(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3625914号
(P3625914)
(45)発行日　平成17年3月2日(2005.3.2)　(24)登録日　平成16年12月10日(2004.12.10)

(51)Int.Cl.[7]　F I
G10H　1/28　G10H　1/28

請求項の数 4　(全 35 頁)

(21)出願番号　特願平7-241056
(22)出願日　平成7年9月20日(1995.9.20)
(65)公開番号　特開平9-90953
(43)公開日　平成9年4月4日(1997.4.4)
審査請求日　平成13年5月23日(2001.5.23)

(73)特許権者　000116068
ローランド株式会社
大阪府大阪市北区曽根崎新地一丁目４番２０号
(74)代理人　100094330
弁理士　山田　正紀
(74)代理人　100079175
弁理士　小杉　佳男
(74)代理人　100098349
弁理士　一徳　和彦
(72)発明者　岸本　哲史
大阪市北区堂島浜１丁目４番１６号　ローランド株式会社内

審査官　石丸　昌平

最終頁に続く

(54)【発明の名称】アルペジエータ

(57)【特許請求の範囲】
【請求項１】
複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段と、
リズムの１ステップ毎に、あるステップと当該ステップに続く次のステップとの間の時間間隔と、当該ステップにおける楽音の発音持続時間情報とを組にして、１つのステップもしくは複数のステップそれぞれに記憶されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、
前記リズムパターンテーブルの各ステップを順次参照するとともに、該各ステップに対応して前記記憶領域を走査し、該走査により検出された押鍵情報、および前記リズムパターンテーブルの該走査に対応するステップに記録された発音持続時間情報との双方に基づく楽音を表わす演奏情報を、当該ステップに記録された時間間隔に従ったタイミングで生成する演奏情報生成手段とを備え、
前記演奏情報生成手段が、さらに発音持続時間の変更の度合いを規定する情報に基づき変更された楽音を表わす演奏情報を生成するものであることを特徴とするアルペジエータ。
【請求項２】
複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段と、
リズムの１ステップ毎に、あるステップと当該ステップに続く次のステップとの間の時間間隔と、当該ステップにおける楽音の発音強度情報とを組にして、１つのステップもしく

は複数のステップそれぞれに記憶されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、
前記リズムパターンテーブルの各ステップを順次参照するとともに、該各ステップに対応して前記記憶領域を走査し、該走査により検出された押鍵情報、および前記リズムパターンテーブルの該走査に対応するステップに記録された発音強度情報との双方に基づく楽音を表わす演奏情報を、当該ステップに記録された時間間隔に従ったタイミングで生成する演奏情報生成手段とを備え、
前記演奏情報生成手段が、さらに発音強度の変更の度合いを規定する情報に基づき変更された楽音を表わす演奏情報を生成するものであることを特徴とするアルペジエータ。
【請求項３】
操作の度合いに応じて、前記発音持続時間の変更の度合いを規定する情報を生成する操作子を備えたことを特徴とする請求項１記載のアルペジエータ。
【請求項４】
操作の度合いに応じて、前記発音強度の変更の度合いを規定する情報を生成する操作子を備えたことを特徴とする請求項２記載のアルペジエータ。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、押鍵情報を走査しその走査結果に従って複数の演奏情報を順次生成するアルペジエータに関する。
【０００２】
【従来の技術】
従来の電子楽器には、いわゆるアルペジエータと呼ばれる機能が搭載されているものがある。このアルペジエータとは、時間の経過毎に、押鍵情報を、例えば音高の低い方から高い方へと順番に走査し、その結果に従って演奏情報を順次出力する機能である。
【０００３】
この機能によって、演奏者にとって本来演奏の困難なアルペジオ演奏を、単に複数の鍵を同時に押鍵するだけの簡単な演奏操作によって実現することができる。
【０００４】
【発明が解決しようとする課題】
しかしながら、従来のアルペジエータは、ある決まった一定の時間間隔で押鍵情報を走査してその一定の時間間隔で演奏情報を順次出力するものであり、このため発音タイミングは完全に均等になり、人間らしさがなく、音楽的に味気ないものとなってしまう傾向があった。
【０００５】
また、楽音の発音持続時間や楽音の音量等の楽音の特質も均等であり、この点からも人間らしさがなく、音楽的に味気ないアルペジオ演奏となってしまっていた。本発明は、従来のアルペジエータでは不可能であった、より音楽的なアルペジオ演奏を簡単な操作で行なうことのできるアルペジエータを提供することを目的とする。
【０００６】
【課題を解決するための手段】
上記目的を達成する本発明の第１のアルペジエータは、
（１－１）複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段と、
（１－２）リズムの１ステップ毎に、あるステップと当該ステップに続く次のステップとの間の時間間隔と、当該ステップにおける楽音の発音持続時間情報とを組にして、１つのステップもしくは複数のステップそれぞれに記憶されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、
（１－３）前記リズムパターンテーブルの各ステップを順次参照するとともに、該各ス

テップに対応して前記記憶領域を走査し、該走査により検出された押鍵情報、および前記リズムパターンテーブルの該走査に対応するステップに記録された発音持続時間情報との双方に基づく楽音を表わす演奏情報を、当該ステップに記録された時間間隔に従ったタイミングで生成する演奏情報生成手段とを備え、
前記演奏情報生成手段が、さらに発音持続時間の変更の度合いを規定する情報に基づき変更された楽音を表わす演奏情報を生成するものであることを特徴とする。
【０００７】
また、上記目的を達成する本発明の第２のアルペジエータは、
（２－１）複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段と、
（２－２）リズムの１ステップ毎に、あるステップと当該ステップに続く次のステップとの間の時間間隔と、当該ステップにおける楽音の発音強度情報とを組にして、１つのステップもしくは複数のステップそれぞれに記憶されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、
（２－３）前記リズムパターンテーブルの各ステップを順次参照するとととともに、該各ステップに対応して前記記憶領域を走査し、該走査により検出された押鍵情報、および前記リズムパターンテーブルの該走査に対応するステップに記録された発音強度情報との双方に基づく楽音を表わす演奏情報を、当該ステップに記録された時間間隔に従ったタイミングで生成する演奏情報生成手段とを備え、
前記演奏情報生成手段が、さらに発音強度の変更の度合いを規定する情報に基づき変更された楽音を表わす演奏情報を生成するものであることを特徴とする。
【０００８】
ここで、本発明の第１のアルペジエータにおいて、操作の度合いに応じて、前記発音持続時間の変更の度合いを規定する情報を生成する操作子を備えることが好ましい。
【０００９】
また、本発明の第２のアルペジエータにおいて、操作の度合いに応じて、前記発音強度の変更の度合いを規定する情報を生成する操作子を備えることが好ましい。
【００１０】
【発明の実施の形態】
以下、本発明の実施形態について説明する。
【００１１】
図１は、本発明のアルペジエータの一実施形態の回路構成図である。ここには、ＣＰＵ１０、アドレスバス１１およびデータバス１２でそのＣＰＵ１０と接続された、ＲＡＭ１３，ＲＯＭ１４，パネル１５、およびＣＰＵ１０にクロックの発生を通知するクロックタイマ１６、ＭＩＤＩ規格に準拠したデータを入力するデータ入力端子（ＭＩＤＩ　ＩＮ）１７、ＭＩＤＩ規格に準拠したデータを出力するデータ出力端子（ＭＩＤＩ　ＯＵＴ）１８が備えられている。
【００１２】
ＲＯＭ１４は、読出し専用のメモリエリアであり、ここには、ＣＰＵ１０で実行されるプログラムや、そのプログラムで参照される各種テーブルが格納されている。プログラムやテーブルの詳細については後述する。ＲＡＭ１３は、ＣＰＵ１０で実行されるプログラムのワーキングエリアとして使用される。ここでは、このＲＡＭ１３は電池でバックアップされており、電源を切ってもそのときのメモリ内容は保存される。メモリの内容について後述する。
【００１３】
パネル１５は、操作スイッチと表示器を持ち、パラメータを独立して選択しそのパラメータに値を設定することができる。設定された値はＲＡＭ１３内のメモリに記憶される。図２は、パネル１５を示した図である。このパネルでは、リズムパターン番号（Ｒｈｙｔｈｅｍ　Ｐａｔｔｅｒｎ＃）、スキャンモード番号（Ｓｃａｎ　Ｍｏｄｅ＃）、およびグル

