901）。 10

**【0178】**

CPU201は、ステップS1901で、変数DELAY\_SYNCにセットされた値が1ではない、すなわち、ディレイテンポシンクロナイズドモードがオフされていると判定した場合には、そのまま図19のフローチャートで示されるシーケンスロック割込み処理を終了し、図11～図17のフローチャートで例示される電子楽器制御処理を再開する。

**【0179】**

一方、CPU201は、ステップS1901で、変数DELAY\_SYNCにセットされた値が1である、すなわち、ディレイテンポシンクロナイズドモードがオンされていると判定した場合には、前述したCPU201内のフリーランニングタイマカウンタ302（図3参照）の値を示すCPU201のレジスタCPU\_FREE\_TIMER（図10（a）参照）の値を、CPU RAM203上の変数LAST\_DELAY\_TIMEに格納する（ステップS1902）。この変数により、DSP206が最後にディレイタイム割込みを受けた時の時刻値が記憶される。 20

**【0180】**

次にCPU201は、CPU RAM203上の変数SYNC\_STATの値が1であるか否かを判定する（ステップS1903）。ディレイタイム割込み処理が実行される場合における変数SYNC\_STATの意味は、今回のディレイタイム割込みに対応する同期拍数に応じた回数分のシーケンスロック割込みが先に発生し終わっているか否かを示している。もし、今回のディレイタイム割込みよりも先にそれに対応する上記同期拍数に応じた回数分のシーケンスロック割込みが発生していれば、前述したシーケンスロック割込み処理では、その発生時刻が変数LAST\_BEAT\_TIMEに記録される処理のみが実行された後に（前述した図18のステップS1807参照）、シーケンスロック補正処理は実行されずに変数SYNC\_STATの値が1にセットされてから、そのシーケンスロック割込み処理がそのまま終了して（前述した図18のステップS1808 S1811参照）、シーケンスロック補正処理はそれに続いて発生するディレイタイム割込み処理にゆだねられる。一方、もし、今回のディレイタイム割込みのほうがそれに対応する上記同期拍数に応じた回数分のシーケンスロック割込みよりも先に発生すれば、変数SYNC\_STATの値は前回のシーケンスロック補正処理の後に0にリセットされたままとなっている（前述した図18のステップS1810又は後述する図19のステップS1905参照）。 30

**【0181】**

従って、変数SYNC\_STATの値が1（ステップS1903の判定がYes）の場合には、上記同期拍数に応じた回数分のシーケンスロック割込みのほうが先に発生しておりその発生時刻が変数LAST\_BEAT\_TIMEに揃っている状態であるため、CPU201は、シーケンスロック補正処理を実行する（ステップS1904）。この処理の詳細は、図20のフローチャートを用いて後述する。

**【0182】**

シーケンスロック補正処理が終了すると、CPU201は、変数SYNC\_STATの値を0にリセットする（ステップS1905）。その後、CPU201は、図19のフ 50

フローチャートで示される今回のディレイタイム割込み処理を終了し、図11～図17のフローチャートで例示される電子楽器制御処理を再開する。

#### 【0183】

一方、変数SYNC\_STATの値が0（ステップS1903の判定がNo）の場合には、今回のディレイタイム割込み処理に対応する上記同期拍数に応じた回数分のシーケンスロック割込みがまだ発生していない。この場合には、CPU201は、変数SYNC\_STATに値1を格納し（ステップS1906）、その後そのまま、図19のフローチャートで示されるディレイタイム割込み処理を終了し、図11～図17のフローチャートで例示される電子楽器制御処理を再開する。この場合には、続けて今回のディレイタイム割込みに対応する上記同期拍数に応じた回数分のシーケンスロック割込みが発生したタイミングで、変数SYNC\_STAT=1が判定されて、シーケンスロック補正処理が実行されることになる（前述した図18のステップS1808の判定がYes S1809）。なお、この場合に、今回のディレイタイム割込みの発生の前にシーケンスロック割込みが再び発生して図18のフローチャートが実行されたとしても、変数SEQ\_CLOCKの値はステップS1806で0にリセットされている。このため、ステップS1801 S1802 S1803の判定がYes S1804と実行された後に、ステップS1805の判定がNoとなる。これにより、今回のディレイタイム割込みに対応する上記同期拍数に応じた回数分のシーケンスロック割込みが発生する前に、次のディレイタイム割込みが発生してシーケンスロック補正処理（ステップS1905）が実行されてしまうことはタイミング的にあり得ない。

10

20

#### 【0184】

図20は、図18のステップS1809又は図19のステップS1904のシーケンスロック補正処理の詳細例を示すフローチャートである。以下のシーケンスロック補正処理は、(7)式から(10)式を用いた説明で前述した原理に基づいて実行される。

#### 【0185】

CPU201はまず、前述した(9)式に基づいて、前述した図18のステップS1807で変数LAST\_BEAT\_TIMEにセットされている今回の同期拍数に応じた回数分のシーケンスロック割込みの発生時刻と、前述した図19のステップS1902で変数LAST\_DELAY\_TIMEにセットされている今回のディレイタイム割込みの発生時刻との時間差dを計測する（ステップS2001）。

30

#### 【0186】

次にCPU201は、ユーザの操作によって図1の機能選択操作子102のDELAYエリアのBEATノブによって選択され、CPU RAM203上の変数DELAY\_SYNC\_BEATに格納されている設定値によって、CPU ROM202に記憶されているSYNC\_BEAT\_TBL（図9(a)）を参照することにより、DELAY\_SYNC\_BEAT値に対応するエントリのNUMERATOR項目値SYNC\_BEAT\_TBL(DELAY\_SYNC\_BEAT).NUMERATORと、同エントリのDENOMINATOR項目値YNC\_BEAT\_TBL(DELAY\_SYNC\_BEAT).DENOMINATORとを用いて、前述した(7)式に対応する下記(19)式によって、ユーザが上記BEATノブで設定した設定値に対応する同期拍数bを算出する（ステップS2002）。

40

#### 【0187】

$b = SYNC_BEAT_TBL(DELAY_SYNC_BEAT).NUMERATOR / SYNC_BEAT_TBL(DELAY_SYNC_BEAT).DENOMINATOR \dots (19)$

#### 【0188】

次にCPU201は、ステップS2001で算出される時間差dと、ステップS2002で算出される同期拍数bとを用いて、前述した(10)式により、シーケンスロックカウント上限値の補正值cを計算する（ステップS2003）。

#### 【0189】

50

そしてCPU201は、下記(20)式で示されるように、CPU201のレジスタCPU\_TIMER\_COUNT(図10(a)参照)にセットされているシーケンスクロックカウンタ301(図3)に設定されるシーケンスクロックカウント上限値から、ステップS2003で算出した補正值cを減算して、再びレジスタCPU\_TIMER\_COUNTにセットし、シーケンスクロックカウント上限値を更新する(ステップS2004)。なお、レジスタCPU\_TIMER\_COUNTの値は、前述した図11のステップS1101の初期化処理におけるステップS1203又は図11のステップS1104のテンポ設定処理における図13のステップS1306で設定される。

#### 【0190】

```
CPU_TIMER_COUNT  
= CPU_TIMER_COUNT - c . . . (20)
```

10

#### 【0191】

この結果、次回のディレイ処理においては、CPU201においてシーケンスロック割込みに従ってディレイタイム分だけ進行する自動演奏の時間と、DSP206におけるディレイタイム分のディレイ処理の時間とのずれが無くなることが期待される。

#### 【0192】

以上のように、本実施形態においては、音響効果として、音響信号に対してエコー効果を付加するディレイ処理を実行する構成について説明したが、本発明において音響効果はこれに限られるものでない。例えば、音響信号に対してビブラート効果及びトレモロ効果の少なくとも一方を付加するためのLFOを生成する処理を実行するように構成してもよい。

20

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)

サンプリングクロックをカウントするカウントとともに、前記カウントされたカウント値が第1のカウント値となる毎に割込み信号を出力する第1のカウント処理と、前記カウントされたカウント値に基づいて、供給される音響波形信号に対して音響効果を付加する音響効果処理と、を実行する第1の処理部と、

前記サンプリングクロックとは非同期でかつ異なる周期のシーケンスロックをカウントする第2のカウント処理と、前記第2カウント処理によりカウントを開始してからカウント値が第2のカウント値になるまでに要する時間と、前記第1の処理部からの割り込み信号の出力間隔時間との差分時間を検出する検出処理と、前記検出された差分時間を小さくするように前記第2のカウント値を変更する補正処理と、前記第2のカウント処理によりカウントされるカウント値が前記第2の値になる毎に自動演奏を進行させる自動演奏進行処理を実行する第2の処理部と、

30

を備えた音響処理装置。

#### (付記2)

前記音響処理装置はさらに、前記サンプリングクロックを発生する第1のクロック発生部と、前記シーケンスロックを発生する第2のクロック発生部と、を有する、付記1に記載の音響処理装置。

#### (付記3)

40

前記第1の処理部は、前記サンプリングクロックをカウントすることにより、前記第1のカウント処理を実行する第1のカウンタを有し、

前記第2の処理部は、前記シーケンスロックをカウントすることにより、前記第2のカウント処理を実行する第1のカウンタを有する、付記1又は2記載の音響処理装置。

#### (付記4)

前記音響処理装置さらに、前記自動演奏のテンポを指定するテンポ指定部を有し、

前記第1のカウント値及び前記第2のカウント値は、前記指定されたテンポに対応して決定される、付記1乃至3のいずれかに記載の音響処理装置。

#### (付記5)

前記音響処理装置はさらに、複数種の自動演奏のテンポ夫々に対応して前記第1及び第

50

2 のカウント値を記憶するテーブルを有し、

前記第 2 の処理部はさらに、指定された自動演奏のテンポに対応する前記第 1 及び第 2 のカウント値を前記テーブルから読み出すとともに、前記第 1 のカウント値を前記第 1 の処理部に設定するカウント値設定処理を実行する、付記 1 乃至 4 のいずれかに記載の音響処理装置。

(付記 6 )