ーブレート（Ｇｒｏｏｖｅ　Ｒａｔｅ）を設定することができる。
【００１４】
リズムパターンは、本実施形態では、発音タイミング、デュレーション、ベロシティ係数のパターンを組み合わせたものである。リズムパターン設定用操作子１５１は、値をインクリメントする操作ボタン１５１ａと値をデクリメントする操作ボタン１５１ｂとのペアから成り、それらの操作ボタン１５１ａ，１５１ｂで設定した値は表示器１５１ｃに表示されるとともに、上述したように、ＲＡＭ１３に格納される。
【００１５】
スキャンモードは押鍵情報をどのように走査するかというスキャン方式のセットをいう。スキャンモード設定用操作子１５２は、上記リズムパターン設定用操作子１５１と同様に、値をインクリメントする操作ボタン１５２ａと値をデクリメントする操作ボタン１５１ｂとのペアから成り、それらの操作ボタン１５１ａ，１５１ｂで設定した値は、表示器１５２ｃに表示されるとともに、ＲＡＭ１３に格納される。
【００１６】
グルーブレートは、いわゆる‘ノリ’の程度を定めるものであり、グルーブレート設定用操作子１５３はスライダから成り、そのスライドボタン１５３ａを一番下に動かすとグルーブレートパラメータに値‘０’が設定され、一番上に動かすとグルーブレートパラメータに値‘１００’が設定され、それらの中間に設定すると設定された位置に応じた、０～１００の間の値が設定される。この設定された値は、ＲＡＭ１３に格納される。
【００１７】
各パラメータの詳細については後述する。図１に戻って説明を続行する。クロックタイマ１６は、一定の周期でクロックを発生し、クロック発生のたびに、ＣＰＵに対し、クロックの発生を通知する。本実施形態では、９６クロック分の時間が４分音符の発音時間と定めれており、このクロック周期によってアルペジオのテンポが決定される。
【００１８】
データ入力端子（ＭＩＤＩ　ＩＮ）１７には、例えば図示のように鍵盤２０が接続され、鍵盤２０の演奏により生成された押鍵（ノートオン）情報、離鍵（ノートオフ）情報が入力される。また、その鍵盤２０には、ホールドペダル２１が接続されており、そのホールドペダル２１の操作により生成されたホールドオン情報、ホールドオフ情報（これらを合わせてホールド情報と称する。）も、ＭＩＤＩデータの一種として、データ入力端子１７を経由して入力される。
【００１９】
ＣＰＵ１０は、データ入力端子１７から入力される押鍵情報、離鍵情報を受信してアルペジオ演奏用の演奏情報を順次生成し、その演奏情報をデータ出力端子１８から外部に向けて出力する。データ出力端子１８には、例えば図示のように音源３０が接続されており、その音源３０では受信した演奏情報に基づいて楽音信号が生成される。その楽音信号は、例えば音源３０に内蔵された、あるいは音源３０に接続されたアンプ、スピーカ等から成るサウンドシステムにより、楽音として空間に放音される。
【００２０】
以下、先ず後述するプログラムで参照される、ＲＯＭ内に記憶された各種テーブルについて説明し、次いで、ＲＡＭ内に確保された各種メモリについて説明し、その後、それらのテーブルやメモリを用いたプログラムについて説明する。図３は、リズムパターンテーブルの基本パターンを示す図である。リズムパターンテーブルは、その各ステップがリズムの各ステップに対応しており、各ステップには、そのステップとそのステップに続く次のステップとの間の時間間隔（ステップタイム；Ｓｔｅｐ　Ｔｉｍｅ）、そのステップのリズム音の持続時間（デュレーション；Ｄｕｒａｔｉｏｎ）、そのステップのリズム音の発音の強さを規定するベロシティ係数（Ｖｅｌｏ．Ｃｏｅｆ．）が記録されており、最後にそのリズムパターンテーブルのサイズ（ステップ数；Ｓｔｅｐ　Ｓｉｚｅ）が記録されている。１ステップ分のメモリ容量は、ステップタイムが２バイト、デュレーションが２バイト、ベロシテイ係数が１バイトの合計５バイトであり、ステップ数は、１バイトである

。ステップタイム、デュレーションの欄は、前述したクロック（４分音符＝９６クロック）を単位とした数値が記録されている。

【００２１】

このリズムパターンテーブルはＲＯＭ内に複数種類記録されており、各リズムパターンテーブルには番号が付されている。図２に示すリズムパターン設定用操作子１５１を操作してそのリズムパターン番号を設定することにより、演奏者が任意のリズムパターンテーブルを選択することができる。ここでは、リズムパターンテーブル、およびそのステップ数は、それぞれ、ＲＨＹ＿ＸＸＸ＿ＴＢＬ［ｎ］、ＲＨＹ＿ＸＸＸ＿ＳＩＺＥと表記される。ＸＸＸはそのリズムパターンテーブルの名称を表わし、［ｎ］のｎは、そのリズムパターンテーブル内部のステップ番号（０≦ｎ＜ＲＨＹ＿ＸＸＸ＿ＳＩＺＥ）である。

【００２２】

またリズムパターンテーブル名ＸＸＸの第ｎステップに記録されたステップタイム、デュレーション、ベロシティ係数は、それぞれ、ＲＨＹ＿ＸＸＸ＿ＴＢＬ［ｎ］．ｓｔｅｐＴｉｍｅＲＨＹ＿ＸＸＸ＿ＴＢＬ［ｎ］．ｄｕｒａｔｉｏｎＲＨＹ＿ＸＸＸ＿ＴＢＬ［ｎ］．ｖｅｌｏＣｏｅｆと表記される。

【００２３】

次にリズムパターンテーブルの各例について説明する。図４は、４分音符用のリズムパターンテーブルおよびそのステップ数を示した図である。このリズムパターンテーブルは４分音符用であることを表わす‘４’という名称を有しており、ＲＨＹ＿４＿ＴＢＬと表記される。ＲＨＹ＿４＿ＴＢＬ［０］．ｓｔｅｐＴｉｍｅ，ＲＨＹ＿４＿ＴＢＬ［０］．ｄｕｒａｔｉｏｎ，ＲＨＹ＿４＿ＴＢＬ［０］．ｖｅｌｏＣｏｅｆは、それぞれ９６（４分音符のクロック数）、９２、１２７である。ステップ数はＲＨＹ＿４＿ＳＩＺＥと表記される。このリズムパターンテーブルＲＨＹ＿４＿ＴＢＬは１ステップのみで成り立っているため、ステップ数ＲＨＹ＿４＿ＳＩＺＥは‘１’である。

【００２４】

このリズムパターンテーブルＲＨＹ＿４＿ＴＢＬ、および以下に示すリズムパターンテーブルＲＨＹ＿ＸＸＸ＿ＴＢＬのそれぞれには、それぞれ異なるリズムパターン番号が付されており、この図４に示すリズムパターンテーブルＲＨＹ＿ＸＸＸ＿ＴＢＬのリズムパターン番号は‘０’である。図５，図６，図７，図８，図９は、それぞれ、１６分音符用（名称：１６）、ワルツ（ＷＡＬＴＺ）用、シャッフル（ＳＨＵＦＦＬＥ）用、ディミニッシュ（ＤＩＭ）用、レゲエ（ＲＥＧＧＡＥ）用のリズムパターンテーブルおよびそのステップ数を示した図である。リズムパターン番号はこの順に、１，２，３，４，５が付されている。これらの構造については上述の説明から明らかであるため、詳細説明は省略する。

【００２５】

図１０は、スキャンモードテーブルの基本パターンを示す図である。スキャンモードテーブルの各ステップには、押鍵情報を走査する方式（スキャン方式）の番号（スキャンファンクション番号；Ｓｃａｎ　Ｆｕｎｃｔｉｏｎ＃）が記録されている。このスキャンモードテーブル１ステップ分のメモリ容量は１バイトであり、ステップ数も１バイトである。

【００２６】

このスキャンモードテーブルは、ＲＯＭ１４に複数種類記憶されており、図２に示すスキャンモード設定用操作子１５２を操作することにより、演奏者が任意のスキャンモードテーブルを選択することができる。ここで、スキャンモードテーブル、およびそのステップ数は、それぞれＳＭＯＤＥ＿ＸＸＸ＿ＴＢＬ［ｎ］、ＳＭＯＤＥ＿ＸＸＸ＿ＳＩＺＥと表記される。ＸＸＸはそのスキャンモードテーブルの名称を表わし、［ｎ］のｎは、そのスキャンモードテーブル内部のステップ番号（０≦ｎ＜ＳＭＯＤＥ＿ＸＸＸ＿ＳＩＺＥ）である。

【００２７】

本実施形態における各スキャン方式（Ｓｃａｎ　Ｆｕｃｔｉｏｎ）は以下のとおりである。

【００２８】
（１）上昇スキャン方式押鍵情報を、音高の低い方から高い方へと走査する。ここでは、これを、ＳＣＡＮ＿Ｕと表記する。
【００２９】
（２）下降スキャン方式音高の高い方から低い方へと走査する。ここではこれを、ＳＣＡＮ＿Ｄと表記する。
【００３０】
（３）上昇下降スキャン方式音高の低い方から高い方へと走査し、最高音に達したら、今度は音高の高い方から低い方へと走査する。ここでは、これをはＳＣＡＮ＿ＵＤと表記する。
【００３１】
（４）ランダムスキャン方式ランダムに走査する。ここではこれを、ＳＣＡＮ＿Ｒと表記する。
【００３２】
（５）押鍵順スキャン方式押鍵情報の入力順に走査する。ここではこれを、ＳＣＡＮ＿Ｏと表記する。
【００３３】
（６）和音スキャン方式現在押鍵されているものすべてを走査する。ここではこれを、ＳＣＡＮ＿Ｃと表記する。
【００３４】
（７）最低音スキャン方式現在押鍵されているもののうちノートナンバのもっとも小さなもの（最低音）だけを走査する。ここではこれを、ＳＣＡＮ＿Ｂと表記する。
【００３５】
（８）最高音スキャン方式現在押鍵されているもののうちノートナンバのもっとも大きなもの（最高音）だけを走査する。ここではこれを、ＳＣＡＮ＿Ｔと表記する。
【００３６】
（９）最低音抜き上昇スキャン方式最低音を除いて、音高の低い方から高い方へと走査する。ここでは、これをＳＣＡＮ＿Ｕ＿ＷＯ＿Ｂ（Ｓｃａｍ　ｕｐ　ｗｉｔｈｏｕｔ　ｂａｓｓ）と表記する。
【００３７】
（１０）最低音抜きランダムスキャン方式最低音を除いて、ランダムに走査する。ここでは、これをＳＣＡＮ＿Ｒ＿ＷＯ＿Ｂと表記する。
【００３８】
（１１）最低音抜き和音スキャン方式最低音を除いて、現在押鍵されているものをすべて走査する。ここでは、これをＳＣＡＮ＿Ｃ＿ＷＯ＿Ｂと表記する。
【００３９】
（１２）最高音抜き上昇スキャン方式最高音を除いて、音高の低い方から高い方へと走査する。ここでは、これをＳＣＡＮ＿Ｕ＿ＷＯ＿Ｔと表記する。
【００４０】
次にスキャンモードテーブルの各例を示す。図１１は、上昇スキャン用のスキャンモードテーブル、およびそのステップ数を示した図である。このスキャンモードテーブルは、上昇スキャン用であることを表わす‘ＵＰ’という名称を表わしており、ＳＭＯＤＥ＿ＵＰ＿ＴＢＬと表記される。ステップ数は、ＳＭＯＤＥ＿ＵＰ＿ＳＩＺＥと表記される。このスキャンモードテーブルＳＭＯＤＥ＿ＵＰ＿ＴＢＬは、上昇スキャン方式ＳＣＡＮ＿Ｕが記録された１ステップのみで成り立っているため、ステップ数ＳＭＯＤＥ＿ＵＰ＿ＳＩＺＥは‘１’である。このスキャンモードテーブルＳＭＯＤＥ＿ＵＰ＿ＴＢＬ、および以下に示す各スキャンモードテーブルＳＭＯＤＥ＿ＸＸＸ＿ＴＢＬのそれぞれには、それぞれ異なるスキャンモード番号が付されており、この図１１に示すスキャンモードテーブルＳＭＯＤＥ＿ＵＰ＿ＴＢＬのスキャンモード番号は‘０’である。
【００４１】

図１２，図１３，図１４，図１５，図１６，図１７，図１８，図１９は、それぞれ下降スキャン用（ＤＯＷＮ）、上昇下降スキャン用（ＵＰ＿ＤＷ），ランダムスキャン用（ＲＡＮＤＯＭ）、和音スキャン用（ＣＨＯＲＤ）、押鍵順スキャン用（ＯＲＤＥＲ）、ワルツ用（ＷＡＬＴＺ）、レゲエ用（ＲＥＧＧＡＥ）、三味線用（ＳＨＡＭＩ）のスキャンモードテーブルおよびそのステップを示した図である。

【００４２】

スキャンモード番号は、この順に、１，２，３，４，５，６，７，８が付されている。これらの構造については、上述の説明から明らかであるため、詳細説明は省略する。

【００４３】

次に各種パラメータ値等が格納されるＲＡＭ内のメモリについて説明する。

【００４４】

図２０はリズムパターンパラメータを示す図である。このリズムパターンパラメータは１バイトのメモリであり、このメモリには、図２に示すパネル１５のリズムパターン設定用操作子１５１の操作により設定されたリズムパターン番号が格納される。ここではこれを、ＰＲＭ＿ＲＨＹＴＨＭと表記する。

【００４５】

図２１は、スキャンモードパラメータを示す図である。このスキャンモードパラメータも、１バイトのメモリであり、このメモリには、図２に示すパネル１５のスキャンモード設定用操作子１５２の操作により設定されたスキャンモード番号が格納される。ここでは、これを、ＰＲＭ＿ＳＭＯＤＥと表記する。図２２は、グルーブレートパラメータを示す図である。このグルーブレートパラメータも１バイトのメモリであり、このメモリには図２に示すパネル１５のグルーブレート設定用操作子の操作により設定されたグルーブレートの値が格納される。この値の範囲は、前述したように、０～１００である。ここではこれを、ＰＲＭ＿ＧＲＯＯＶＥと表記する。

【００４６】

図２３は、ノートバッファを示す図である。このノートバッファは、１２８バイトの配列であり、各１バイトは、鍵盤２０（図１参照）の各鍵（ノートナンバ）に対応しており、押鍵情報を受信すると、その押鍵情報中の押鍵強さを表わす情報（ノートオンベロシティ）が、その押鍵情報中のノートナンバに対応する領域に格納される。対応する鍵が離鍵（ノートオフ）されている場合、あるいはそれまで押鍵されていて離鍵された場合は、その領域には‘－１’が格納される。

【００４７】

ここでは、このノートバッファの各領域をＮＯＴＥＢＵＦ［ｎ］と表記する。ｎはノートナンバを表わす。図２４は、プレイバッファを示す図である。このプレイバッファは、図２３に示すノートバッファと同様１２８バイトの配列であり、各１バイトは、各ノートナンバに対応している。ただし、このプレイバッファには、図２３に示すノートバッファとは異なり、ホールド情報を加味したノートオンベロシティが記憶される。すなわち、図１に示す鍵盤２０に接続されたホールドペダル２１が踏まれると、データ入力端子１７を経由してホールドオン情報が入力され、ホールドペダル２１が離されるとデータ入力端子１７を経由してホールドオフ情報が入力されるが、ホールドオン情報を受信すると、ホールドオフ情報を受信する迄の間、この図２４に示すプレイバッファに格納された押鍵情報の消去が禁止される。ホールドオフ情報が入力されると、プレイバッファにはそのときのノートバッファの内容がコピーされ、以後、次にホールドオン情報が入力されない限り、プレイバッファは常にノートバッファと同一の内容を維持する。アルペジオ演奏を行なうにあたっては、ホールド情報が加味されたベロシティが配列された、このプレイバッファが走査される。ここでは、このプレイバッファの領域を、ＰＬＡＹＢＵＦ［ｎ］と表記する。ｎはノートナンバを表わす。

【００４８】

図２５は、カレントプレイノートナンババッファを示す図である。このカレントプレイノートナンババッファは１バイトのメモリであり、このメモリには、アルペジオ演奏中の現

在のステップのノートナンバが格納される。ここではこれを、ＣＵＲ＿ＮＯＴＥと表記する。図２６は、ホールドバッファを示す図である。このホールドバッファは１バイトのメモリであり、ここには０～１２７の数値のホールド情報が格納される。図１に示すホールドペダル２１を操作すると、データ入力端子１７からは、その操作量に応じた０～１２７の数値で表わされるホールド情報が入力され、そのホールド情報がこのホールドバッファに格納される。ここでは、この数値が６４以上であればホールドオン情報、６４未満であれはホールドオフ情報とみなされる。ここではこれを、ＨＯＬＤと表記する。

【００４９】

図２７は、オーダバッファを示す図である。このオーダバッファは、１ステップが２バイトからなる１６ステップの配列であり、各ステップには、ノートナンバとそのノートナンバのベロシティが格納される。このオーダバッファには、押鍵された鍵のノートナンバとその押鍵時のベロシティが押鍵順に格納され、離鍵されたときは、その鍵のノートナンバおよびベロシティが抹消され先頭詰めに並べ直される。ここではこれを、ＯＲＤＥＲ［ｎ］と表記する。ｎは、ステップ番号（ｎ＝０～１５）である。特に、第ｎステップのノートナンバのみ、ベロシティのみを指すときは、それぞれＯＲＤＥＲ［ｎ］．ｎｏｔｅ、ＯＲＤＥＲ［ｎ］．ｖｅｌｏと表記される。

【００５０】

図２８は、オーダライトカウンタを示す図である。このオーダライトカウンタは１バイトのメモリであり、このメモリには、押鍵情報を次に受信した場合に、その受信した押鍵情報を、図２７に示すオーダバッファＯＲＤＥＲ［］の何番目のステップに格納するかを示す値が格納される。このオーダライトカウンタのとり得る値は０～１６である。０～１５は、オーダバッファＯＲＤＥＲ［］の各ステップに対応し、１６は、オーダバッファＯＲＤＥＲ［］が満杯であることを意味する。ここではこれを、ＯＲＤＥＲ＿ＷＲと表記する。

【００５１】

図２９は、オーダポジションカウンタを示す図である。このオーダポジションカウンタは１バイトのメモリであり、このメモリには、アルペジオ演奏中の現在のステップの楽音が図２７に示すオーダバッファＯＲＤＥＲ［］の何番めのステップに対応する楽音であるかを示す値が格納される。このオーダポジションカウンタのとり得る値は－１～１５である。０～１５は、図２７に示すオーダバッファＯＲＤＥＲ［］の各ステップに対応し、－１はアルペジオ演奏の一連のステップをこれから開始するタイミング（前回の一連のステップが終了したタイミング）であることを意味している。ここではこれを、ＣＵＲ＿ＯＲＤＥＲと表記する。

【００５２】

図３０は、リズムパターンテーブルポジションカウンタを示す図である。このリズムパターンテーブルポジションカウンタは１バイトのメモリであり、このメモリは、現在選択されているリズムパターンテーブル（図３、および図４～図９の各例を参照）中のステップを指し示すポインタである。このポインタはアルペジオ演奏中の現在のリズムステップに対応している。このリズムパターンテーブルポジションカウンタのとり得る値は０以上、かつ現在選択されているリズムパターンテーブルのステップ数ＲＨＹ＿ＸＸＸ＿ＳＩＺＥ未満である。ここではこれを、ＣＵＲ＿ＲＨＹと表記する。

【００５３】

図３１は、スキャンモードテーブルポジションカウンタを示す図である。このスキャンモードテーブルポジションカウンタは１バイトのメモリであり、現在選択されているスキャンモードテーブル（図１０、および図１１～図１９の各例を参照）中のステップを指し示すポインタである。このポインタも、アルペジオ演奏中の現在のリズムステップに対応している。スキャンモードテーブルポジションカウンタのとり得る値は、０以上、かつ現在選択されているスキャンモードテーブルのステップ数ＳＭＯＤＥ＿ＸＸＸ＿ＳＩＺＥ未満である。ここでは、これをＣＵＲ＿ＳＭＯＤＥと表記する。

【００５４】

図３２は、クロックカウンタを示す図である。このクロックカウンタは２バイト（１６ビット）のメモリであり、図１に示すクロックタイマ１６がクロックを１つ発生するたびにこのクロックカウンタの値が‘１’ずつ増加する。このクロックカウンタの値の範囲は、００００ｈ～ＦＦＦＦｈであり、ＦＦＦＦｈの次は００００ｈに戻り、電源がオンである間、無限に巡回する。このクロックの値の増加に基づいてアルペジオが進行する。ここではこれを、ＣＬＯＣＫと表記する。

【００５５】

図３３は、ネクストクロックバッファを示す図である。このネクストクロックバッファは２バイト（１６ビット）のメモリであり、このメモリには、アルペジオ演奏の次のステップの時刻が格納される。すなわち、図３２に示すＣＬＯＣＫが増加して、このネクストクロックバッファに格納されている値に達すると、押鍵情報を走査してノートオンデータを送信する。１回のステップの処理が終了すると、このネクストクロックバッファの値は次のステップの時刻に更新される。このネクストクロックバッファのとり得る値の範囲は００００ｈ～ＦＦＦＦｈである。ここではこれを、ＮＥＸＴＣＬＫと表記する。