前記第 1 の処理部は、前記音響効果処理として、前記カウントされたカウント値が前記第 1 のカウント値に達するまでの時間の自然数倍及び所定自然数分の 1 倍のいずれか一方の関係の時間に同期したタイミングで、供給される音響信号に対して音響効果を付加する処理を実行する、付記 1 乃至 5 のいずれかに記載の音響処理装置。 10

(付記 7 )

前記第 1 の処理部は、前記音響効果処理として、前記音響信号に対してエコー効果を付加するディレイ処理を実行する、付記 1 乃至 6 のいずれかに記載の音響処理装置。

(付記 8 )

前記第 1 の処理部は、前記音響効果処理として、前記音響信号に対してビブラート効果及びトレモロ効果の少なくとも一方を付加するための LFO を生成する処理を実行する、付記 1 乃至 6 のいずれかに記載の音響処理装置。

(付記 9 )

第 1 の処理部及び第 2 の処理部を有する音響処理装置に用いられる音響処理方法であつて、 20

前記第 1 の処理部が、

サンプリングクロックをカウントするカウントとともに、前記カウントされたカウント値が第 1 のカウント値となる毎に割込み信号を出力し、

前記カウントされたカウント値に基づいて、供給される音響波形信号に対して音響効果を付加し、

前記第 2 の処理部が、

前記サンプリングクロックとは非同期でかつ異なる周期のシーケンスクロックをカウントし、

前記カウントを開始してからカウント値が第 2 のカウント値になるまでに要する時間と、前記第 1 の処理部からの割り込み信号の出力間隔時間との差分時間を検出し、 30

前記検出された差分時間を小さくするよう前記第 2 のカウント値を変更し、

前記カウントされたカウント値が前記第 2 のカウント値になる毎に自動演奏を行わせる、

音響処理方法。

(付記 10 )

音響処理装置として用いられるコンピュータに、

サンプリングクロックをカウントするカウントとともに、前記カウントされたカウント値が第 1 のカウント値となる毎に割込み信号を出力するステップと、

前記カウントされたカウント値に基づいて、供給される音響波形信号に対して音響効果を付加するステップと、 40

前記サンプリングクロックとは非同期でかつ異なる周期のシーケンスクロックをカウントするステップと、

前記シーケンスクロックのカウントを開始してからカウント値が第 2 のカウント値になるまでに要する時間と、前記割り込み信号の出力間隔時間との差分時間を検出するステップと、

前記検出された差分時間を小さくするよう前記第 2 のカウント値を変更するステップと、

前記シーケンスクロックをカウントすることに得られるカウント値が前記第 2 のカウント値になる毎に自動演奏を行わせるステップと、

を実行させるプログラム。 50

## (付記 11)

請求項 1 乃至 8 のいずれかに記載の音響処理装置と、  
発生すべき楽音の音高を指定する演奏操作子と、  
前記演奏操作子で指定された音高の楽音波形を音響波形信号として生成して、前記第 1  
の処理部に供給する波形発生器と、  
を備えた電子楽器。