【００５６】

図３４は、ノートオフリザベーションバッファを示す図である。このノートオフリザベーションバッファは３バイト×１６の配列であり、このノートオフリザベーションバッファには、ノートオフすべきノートナンバと、それをノートオフデータとして実際に送信すべきタイミングとがペアで格納される。この送信タイミングは図３２のＣＬＯＣＫの値で表わされる。ＣＬＯＣＫが順次増加していってこのノートオフリザベーションバッファに格納されている送信タイミングのうちのいずれかの送信タイミングに到達すると、その送信タイミングとペアで格納されているノートナンバについてノートオフデータが送信される。このノートオフリザベーションバッファ中の有効なデータが格納されていない領域、あるいはノートオフデータを送信し終わった領域には‘－１’が格納される。

【００５７】

ここではこれを、ＯＦＦＲＳＶ［ｎ］と表記する。ｎは、ノートナンバとその送信タイミングとのペアのｎ番目の格納領域を意味する。特にＯＦＦＲＳＶ［ｎ］の中ノートナンバ、送信タイミングを指し示すときは、それぞれ、ＯＦＦＲＳＶ［ｎ］．ｎｏｔｅ，ＯＦＦＲＳＶ［ｎ］．ｃｌｏｃｋと表記される。図３５は、エンドフラグを示す図である。このエンドフラグは、ここでは１バイトのメモリで構成されているが、フラグの性質上１ビットで構成してもよい。このエンドフラグは‘０’と‘１’との２値をとり、‘０’は、アルペジオ演奏中の１回のステップの発音のための走査が未了であることを示し、‘１’は、１回のステップの発音のための走査が終了したことを示している。ここではこれを、ＥＮＤ＿ＦＬＧと表記する。

【００５８】

図３６は、ベースノートバッファを示す図である。このベースノートバッファは１バイトのメモリであり、このメモリには、ホールド情報を加味した押鍵中の鍵のうちの最低音の鍵のノートナンバが格納される。ここではこれを、ＬＯと表記する。図３７は、トップノートバッファを示す図である。このトップノートバッファは１バイトのメモリであり、このメモリにはホールド情報を加味した押鍵中の鍵のうちの最高音の鍵のノートナンバが格納される。ここではこれを、ＨＩと表記する。

【００５９】

以上説明した各メモリの他にも作業領域として使用するメモリもあるが、後述するプログラムの説明で足りるため、ここでは、ここに説明した以外のメモリについて１つずつ取りあげた説明は省略する。以下に、これまで説明したテーブルやメモリを用いてＣＰＵで実行されるプログラムについて説明する。

【００６０】

図３８は、電源オン時に動作を開始し、電源オフ時まで動作し続けるゼネラルプログラムのフローチャートである。電源が投入されると、先ず所定の初期化が行なわれ（ステップ３８＿１）、それ以後、クロックタイマ１６（図１参照）からクロックの発生が通知され

たか否か（ステップ３８＿２）、パネル１５（図１，図２参照）が操作されいずれかのパラメータがエディットされたか否か（ステップ３８＿４）、ノートオンデータ（押鍵情報）が入力されたか否か（ステップ３８＿６）、ノートオフデータ（離鍵情報）が入力されたか否か（ステップ３８＿８）、ホールドデータ（ホールド情報）が入力されたか否か（ステップ３８＿１０）が順次循環的にモニタされ、各ステップ３８＿２，３８＿４，３８＿６，３８＿８，３８＿１０で各イベントが発生したことが認識されると、それぞれ、クロック処理（ステップ３８＿３）、エディット処理（ステップ３８＿５）、ノートオン処理（ステップ３８＿７）、ノートオフ処理（ステップ３８＿９）、ホールドデータ処理（ステップ３８＿１１，３８＿１２，３８＿１３）が行なわれる。

【００６１】

ホールドデータ処理では、先ずＨＯＬＤ（図２６参照）にホールド値が格納され（ステップ３８＿１１）、そのホールド値が、ホールドオンを表わす６４以上であるか否かが判定され（ステップ３８＿１２）、ホールドオフを表わす６３以下の場合、ホールドオフ処理が行なわれる（ステップ３８＿１３）。図３９は、初期化処理ルーチンのフローチャートである。この初期化処理ルーチンは、図３８に示すゼネラルプログラムのステップ３８＿１で実行される。

【００６２】

この初期化処理では、先ず、押鍵時のベロシティが押鍵順に格納されるオーダバッファＯＲＤＥＲ［］（図２７参照）がクリアされ、そのオーダバッファＯＲＤＥＲ［］の格納ポインタＯＲＤＥＲ＿ＷＲ（図２８参照）が‘０’（ＯＲＤＥＲ［］の先頭）に初期化され、さらに、オーダポジションカウンタＣＵＲ＿ＯＲＤＥＲ（図２９）が、アルペジオ演奏の一連のステップをこれから開始するタイミングであることを表わす‘－１’に初期化される（ステップ３９＿１）。

【００６３】

次いで、クロックタイマ１６（図１参照）で発生したクロックをカウントするクロックカウンタＣＬＯＣＫ（図３２参照）が‘０’に初期化され（ステップ３９＿２）、次の演奏情報の生成のタイミングが格納されるネクストクロックバッファＮＥＸＴＣＬＫ（図３３参照）も‘０’に初期化される（ステップ３９＿３）。さらに、ノートオフすべきノートナンバやそのノートオフすべきタイミングが格納されるノートオフリザベーションバッファＯＦＦＲＳＶ［］（図３４参照）がクリアされ（ステップ３９＿４）、鍵盤２０（図１参照）の鍵の配列に対応した配列を有し押鍵された鍵のベロシティが格納されるノートバッファＮＯＴＥＢＵＦ［］（図２３参照）がクリアされ（ステップ３９＿５）、さらに、ホールド情報を加味したベロシティが格納されるプレイバッファＰＬＡＹＢＵＦ［］（図２４参照）がクリアされる（ステップ３９＿６）。さらにホールド値が格納されるホールドバッファＨＯＬＤ（図２６参照）が‘０’にクリアされ（ステップ３９＿７）、さらに、スキャナをリセットするリセットスキャナルーチンが実行される（ステップ３９＿８）。

【００６４】

図４０は、リセットスキャナルーチンのフローチャートである。リセットスキャナルーチンは、図３９に示す初期化ルーチンのステップ３９＿８で実行される。ここでは、現在アルペジオ演奏中のノートナンバが格納されるカレントプレイノートナンババッファＣＵＲ＿ＮＯＴＥ（図２５参照）に空きを表わす‘－１’が格納され、オーダバッファＯＲＤＥＲ［］（図２７参照）の、現在アルペジオ演奏中のステップを指し示すオーダポジションカウンタＣＵＲ＿ＯＲＤＥＲ（図２９参照）にも、空きを表わす‘－１’が格納され、さらに、スキャンモードテーブル、リズムパターンテーブルの各ポインタであるスキャンモードテーブルポジションカウンタＣＵＲ＿ＳＭＯＤＥ（図３１参照）、リズムパターンテーブルポジションカウンタＣＵＲ＿ＲＨＹ（図３０参照）が、先頭を指し示す‘０’に初期化される。さらに、クロックＣＬＯＣＫ（図３２参照）に１を加えた値がネクストクロックＮＥＸＴＣＬＫ（図３３）参照に格納される。ＣＬＯＣＫに‘１’を加えた値をＮＥＸＴＣＬＫとする理由は、後述するようにＮＥＸＴＣＬＫ＝ＣＬＯＣＫのときに発音され

る（ノートオンデータが送信される）が、このプログラムが動作中にＣＬＯＣＫがインクリメントされてＣＬＯＣＫが発音開始タイミングを示すＮＥＸＴＣＬＫを越えてしまい、その発音が行なわれないことが生じる可能性をなくすためである。

【００６５】

図４０に示すリセットスキャナルーチンでは、さらに、演奏中のアルペジオ演奏の１回ステップの発音のための操作が未了か終了かを示すエンドフラグＥＮＤ＿ＦＬＧに、未了を示す‘０’が格納される。図４１は、クロック処理ルーチンのフローチャートである。このクロック処理ルーチンは、図３８に示すゼネラルプログラムのステップ３８＿３で実行される。

【００６６】

このクロック処理ルーチンでは、先ずＣＬＯＣＫ（図３２参照）がインクリメントされる（ステップ４１＿１）。次に、ＯＦＦＲＳＶ［］（図３４参照）をサーチし、そこに現在のタイミングで送信すべきノートオフデータが存在するかどうかを調べ、現在のタイミングで送信すべきノートオフデータが存在する場合にそのノートオフデータを送信する（ステップ４１＿２～４１＿８）。

【００６７】

具体的には、先ずｉに‘０’を置いて（ステップ４１＿２）、ＯＦＦＲＳＶ［ｉ］．ｎｏｔｅに有効なノートオフデータが格納されているか（０～１２７）否か（－１）を調べ（ステップ４１＿３）、‘－１’の場合はｉをインクリメントして（ステップ４１＿７）、ｉが１６に達するまで（ステップ４１＿８）、そのサーチが行なわれる。ステップ４１＿３で有効なノートオフデータの格納が確認されるとステップ４１＿４に進み、ＯＦＦＲＳＶ［ｉ］．ｃｌｏｃｋが現在のＣＬＯＣＫと等しいかどうかが判定され、等しい場合にそのノートオフデータを演奏情報として送信し（ステップ４１＿５）、そのノートオフデータが格納されていた領域のＯＦＦＲＳＶ［ｉ］．ｎｏｔｅに‘－１’を書き込む（ステップ４１＿６）。

【００６８】

ノートオフデータのサーチが終了すると、今度は送信すべきノートオンデータを見つけに行く処理を行なう（ステップ４１＿９～４１＿１７）。具体的には、先ず現在のＣＬＯＣＫがＮＥＸＴＣＬＫに達したか否か、すなわちノートオンデータを送信すべきタイミングに達したか否かが判定され（ステップ４１＿９）、未だそのタイミングに達していないときはそのまま終了する。