## 【符号の説明】

## 【0193】

100 電子鍵盤楽器

101 鍵盤

201 CPU

202 CPU ROM

204 音源 LSI

205 波形発生器

206 DSP

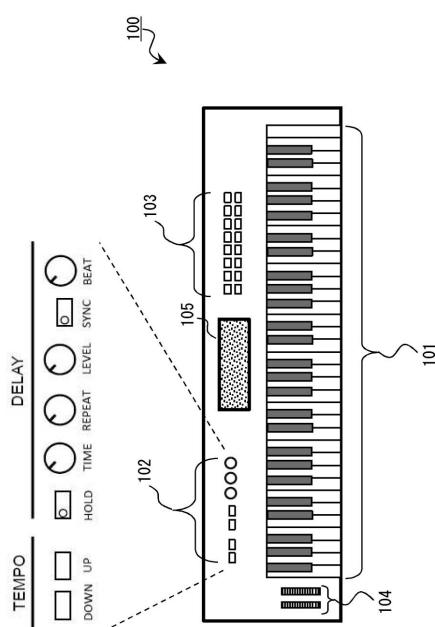
209、210 発振子

301 シーケンスロックカウンタ

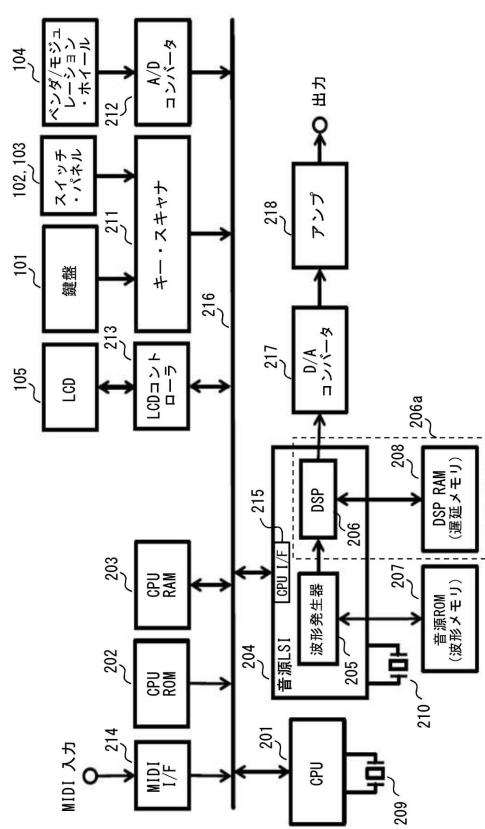
302 フリーランニングタイマカウンタ

10

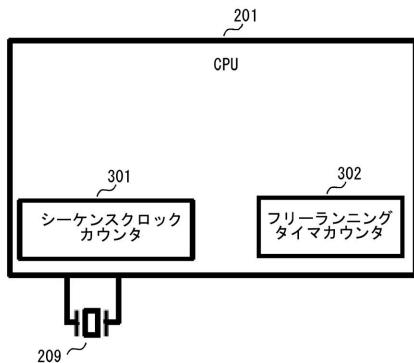
【図 1】



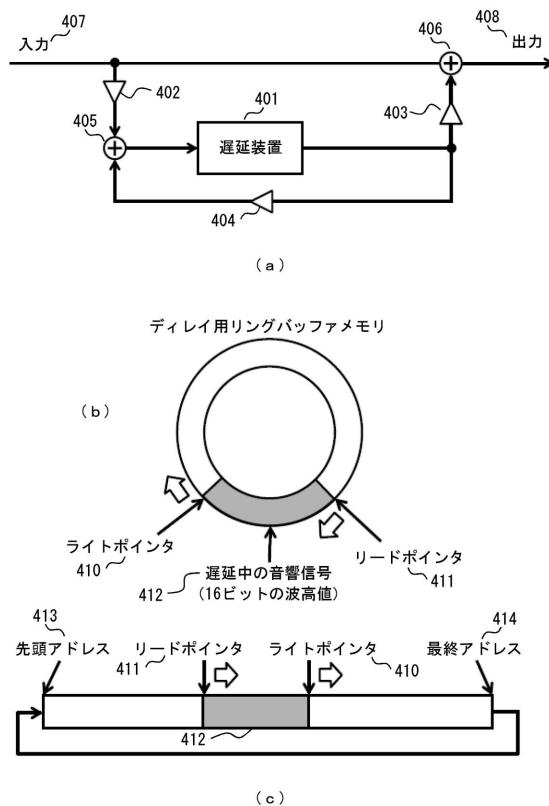
【図 2】



【図3】



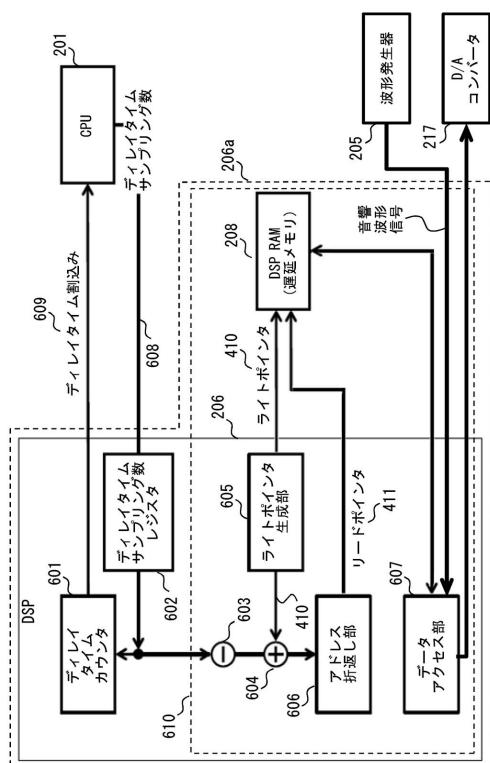
【図4】



( 5 )

TEMPO_COUNT_TBL			補足情報		
TEMPO	DELAY_COUNT	SEQ_CLOCK_COUNT	補足情報1	補足情報2	補足情報3
テンボ (BPM)	ディレイタイム を指すインデックス の1)拍に同期 させたときの、 DSPサンプリ ングクロック 数(自然数)	指すテンボの 拍×480分割 するシーケンス ワップ割込み の発生に依る、 CPU×ステ ムクロック数(μ sec)	DELAY COUNT T分をサンプリ ングするのに 要する所要時 間(msec)	DELAY COUNT 所要時間と、 シーケンス ワップ割込み の拍分をカウト したときの所要 時間の時間差 (msec)	4/4拍子(32 小節後に生ず る時間差 (msec)
30	88200	4167	2000	-0.16	-20.48
31	85355	4032	1935.49	0.13	16.64
32	82688	3906	1875.01	0.13	16.64
33	80182	3788	1818.19	-0.05	-6.4
34	77824	3676	1764.72	0.24	30.72
35	75600	3571	1714.29	0.21	26.88
:	:	:			
115	23009	1087	521.75	-0.01	-1.28
116	22810	1078	517.23	-0.21	26.88
117	22615	1068	512.81	0.17	21.76
118	22424	1059	508.48	0.16	20.48
119	22235	1050	504.2	0.2	25.6
120	22050	1042	500	-0.16	-20.48
121	21868	1033	495.87	0.03	3.84
122	21689	1025	491.81	-0.19	-24.32
123	21512	1016	487.8	0.12	15.36
124	21339	1008	483.88	0.04	5.12
125	21168	1000	480	0	0
:	:	:			
295	8969	424	203.38	-0.14	-17.92
296	8939	422	202.7	0.14	17.92
297	8909	421	202.02	-0.06	-7.68
298	8879	419	201.34	0.22	28.16
299	8849	418	200.66	0.02	2.56
300	8820	417	200	-0.16	-20.48

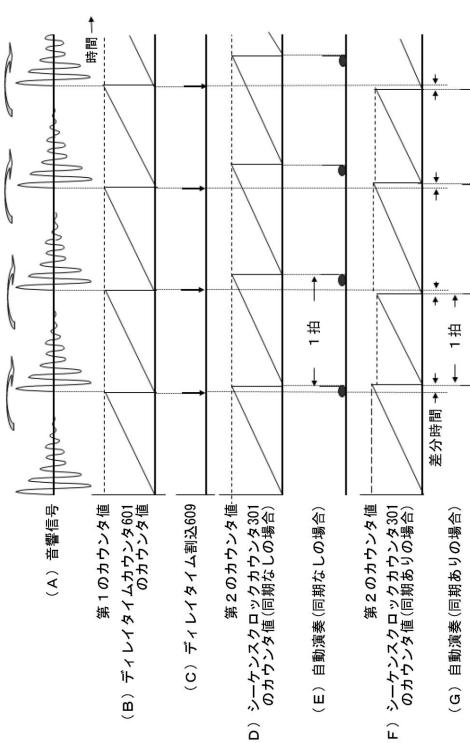
( 6 )



【図7】

SYNC.BEAT.TBL			補助情報	
設定値	NUMERATOR (分子)	DENOMINATOR (分母)	同期拍数	同期シケンス クロック数
0	1	4	1/4	120
1	1	3	1/3	160
2	1	2	1/2	240
3	2	3	2/3	320
4	1	1	1	480
5	3	2	3/2	720
6	2		2	960
7	3	1	3	1440

【 义 8 】



【図9】

変数 [配列]名	配列メンバー	サイズ	値域	説明
TEMPO_COUNT_TBL[301] 配列: 予ん拂-0~300 0~29で使用	DELAY_COUNT	32bit	0~100000	ディレイタイムを指定テンポの 1拍に同様させたときの、 DSPサンプリングクロック数
	SEQ_CLOCK_COUNT	32bit	0~4167	指定テンポの1拍を480分割 する1シーケンスクロックの、 CPUシステムクロック数
SYNC_BEAT_TBL[8] 配列: 設定値-0~7	NUMERATOR	32bit	1,1,1,2,1,3,2,3	同期拍数の分子
	DENOMINATOR	32bit	4,3,2,3,1,2,1,1	同期拍数の分母

( a )

実数 [配列数]	サイズ	値域 (* <sup>=====</sup> R16連数)	初期 値	説明
TEMPO	16bit	30-300	120	設定テンポ(BPM)
SEQ_RUN	1bit	0..1	0	自動演奏を行っている状態かどうか。 0...No 1...Yes
DELAY_HOLD	1bit	0..1	0	ディレイホールドモードかどうか。 0...No 1...Yes
DELAY_SYNC	1bit	0..1	0	ディレイテンボシンクロナズモードかどうか。 0...No 1...Yes
DELAY_TIME	16bit	0-FFFFH	-	ディレイ・タイム(ms単位) 初期化時にモードに応じた値に設定
DELAY_FEEDBACK	16bit	0-FFFFH	0	ディレイのフィードバック量。FFFFは0.1(100%)を表す。
DELAY_LEVEL	16bit	0-FFFFH	0	ディレイ音のレベル。FFFFは1.0(100%)を表す。
DELAY_SYNC_BEAT	3bit	0-7	-	同期駆動[ターゲット] (同期拍数ビンディングスクラップ数対応参考)
SEQ_CLOCK	32bit	0-xFFFFFFFF	0	自動演奏のテンポを司るシーケンスクロックカウントの値
LAST_SEQ_CLOCK	32bit	0-xFFFFFFFF	0	CPUが最後に自動演奏処理を行った際の、SEQ_CLOCKの値
LAST_DELAY_TIME	32bit	0-xFFFFFFFF	0	DSPが最後にディレイタイム割り込みを受けた時刻の フリーランタイミング時間
LAST_BEAT_TIME	32bit	0-xFFFFFFFF	0	CPUが最初にディレイタイムと一致する想定の拍を経过了したと判断した 時のフリーランタイミング時間
SYNC_SEQ_CLOCK	16bit	0-FFFFH	0	ディレイタイム割り込みとシーケンスクロック共にインクリメントし、 同期タイミングで0になります。
SYNC_STAT	1bit	0-1	0	同期タイミング近辺においてディレイタイム割り込み。 またはCPUが最初の経过了が発生した時刻に1に設定され、 両方のイベント発生が終った時点に0にリセットされる。