【００６９】

ＮＥＸＴＣＬＫ＝ＣＬＯＣＫのときは、ステップ４１＿１０に進み、押鍵情報を走査するスキャンノートオンルーチンが実行される（ステップ４１＿１０）。このスキャンノートオンルーチンの詳細は後述するが、このスキャンノートオンルーチンでは、送信すべきノートナンバのうちの１つがＮＴに格納される。ＮＴ＝－１は、送信すべきノートが存在しないことを意味している。ステップ４１＿１１においてＮＴ＝－１ではない、すなわち送信すべきノートが存在すると判定されると、ステップ４１＿１２に進み、ノートオフ予約ルーチンが実行される。このノートオフ予約ルーチンの詳細についても後述するが、このノートオフ予約ルーチンでは、これから送信しようとするノートオンデータに対応するノートオフデータをＯＦＦＲＳＶ［］（図３４参照）に格納することにより、ノートオフ予約を行なう。ＯＦＦＲＳＶ［］が満杯のときは、ＮＴに、ノートオフ予約ができないことを示す‘－１’が格納される。

【００７０】

ステップ４１＿１３ではＮＴ＝－１かどうかを調べることによりノートオフ予約が行なわれたか否かを知り、ノートオフ予約不能であったときはノートオンデータを送信せずにステップ４１＿１０に戻り、送信すべき次のノートオンデータのサーチを行なう。ステップ４１＿１３でＮＴ≠－１であったとき、すなわち正常にノートオフ予約が行なわれたときは、ステップ４１＿１４に進み、ベロシティ生成ルーチンが実行される。このベロシティ生成ルーチンの詳細についても後述する。このベロシティ生成ルーチンでは、後述する演

算により送信しようとするノートオンデータのベロシティが生成される。その後ステップ４１＿１５においてそのノートオンデータが送信され、ステップ４１＿１０に戻り、現在のタイミングでさらに送信すべきノートオンデータが存在するかどうかのサーチが行なわれる。

【００７１】

ステップ４１＿１０でＮＴに‘－１’が格納された場合、すなわち現在のタイミングで送信すべきノートオンデータが存在しない（もしくは、現在のタイミングで送信すべきノートオンデータは全て送信してしまった）場合、ステップ４１＿１１を経由してステップ４１＿１６に進む。ステップ４１＿１６では、ネクストクロック更新ルーチンが実行される。このネクストクロック更新ルーチンの詳細は後述するが、ここでは、ノートオンデータ送信の次のタイミングがＮＥＸＴＣＬＫに格納される。

【００７２】

その後、ステップ４１＿１７に進み、スキャナ更新ルーチンが実行される。このスキャナ更新ルーチンでスキャナの更新が行なわれる。このスキャナ更新ルーチンの詳細についても後述する。図４２は、スキャンノートオンルーチンのフローチャートである。このスキャンノートオンルーチンは、図４１のクロック処理ルーチンのステップ４１＿１０で実行される。

【００７３】

このスキャンノートオンルーチンでは、先ずスキャンモード番号が格納されているＰＲＭ＿ＳＭＯＤＥ（図２１参照）が参照され（ステップ４２＿１）、そのＰＲＭ＿ＳＭＯＤＥに格納されているスキャンモード番号に応じたスキャンモードテーブル（図１０、および図１１～図１９の各例参照）中の現在のステップ番号ＣＵＲ＿ＳＭＯＤＥに格納されたスキャン方式（スキャンファンクション番号）ＳＭＯＤＥ＿ＸＸＸ＿ＴＢＬ［ＣＵＲ＿ＭＯＤＥ］が作業領域ｆｕｎｃに格納される（ステップ４２＿２）。次いで、そのｆｕｎｃに格納されたスキャンファンクション番号がＳＣＡＮ＿Ｕ，ＳＣＡＮ＿Ｄ，ＳＣＡＮ＿ＵＤ，……，ＳＣＡＮ＿Ｕ＿ＷＯ＿Ｔのいずれを指しているかに応じて、それぞれ、図示の、スキャンアップルーチン（ステップ４２＿４）、スキャンダウンルーチン（ステップ４２＿５）、スキャンアップダウンルーチン（ステップ４２＿６）、……、スキャンアップウィズアウトトップルーチン（４２＿１５）が実行される。

【００７４】

以下では、代表的なものとして、スキャンアップルーチン（ステップ４２＿４）、スキャンオーダルーチン（ステップ４２＿８）、スキャンコードルーチン（ステップ４２＿９）、スキャンベースルーチン（ステップ４２＿１０）、スキャンコードウィズアウトベース（ステップ４２＿１４）について説明する。図４３は、スキャンアップルーチンのフローチャートである。このスキャンアップルーチンは、前述した上昇スキャン方式を実現するルーチンである。

【００７５】

ここでは、先ず、ＥＮＤ＿ＦＬＧ（図３５参照）が‘１’か‘０’か、すなわち、今回の１ステップ分の発音のための走査が既に終了しているか否かが判定され（ステップ４３＿１）、ＥＮＤ＿ＦＬＧ＝１の場合、すなわち、今回の走査が既に終了している場合は、ステップ４３＿２に進み、ＮＴに‘－１’が格納されて終了する。ただし、今回のステップで最初にこのスキャンアップルーチンが実行される際はＥＮＤ＿ＦＬＧ＝０である。

【００７６】

ステップ４３＿１でＥＮＤ＿ＦＬＧが‘１’でなかったときは、ステップ４３＿３に進み、ＰＬＡＹＢＵＦ［］（図２４参照）に１つでもノートオンデータが存在するか否かが判定され、ノートオンデータが１つも存在しないときは、発音しようがないので、ステップ４３＿２に進み、ＮＴに‘－１’が格納されて終了する。

【００７７】

ＰＬＡＹＢＵＦ［］にノートオン情報が存在することが認識されると、次に、上昇スキャン方式に従って、次に送信すべきノートオンデータが走査される（ステップ４３＿４～４

３＿７）。具体的には、ＣＵＲ＿ＮＯＴＥ（図２５参照）は、直前に送信したノートオンデータのノートナンバが格納されているため、そのＣＵＲ＿ＮＯＴＥを次のノートナンバに進め（ステップ４３＿４）、そのノートナンバが１２８に達すると（ステップ４３＿５）、ＣＵＲ＿ＮＯＴＥに‘０’が格納される（ステップ４３＿６）、このようにして更新されたＣＵＲ＿ＮＯＴＥを用いて、ＰＬＡＹＢＵＦ［ＣＵＲ＿ＮＯＴＥ］に有効なベロシティデータが格納されているか否かが判定される（ステップ４３＿７）。ＰＬＡＹＢＵＦ［ＣＵＲ＿ＮＯＴＥ］＝－１、すなわちそこには有効なベロシティデータが格納されていないときは、ステップ４３＿４に進み、ＣＵＲ＿ＮＯＴＥが再度インクリメントされ、ＰＬＡＹＢＵＦ［］の有効なベロシティが格納されている領域を見い出すべく、音高の低い方から高い方へと走査される。

【００７８】

このようにして、有効なベロシティデータが格納されているＰＬＡＹＢＵＦ［］が見い出されると、そのときのＣＵＲ＿ＮＯＴＥがＮＴに格納され（ステップ４３＿８）、ＰＬＡＹＢＵＦ［ＮＴ］に格納されている、ＣＵＲ＿ＮＯＴＥのベロシティデータがＶＬに格納され（ステップ４３＿９）、上昇スキャン方式では１つのステップでは１つのノートオンデータしか送信しないためＥＮＤ＿ＦＬＧに走査の終了を示す‘１’が格納される（ステップ４３＿１０）。

【００７９】

図４４は、スキャンオーダルーチンのフローチャートである。このスキャンオーダルーチンは、前述した押鍵順スキャン方式を実現するルーチンである。ステップ４４＿１，４４＿２は、図４３に示すスキャンアップルーチンのステップ４３＿１，４３＿２と同様であり、説明は省略する。ステップ４４＿３では、ＯＲＤＥＲ＿ＷＲ＝０か否かが調べられる。ＯＤＥＲ＿ＷＲは、図２８と参照して説明したように、押鍵情報を押鍵順に格納するＯＲＤＥＲ［］（図２７参照）のポインタであり、ＯＲＤＥＲ　ＷＲ＝０ということは、ＯＲＤＥＲ［］が空であって、押鍵順アルペジオ演奏を行なうことができないことを意味しており、この場合もステップ４４＿２に進み、ＮＴに‘－１’が格納されて終了する。

【００８０】

ステップ４４＿３でＯＲＤＥＲ＿ＷＲ＝０ではない、すなわちＯＲＤＥＲ［］が空ではないと判定されると、ステップ４４＿４に進んでＣＵＲ＿ＯＲＤＥＲ（図２９参照）がインクリメントされ、ＣＵＲ＿ＯＲＤＥＲがＯＲＤＥＲ［］の書き込まれる最大ステップを越えたか否かが判定されて（ステップ４４＿５）、越えた場合は、ＣＵＲ＿ＯＲＤＥＲに０が格納される（ステップ４４＿６）。このようにして値が進められたＣＵＲ＿ＯＲＤＥＲを用いて、次に発音すべきノートナンバＯＲＤＥＲ［ＣＵＲ＿ＯＲＤＥＲ］．ｎｏｔｅがＮＴに格納され（ステップ４４＿７）、そのノートナンバのベロシティＯＲＤＥＲ［ＣＵＲ＿ＯＲＤＥＲ］．ｖｅｌｏがＶＬに格納される（ステップ４４＿８）。この押鍵順スキャン方式においても、１つのステップでは１つのノートオンデータしか送信しないため、ＥＮＤ＿ＦＬＧに、走査の終了を示す‘１’が格納される（ステップ４４＿９）。

【００８１】

図４５は、スキャンコードルーチンのプログラムである。このスキャンコードルーチンは、前述した和音スキャン方式を実現するルーチンである。ステップ４５＿１，４５＿２，４５＿３は、図４３に示すスキャンアップルーチンのステップ４３＿１，４３＿２，４３＿３と同様であり、説明は省略する。今回のステップにおいて、このスキャンコードルーチンが第１回目に呼ばれたときは、ＣＵＲ＿ＮＯＴＥには‘－１’が格納されており、したがってステップ４５＿４，４５＿５，４５＿６のループでは、ＣＵＲ＿ＮＯＴＥ＝０から始まり、ＣＵＲ＿ＮＯＴＥ＝１２７まで走査される。その走査の途中で、ＰＬＡＹＢＵＦ［ＣＵＲ＿ＮＯＴＥ］に‘－１’以外の値、すなわち有効なベロシティデータが格納されていることを見出すと、ステップ４５＿７に進み、ＮＴにそのノート番号ＣＵＲ＿ＮＯＴＥを格納し、ＶＬにそのノートナンバのベロシティデータＰＬＡＹＢＵＦ［ＮＴ］を格納して（ステップ４５＿８）、このルーチンを一旦抜ける。

【００８２】

和音スキャン方式では、ＰＬＡＹＢＵＦ［］の全域を走査し有効なベロシティデータの格納された全てのノートオンデータを送信する必要があるため、ＣＵＲ＿ＮＯＴＥ≧１２８に達する前にこのスキャンコードルーチンを抜けるときはＥＮＤ＿ＦＬＧには‘１’には格納しない。このスキャンコードルーチンを抜けると、図４２に示すスキャンノートオンルーチンも抜け、図４１のクロック処理ルーチンのステップ４１＿１０を抜けることになる。ＮＴ＝－１ではないときは、ステップ４１＿１２，４１＿１３，４１＿１４，４１＿１５と進んで１つのノートオンデータの送信が行なわれ、再度ステップ４１＿１０に戻り、図４２のスキャンノートオンルーチンを経由して、図４５のスキャンコードルーチンが再度実行される。そのときには、再度実行を開始したときの、すなわち前回このルーチンを抜けたときＣＵＲ＿ＮＯＴＥの次の値ＣＵＲ＿ＮＯＴＥから走査が行なわれる（ステップ４５＿４）。ステップ４５＿５においてＣＵＲ＿ＮＯＴＥ≧１２８、すなわちＰＬＡＹＢＵＦ［］の走査が終了したことが判定されると、ステップ４５＿９に進んでＮＴに‘－１’が格納され、さらにステップ４５＿１０に進んで、ＥＮＤ＿ＦＬＧに‘１’が格納される。

【００８３】

図４６は、スキャンベースルーチンのプログラムである。このスキャンベースルーチンは、前述した最低音スキャン方式を実現するルーチンである。ステップ４６＿１，４６＿２，４６＿３は、図４３に示すスキャンアップルーチンのステップ４３＿１，４３＿２，４３＿３と同様であり、説明は省略する。ステップ４６＿４では、最低音ＬＯ（図３６参照）がＮＴに格納される。ＬＯへの最低音の格納処理については後述する。

【００８４】

ステップ４６＿５では、その最低音ＬＯのベロシティＰＬＡＹＢＵＦ［ＮＴ］がＶＬに格納される。さらにステップ４６＿６において、ＥＮＤ＿ＦＬＧに‘１’が格納される。最低音は１つしか存在しないからである。図４７は、スキャンコードウィズアウトベースルーチンのプログラムである。このルーチンは、前述した最低音抜き和音スキャン方式を実現するルーチンである。

【００８５】

ステップ４７＿１，４７＿２，４７＿３は、図４３に示すスキャンアップルーチンのステップ４３＿１，４３＿２，４３＿３と同様であり、説明は省略する。ステップ４７＿４では、ＰＬＡＹＢＵＦ［］にノートオンの数が１つだけ存在するか、それとも複数存在するかが判定される、ノートオンの数が１つだけの場合、本来は最低音抜き和音スキャン方式は成立しないが、ここでは、その唯一のノートオンのノートナンバを送信することとし、ステップ４７＿５に進んで、ＮＴにそのノートのノートナンバを格納し、そのノートナンバのベロシティＰＬＡＹＢＵＦ［ＮＴ］をＶＬに格納し（ステップ４７＿６）、ＥＮＤ＿ＦＬＧに‘１’を格納する（ステップ４７＿７）。

【００８６】

ステップ４７＿４においてノートオンの数が複数存在することが確認されるとステップ４７＿８に進む。今回のステップにおいて、このスキャンコードウィズアウトベースルーチンが第１回目に呼ばれたときは、図４５に示すスキャンコードルーチンの場合と同様に、ＣＵＲ＿ＮＯＴＥには‘－１’が格納されている。ＣＵＲ＿ＮＯＴＥに‘－１’を格納する処理については後述する。したがって、ここでは、ＰＬＡＹＢＵＦ［］の全域にわたって、音高の低い方から高い方へと順次走査される。ここでの走査の手順（ステップ４７＿８～４７＿１５）は、図４５に示すスキャンコードルーチンにおける走査の手順（ステップ４５＿４～４５＿１０）と同様であるが、ステップ４７＿１１において、ＣＵＲ＿ＮＯＴＥが最低量ＬＯであるか否かを判定し、最低音の場合にそのノートオンを無視するようにしている点のみが異なる。詳細説明は省略する。尚、ＬＯに最低音を格納する処理については後述する。

【００８７】

以上では、図４２に示すスキャンノートオンルーチン中で実行される、スキャンアップルーチン（ステップ４２＿４）、スキャンオーダルーチン（ステップ４２＿８）、スキャン

コードルーチン（ステップ４２＿９）、スキャンベースルーチン（４２＿１０）、スキャンコードウィズアウトベースルーチン（ステップ４２＿１３）について説明した。スキャンダウンルーチン（ステップ４２＿５）、スキャンアップダウンルーチン（ステップ４２＿６）、スキャンランダムルーチン（ステップ４２＿７）、スキャントップルーチン（ステップ４２＿１１）、スキャンアップウィズアウトベースルーチン（ステップ４２＿１２）、スキャンランダムウィズアウトベースルーチン（ステップ４２＿１３）、スキャンアップウィズアウトトップルーチン（４２＿１４）は、前述した、それぞれ、下降スキャン方式、上昇下降スキャン方式、ランダムスキャン方式、最高音スキャン方式、最低音抜き上昇スキャン方式、最低音抜きランダムスキャン方式、最高音抜き上昇スキャン方式を実現するルーチンであるが、上述した各種のルーチン（図４３～図４７）から自明であるため、ここではそれらについての図示および説明は省略する。

【００８８】

図４８は、ノートオフ予約ルーチンのプログラムである。このノートオフ予約ルーチンは、図４１に示すクロック処理ルーチンのステップ４１＿１２で実行される。このノートオフ予約ルーチンでは、ノートオフ予約に用いられるＯＦＦＲＳＶ［］（図３４参照）をサーチしてその空いている領域を見つけ、一方、リズムパターンテーブルから読み出したデュレーションと、ステップタイムと、グルーブレート設定用操作子１５３（図２参照）の操作により設定されたグルーブレートＰＲＭ＿ＧＲＯＯＶＥとに従ってデュレーションを計算し、そのデュレーションに基づいてノートオフのタイミングを求めて、そのノートオフのタイミングをＯＦＦＲＳＶ［］の空き領域に設定することにより、ノートオフ予約を行なう。

【００８９】

すなわち、先ずステップ４８＿１においてｉに‘０’を設定し、ステップ４８＿２においてＯＦＦＲＳＶ［ｉ］．ｎｏｔｅ＝－１であるか否か、すなわち、ＯＦＦＲＳＶ［ｉ］が空いているか否かが判定される。ＯＦＦＲＳＶ［ｉ］．ｎｏｔｅ＝－１ではないとき、すなわちＯＦＦＲＳＶ［ｉ］が空いていないときは、ｉがインクリメントされ（ステップ４８＿３）、ｉがＯＦＦＲＳＶ［］の配列の大きさを越えたか否かが判定され（ステップ４８＿４）、その配列以内であるときはステップ４８＿２に戻ってインクリメントされた新たなｉに対してＯＦＦＲＳＶ［ｉ］．ｎｏｔｅ＝－１か否かに判定される。これを繰り返し、空き領域を発見できないままｉ＝１６に達すると、ステップ４８＿５に進み、ＮＴに、ＯＦＦＲＳＶ［］が満杯であることを示す‘－１’を格納して終了する。

【００９０】

ステップ４８＿２においてＯＦＦＲＳＶ［ｉ］．ｎｏｔｅ＝－１、すなわち空き領域を見つけたときは、その空き領域ＯＦＦＲＳＶ［ｉ］．ｎｏｔｅに、図４１のクロック処理ルーチンのステップ４１＿１０で得られたノートナンバＮＴが格納される（ステップ４８＿６）。次いで、リズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭ（図２０参照）に応じて（ステップ４８＿７）、そのリズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭに対応するリズムパターンテーブルＲＨＹ＿ＸＸＸ＿ＴＢＬ［］（図３、および図４～図９の各例を参照）の現在のステップ番号ＣＵＲ＿ＲＨＹ（図３０参照）のステップに格納されている、デュレーションＲＨＹ＿ＸＸＸ＿ＴＢＬ［ＣＵＲ＿ＲＨＹ］．ｄｕｒａｔｉｏｎおよびステップタイムＲＨＹ＿ＸＸＸ＿ＴＢＬ［ＣＵＲ＿ＲＨＹ］．ｓｔｅｐＴｉｍｅが読み出されて、それぞれＤ１，Ｄ２に格納される（ステップ４８＿８）。

【００９１】

ステップ４８＿９では、デュレーションＤ１，ステップタイムＤ２、およびグルーブレートＰＲＭ＿ＧＲＯＯＶＥ（図２２参照）に基づいて、そのノートナンバのデュレーションＤＵＲＡＴＩＯＮが、ＤＵＲＡＴＩＯＮ＝（Ｄ１－Ｄ２）×ＰＲＭ＿ＧＲＯＯＶＥ／１００＋Ｄ２の計算により求められる。