( b )

【図 10】

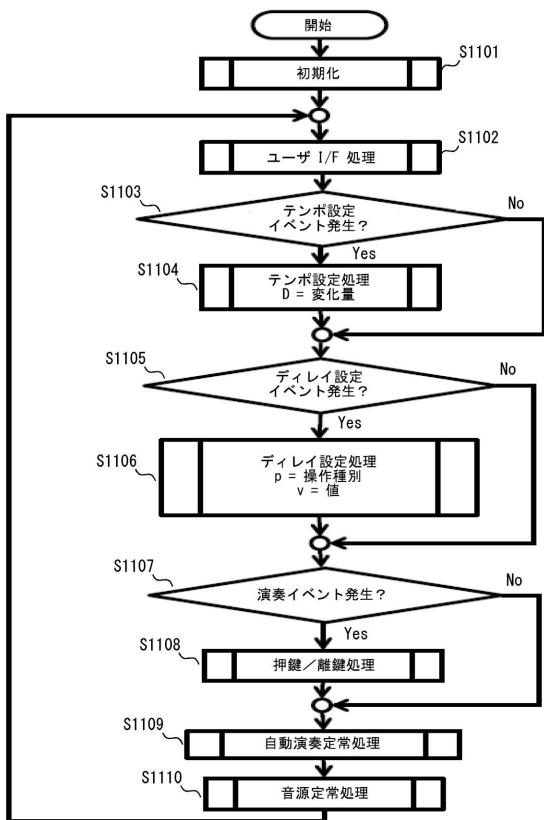
変数 [配列数]	サイズ	値域 (****Hは16進数)	初期値	説明
CPU_FREE_TIMER	32bit	0-FFFFFFFFFFH	0	CPUのフリーランタイマの値
CPU_TIMER_COUNT	32bit	0-FFFFFFFFFFH	0	CPUのタイマの設定値 ここで設定した値だけカウントする割込みを 掛けていく間に、0に到達時は動作しない。

( a )

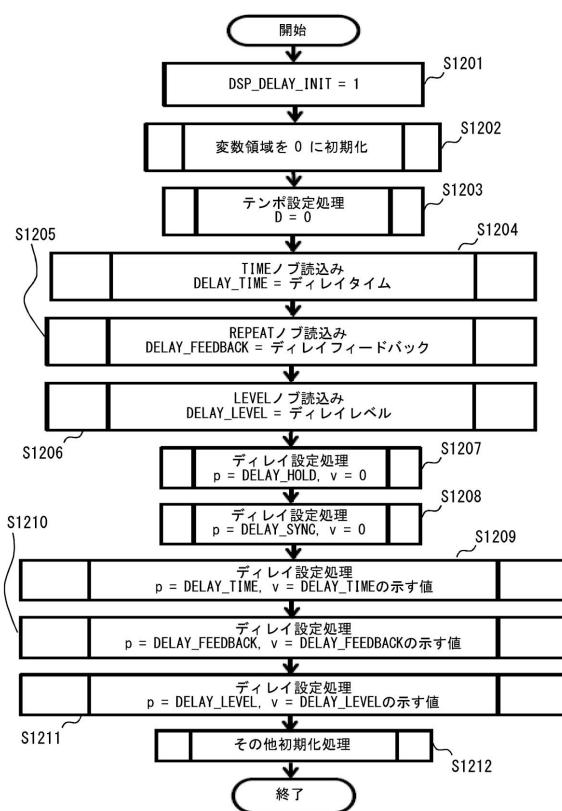
変数 [配列数]	サイズ	値域 (*-*-*-*H*16 連)	初期値	説明
DSP_DELAY_SAMPLE	16bit	0~600000	0	ディレイサンプリング取扱リスト (書き込みとディレイサンプリングカウントと、 リードポイントに値が設定される)
DSP_DELAY_INPUT	16bit	0~FFFFH	FFFFH	ディレイ入力音量調整アドレス (0#0~0, FFFFH#1.0倍係数)
DSP_DELAY_OUTPUT	16bit	0~FFFFH	FFFFH	ディレイ出力音量調整アドレス (0#0~0, FFFFH#1.0倍係数)
DSP_DELAY_FEEDBACK	16bit	0~FFFFH	FFFFH	ディレイフィードバック音量調整アドレス (0#0~0, FFFFH#1.0倍係数)
DSP_DELAY_INIT	1bit	0,1	0	1を書き込むことでディレイモリやホンタを初期化する

( b )