【００９２】

この計算結果は、ＰＲＭ＿ＧＲＯＯＶＥ＝０のときは、ＤＵＲＡＴＩＯＮ＝Ｄ２、すなわちステップタイムと同値となり、そのアルペジオ演奏にテヌートのような効果を与え、Ｐ

ＲＭ＿ＧＲＯＯＶＥ＝１００のときは、ＤＵＲＡＴＩＯＮ＝Ｄ１、すなわちリズムパターンテーブルＲＨＹ＿ＸＸＸ＿ＴＢＬ［］に格納されたデュレーションのままとなる。このリズムパターンテーブルＲＨＹ＿ＸＸＸ＿ＴＢＬ［］にスタッカートのような短いデュレーションを格納しておくと、図２に示すグルーブレート設定用操作子１５３の操作により、スタッカートのようなアルペジオ演奏からテヌートのようなアルペジオ演奏まで、各音のデュレーションを連続的に変えることができる。このようにして求めたＤＵＲＡＴＩＯＮは、ＣＬＯＣＫ（図３２参照）と加算されて、ＯＦＦＲＳＶ［ｉ］．ｃｌｏｃｋに格納される（ステップ４８＿１０）。

【００９３】

図４９は、ベロシティ生成ルーチンのフローチャートで、図５０は、図４９に示すベロシティ生成ルーチンで生成されるベロシティの説明図である。図４９に示すベロシティ生成ルーチンは、図４１に示すクロック処理ルーチンのステップ（４１＿１４）で実行される。図４９に示すベロシティ生成ルーチンでは、リズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭ（図２０参照）が参照され（ステップ４９＿１）、そのリズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭに対応するリズムパターンテーブルＲＨＹ＿ＸＸＸ＿ＴＢＬ（図３、および図４～図９の各例参照）の現在のステップＣＵＲ＿ＲＨＹ（図３８参照）のベロシティ係数ＲＨＹ＿ＸＸＸ＿ＴＢＬ［ＣＵＲ＿ＲＨＹ］．ｖｅｌｏＣｏｅｆが読み出されてＣＯＥＦに格納される（ステップ４９＿２）。その後、ステップ４９＿３において、ベロシティ係数ＣＯＥＦと、押鍵により生成されたベロシティＶＬとの間で、ＶＬ２＝ＶＬ×ＣＯＥＦ／１２７の計算が行なわれ、さらにステップ４９＿４において、ＶＬ－（ＶＬ－ＶＬ２）×ＰＲＭ＿ＧＲＯＯＶＥ／１００の計算が行なわれて、その計算の結果求められたベロシティが再度ＶＬに格納される。

【００９４】

ここでは、図５０に示すように、図２に示すグルーブレート設定用操作子１５３の操作により設定されたグルーブレートＰＲＭ＿ＧＲＯＯＶＥがＰＲＭ＿ＧＲＯＯＶＥ＝０の時、リズムパターンテーブルＲＨＹ＿ＸＸＸ＿ＴＢＬ［］に格納されたベロシティにかかわらず、押鍵ベロシティがそのまま採用され、ＰＲＭ＿ＧＲＯＯＶＥ＝１００のときは、（押鍵ベロシティ×ベロシティ係数）が採用され、グルーブレート設定用操作子１５３をその中間に設定すると、その位置に応じて、押鍵ベロシティと、（押鍵ベロシティ×ベロシティ係数）との間で連続的に変化するベロシティが採用される。

【００９５】

このようにグルーブレートを変えることにより、いわゆる‘ノリ’の程度を変化させることができる。図４１のクロック処理ルーチンのステップ４１＿１４では、上記のようにしてベロシティが求められ、ステップ４１＿１５では、このようにして求められたベロシティデータを伴ったノートオンデータが送信される。

【００９６】

図５１は、ネクストクロック更新ルーチンのフローチャートである。このネクストクロック更新ルーチンは、図４０に示すクロック処理ルーチンのステップ４１＿１６で実行される。このネクストクロック更新ルーチンでは、先ずリズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭ（図２０参照）が参照され、そこに格納されたリズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭに応じたリズムパターンテーブルＲＨＹ＿ＸＸＸ＿ＴＢＬ［］の現在のステップ番号ＣＵＲ＿ＲＨＹ（図３０参照）のステップに格納されているステップタイムＲＨＹ＿ＸＸＸ＿ＴＢＬ［ＣＵＲ＿ＲＨＹ］．ｓｔｅｐＴｉｍｅが読み出されてＳＴＥＰに格納され（ステップ５１＿２）、ＣＬＯＣＫ（図３２参照）にそのＳＴＥＰが加算されてＮＥＸＴＣＬＫ（図３３参照）に格納される（ステップ５１＿３）。これにより、次の発音タイミングがＮＥＸＴＣＬＫに設定される。

【００９７】

図５２、５３は、それぞれ、スキャナ更新ルーチンのフローチャートの前半部分、後半部分である。このスキャナ更新ルーチンは、図４１に示すクロック処理ルーチンのステップ４１＿１７で実行される。このスキャナ更新ルーチンでは、先ず、前回のノートオンデー

タの走査の最後でＥＮＤ＿ＦＬＧに‘１’が格納されたのを解除してＥＮＤ＿ＦＬＧに‘０’を格納し（ステップ５２＿１）、次に、リズムパターンテーブルの現在のステップＣＵＲ＿ＲＨＹ（図３０参照）が更新される（ステップ５２＿２～５２＿６）。具体的には、ステップ５２＿２においてＣＵＲ＿ＲＨＹがインクリメントされ、次いで、リズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭ（図２０参照）が参照されそのリズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭに応じたリズムパターンテーブルのステップ数ＲＨＹ＿ＸＸＸ＿ＳＩＺＥが読み出されてＳＩＺＥに格納され（ステップ５２＿４）、ＣＵＲ＿ＲＨＹ≦ＳＩＺＥであるか否か、すなわち、ＣＵＲ＿ＲＨＹがインクリメントされた（ステップ５２＿２）結果、現在使用中のリズムパターンテーブルＲＨＹ＿ＸＸＸ＿ＴＢＬ［］のステップ数を越えたか否かが判定され、越えていないときはそのままステップ５２＿７に進み、越えたときは、ＣＵＲ＿ＲＨＹに‘０’が格納されることにより先頭のステップに戻った上で、ステップ５２＿７に進む。

【００９８】

ステップ５２＿７～５２＿１１では、スキャンモードテーブルの現在のステップＣＵＲ＿ＳＭＯＤＥ（図３１参照）が更新される。すなわち、先ずステップ５２＿７でＣＵＲ＿ＳＭＯＤＥがインクリメントされ、その後スキャンモード番号ＰＲＭ＿ＳＭＯＤＥ（図２１参照）が参照され（ステップ５２＿８）、そのスキャンモード番号ＰＲＭ＿ＭＯＤＥに対応するスキャンモードテーブルのステップ数ＳＭＯＤＥ＿ＸＸＸ＿ＳＩＺＥが読み出されてＳＩＺＥに格納され（ステップ５２＿９）、ＣＵＲ＿ＳＭＯＤＥ≧ＳＩＺＥの場合に（ステップ５２＿１０）、ＣＵＲ＿ＳＭＯＤＥが先頭（ステップ０）に戻される（ステップ５２＿１１）。

【００９９】

次に図５３に示すステップ５３＿１に進み、スキャンモード番号ＰＲＭ＿ＳＭＯＤＥ（図２１参照）が参照され、そのスキャンモード番号ＰＲＭ＿ＳＭＯＤＥに対応したスキャンモードテーブルＳＭＯＤＥ＿ＸＸＸ＿ＴＢＬ［］の、図５２のステップ５２＿７～５２＿１１で更新されたステップＣＵＲ＿ＳＭＯＤＥのスキャン方式（スキャンファンクション番号）ＳＭＯＤＥ＿ＸＸＸ＿ＴＢＬ［ＣＵＲ＿ＳＭＯＤＥ］が読み出されて、ｆｕｎｃに格納される。ステップ５３＿３，５３＿４では、それぞれ、ｆｕｎｃが和音スキャン方式ＳＣＡＮ＿Ｃであるか否か、および最低音抜き和音スキャン方式ＳＣＡＮ＿Ｃ＿ＷＯ＿Ｂであるか否かが判定され、ＳＣＡ＿ＣもしくはＳＣＡＮ＿Ｃ＿ＷＯ＿Ｂの場合は、ＣＵＲ＿ＮＯＴＥ（図２５参照）に‘－１’が格納される。ＳＣＡＮ＿ＣもしくはＳＣＡＮ＿Ｃ＿ＷＯ＿Ｂの場合、ＣＵＲ＿ＮＯＴＥに‘－１’を格納しておくことによって、ノートオンデータの走査のときに音高の低い方から高い方に向かってＰＬＡＹＢＵＦ［］の領域が走査される（図４５，図４７参照）。図５４は、エディットルーチンのフローチャートである。このエディットルーチンは、パネル（図２参照）が操作された際に、図３８に示すゼネラルプログラムのステップ３８＿５で実行される。

【０１００】

ステップ５４＿１では、リズムパターンが変更されたか否かが判定される。リズムパターンの変更は、図２に示すリズムパターン設定用操作子１５１を操作することにより行われる。リズムパターン設定用操作子１５１を操作することによりリズムパターンが変更される。ステップ５４＿２において、ＰＲＭ＿ＲＨＹＴＨＭ（図２０参照）に、変更後の新たなリズムパターン番号が格納され、ステップ５２＿３に進み、リセットスキャナルーチン（図４０参照）が実行されてスキャナのリセットが行われる。

【０１０１】

ステップ５４＿４では、スキャンモードが変更されたか否かが判定される。スキャンモードの変更は、図２に示すスキャンモード設定用操作子１５２を操作することにより行われる。スキャンモード設定用操作子１５２が操作されることによりスキャンモードが変更されると、ステップ５４＿５において、ＰＲＭ＿ＳＭＤＥ（図２１参照）に変更後の新たなスキャンモード番号が格納され、ステップ５２＿３でスキャナのリセットが行われる。