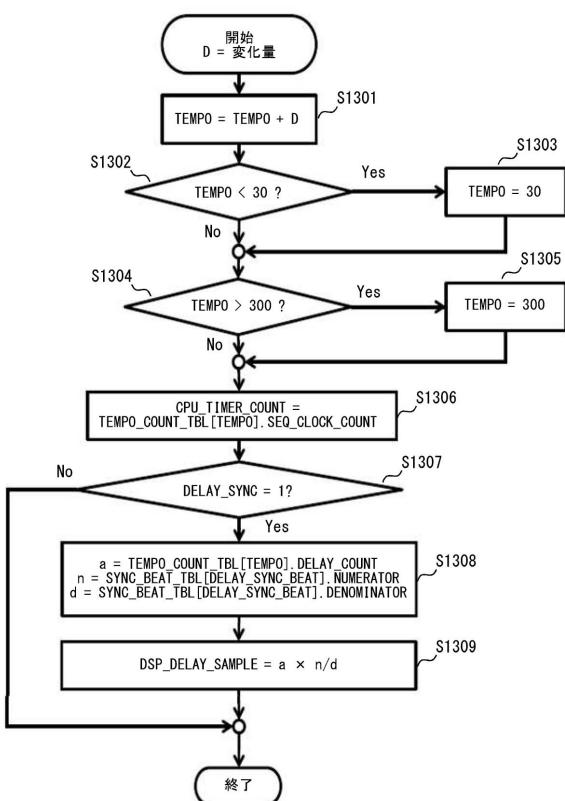
【図11】



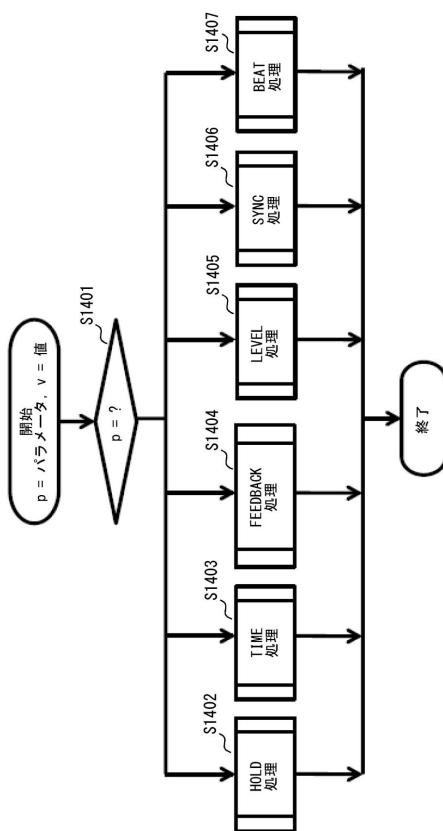
【図12】



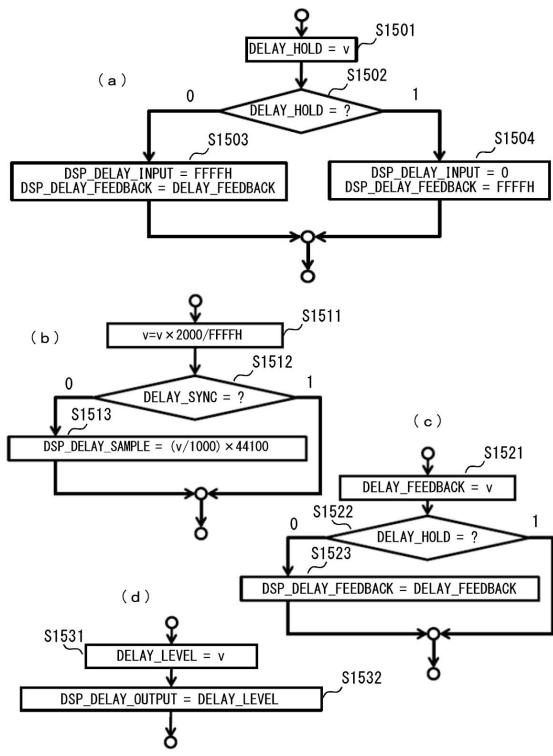
【図13】



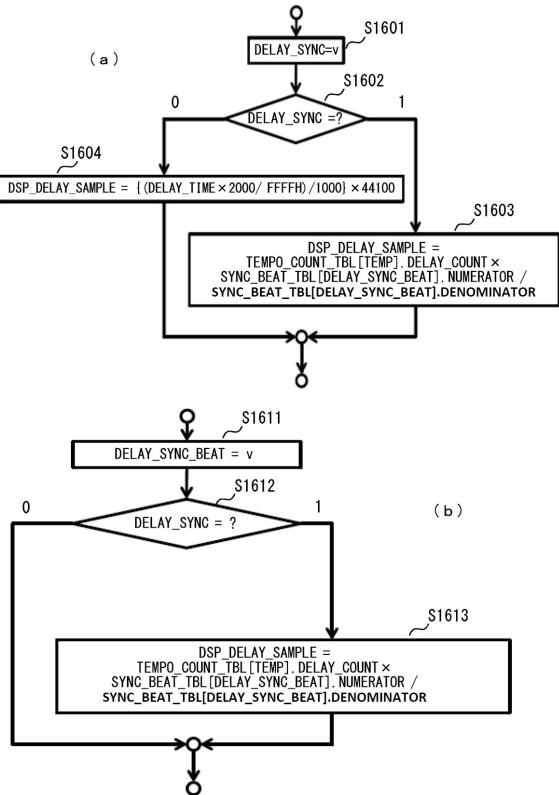
【図14】



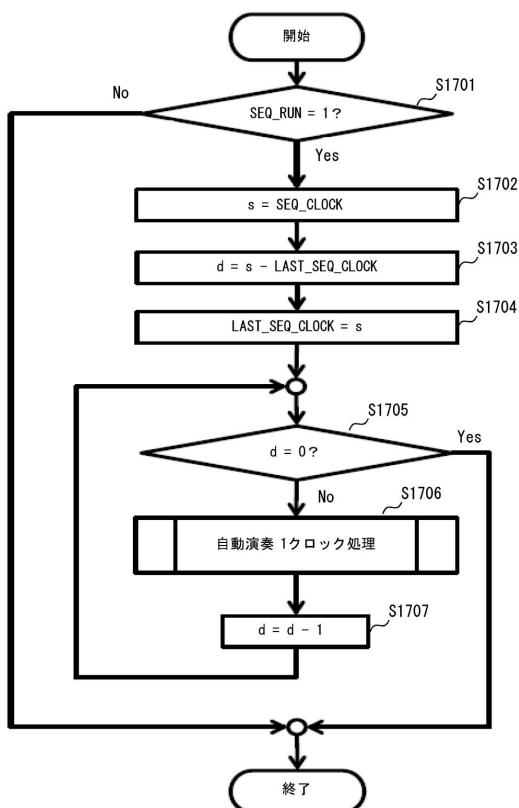
【図15】



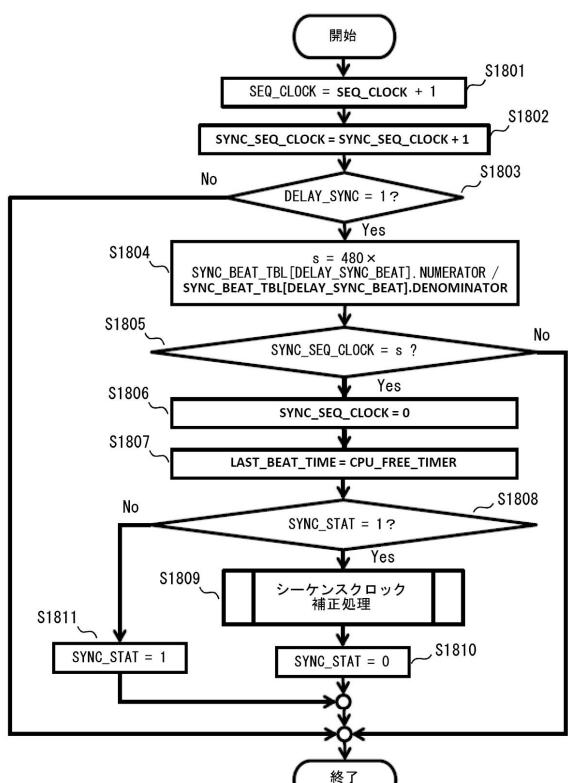
【図16】



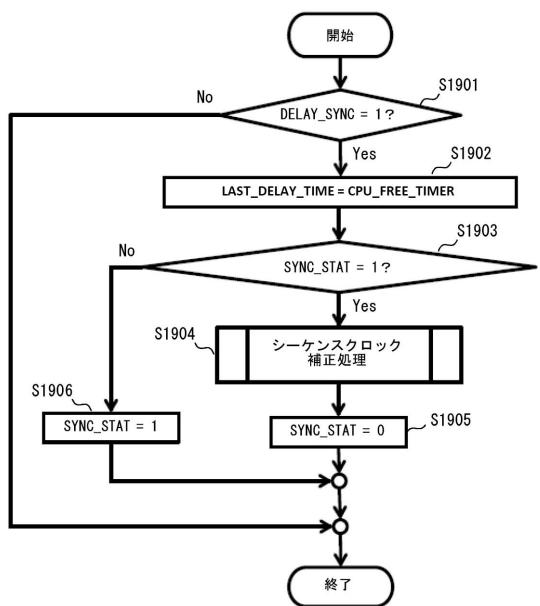
【図17】



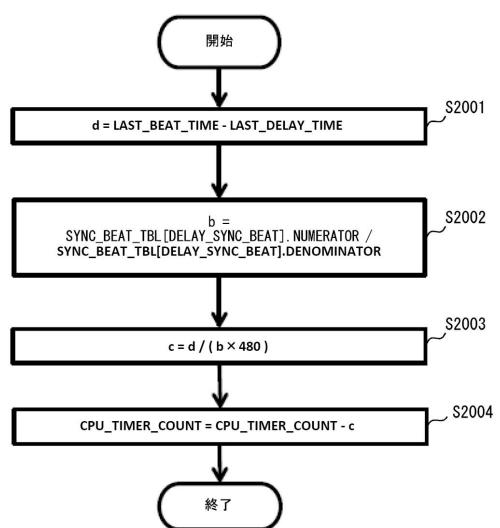
【図18】



【図19】



【図20】



---

フロントページの続き

(56)参考文献 特開2003-280649(JP,A)  
特開2010-113176(JP,A)  
特開平5-094180(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 10 H 1 / 0 0 - 7 / 1 2