【０１０２】

ステップ５４＿６では、グルーブレートが変更されたか否かが判定される。グルーブレートの変更は、図２に示すグルーブレート設定用操作子１５３の操作により行われる。グルーブレート設定用操作子１５３の操作によりグルーブレートが変更されると、ステップ５４＿７に進み、ＰＲＭ＿ＧＲＯＯＶＥ（図２２参照）にグルーブレートの新たな値が格納される。

【０１０３】

図５５は、ノートオン処理ルーチンのフローチャートである。このノートオン処理分は、ノートオンデータを受信した際に、図３８に示すゼネラルブロックプログラムのステップ３８＿７で実行される。ステップ５５＿１では、受信したノートオンデータのうちのノートナンバがＮＴに格納され、ベロシティがＶＬに格納される。このベロシティＶＬは、ＮＯＴＥＢＵＦ［ＮＴ］（図２３参照）およびＰＬＡＹＢＵＦ［ＮＴ］（図２４参照）に格納される（ステップ５５＿２，５５＿３）。次に、ＯＲＤＥＲ＿ＷＲ（図２８参照）が１６未端か否か判定され、１６未満のときは、押鍵順バッファＯＲＤＥＲ［］（図２７参照）に空きがあるため、ＯＲＤＥＲ［ＯＲＤＥＲ＿ＷＲ］．ｎｏｔｅにノートナンバＮＴが格納され、ＯＲＤＥＲ［ＯＲＤＥＲ＿ＷＲ］．ｖｅｌｏにベロシティＶＬが格納される。ステップ５５＿７では、今回入力されたノートオンデータが全て離鍵（ノートオフ）されていた状態から最初に受信したノートオンデータであるか否かが判定される。すなわち、具体的には、ＰＬＡＹＢＵＦ［］に格納されている有効なベロシティデータは、ステップ５５＿３で今回格納したベロシティ１個のみであるか否かが検出される。最初のノートオンデータであった場合はステップ５５＿８に進み、スキャナのリセットが実行される。その後、ＰＬＡＹＢＵＦ［］が走査されて、最低音ＬＯ，最高音ＨＩが更新される（ステップ５５＿９，５５＿１０）。

【０１０４】

図５６は、ノートオフ処理ルーチンのフローチャートである。このノートオフ処理ルーチンは、ノートオフデータを受信した際に、図３８に示すゼネラルプログラムのステップ３８＿９で実行される。ステップ５６＿１では、受信したノートオフデータのノートナンバがＮＴに格納され、ＮＯＴＥＢＵＦ［ＮＴ］（図２３参照）に、有効なデータが格納されていないことを表わす‘－１’が格納される（ステップ５６＿２）。ステップ５６＿３では、ＨＯＬＤ（図２６参照）が６４未満（ホールドオフ）か否かが判定され、ホールドオン（６４以上）のときはそのまま終了する。ホールドオフ（６４未満）のときは、ＰＬＡＹＢＵＦ［ＮＴ］（図２４参照）に‘－１’が格納され（ステップ５６＿４）、押鍵順バッファＯＲＤＥＲ［］（図２７参照）から今回のノートオフデータに対応するノートオンデータを削除するデリートオーダルーチンが実行され（ステップ５６＿５）、さらにデリートオーダルーチンにより削除が行われた押鍵順バッファＯＲＤＥＲ［］を前詰めにするパックオーダルーチンが実行される（ステップ５６＿６）。ステップ５６＿７，５６＿８ではＰＬＡＹＢＵＦ［］がサーチされて最低音ＬＯ，最高音ＨＩが更新される。

【０１０５】

図５７は、デリートオーダルーチンのフローチャートである。このデリートオーダルーチンは、図５６に示すノートオフ処理ルーチンのステップ５６＿５で実行される。ここではＯＲＤＥＲ［］（図２７参照）がサーチされ、ノートオフデータのノートナンバＮＴの存在がサーチされ（ステップ５７＿１～５７＿５）、ＯＲＤＥＲ［ｉ］＝ＮＴの場合、ＯＲＤＥＲ［ｉ］．ｎｏｔｅおよびＯＲＤＥＲ［ｉ］．ｖｅｌｏの双方に－１が格納される。複数のＯＲＤＥＲ［ｉ］にＮＴが格納されていた場合、その全てについて‘－１’が格納される。

【０１０６】

図５８は、パックオーダルーチンのフローチャートである。このパックオーダルーチンは、図５６に示すノートオフ処理ルーチンのステップ５６＿６で実行される。ここでは、押鍵順バッファＯＲＤＥＲ［］がサーチされ、‘－１’が格納されている領域をなくすように前詰めされる。具体的には、ステップ５８＿１において、ｉ，ｊ双方に初期値‘０’が格納され、ステップ５８＿２においてＯＲＤＥＲ［ｉ］．ｎｏｔｅに‘－１’が格納され

ているか否かが調べられる。ＯＲＤＥＲ［ｉ］．ｎｏｔｅに有効なデータ（すなわち‘－１’以外）が格納されていたときは、ステップ５８＿３に進み、ＯＲＤＥＲ［ｉ］．ｎｏｔｅ，ＯＲＤＥＲ［ｉ］．ｖｅｌｏが、それぞれ、ＯＲＤＥＲ［ｊ］．ｎｏｔｅ，ＯＲＤＥＲ［ｊ］．ｖｅｌｏに転記され、ステップ５８＿４でｊがインクリメントされる。ＯＲＤＥＲ［］の前詰めに伴って、押鍵順アルペジオ演奏における現在のステップＣＵＲ＿ＯＲＤＥＲ（図２９参照）も変更する必要があることから、ステップ５８＿５ではＣＵＲ＿ＯＲＤＥＲがｉか否か調べられ、ＣＵＲ＿ＯＲＤＥＲ＝ｉのときは、ＣＵＲ＿ＯＲＤＥＲにｊ－１が格納される（ステップ５８＿６）．ステップ５８＿７ではｉがインクリメントされ、ステップ５８＿８でｉがＯＲＤＥＲ＿ＷＲ（図２８参照）以下か否か判定され、ｉ≦ＯＲＤＥＲ＿ＷＲの場合ステップ５８＿２に戻って、更新されたｉについてのＯＲＤＥＲ［ｉ］．ｎｏｔｅが‘－１’か否か調べられる。

【０１０７】

ＯＲＤＥＲ［］のポインタＯＲＤＥＲ＿ＷＲ（図２８参照）もＯＲＤＥＲ［］の前詰めに伴って変更する必要があり、ステップ５８＿９では、ＯＲＤＥＲ＿ＷＲにｊが格納される。図５９は、ホールドオフ処理ルーチンのフローチャートである。このホールドオフ処理ルーチンはホールドデータを受信した際に、図３８に示すゼネラルプログラムのステップ３８＿１３で実行される。このホールドオフ処理ルーチンが実行されるときは、図３８のステップ３８＿１２の判定により、ＨＯＬＤが６４未満（ホールドオフ）であることがわかっている。

【０１０８】

図５９に示すホールドオフ処理ルーチンが実行されると、押鍵，離鍵をそのまま反映したＮＯＴＥＢＵＦ［］（図２３参照）が、その全域にわたって、ホールド情報を加味した押鍵情報を格納するＰＬＡＹＢＵＦ［］（図２４参照）に転記される（ステップ５９＿１～５９＿４）。次いで、押鍵順バッファＯＲＤＥＲ［］（図２７参照）の再構築を行うリメークオーダルーチンが実行される（ステップ５９＿５）。ＯＲＤＥＲ［］が再構築された後、ステップ５９＿６，５９＿７では、ＰＬＡＹＢＵＦ［］がサーチされて最低音ＬＯ，最高音ＨＩの更新が行われる。

【０１０９】

図６０は、リメークオーダルーチンのフローチャートである。このリメークオーダルーチンは図５９に示すホールドオフ処理ルーチンのステップ５９＿５で実行される。ここでは、ＰＬＡＹＢＵＦ［］を音高の低い方から高い方と順にサーチするために、ＮＴに、先ず初期値‘０’が格納される（ステップ６０＿１）。

【０１１０】

次いで、ＰＬＡＹＢＵＦ［ＮＴ］が読み出されてＶＬに格納され（ステップ６０＿２）、そのＶＬが‘－１’か否か、すなわちＰＬＡＹＢＵＦ［ＮＴ］に有効なベロシティデータが格納されていた（－１以外）か否（－１）かが判定される。ＶＬ＝－１のときは、ステップ６０＿４に進み、デリートオーダルーチン（図５７参照）が実行され、このデリートオーダルーチンの中で、ＯＲＤＥＲ［］にノートナンバＮＴのノートオンデータが格納されていた場合にそれを削除する（具体的には‘－１’を格納する）。ステップ６０＿５ではＮＴがインクリメントされ、ステップ６０＿６では、ＮＴが１２８未満か否か判定され、ＮＴが１２８未満のときはステップ６０＿２に戻って新たなＮＴについてＰＬＡＹＢＵＦ［ＮＴ］が読み出される。

【０１１１】

このようにして、ＰＬＡＹＢＵＦ［］の全域がサーチされ、ＯＲＤＥＲ［］から、ＶＬ＝－１が格納されているＰＬＡＹＢＵＦ［ＮＴ］に対応するノートナンバＮＴが全て削除される。その後、ステップ６０＿７において、パックオーダルーチン（図２９参照）が実行されて、ＯＲＤＥＲ［］の‘－１’が格納された領域をなくすようにＯＲＤＥＲ［］が前詰めされる。これにより、ＯＲＤＥＲ［］は、現在押鍵中の鍵のみについて、押鍵順に押鍵情報が並んだ状態となる。

【０１１２】

本実施形態におけるアルペジエータは、以上説明したように構成されており、図２に示すリズムパターン設定用操作子１５１およびスキャンモード設定操作子１５２により設定された、リズムパターンテーブル（図３、および図４～図９の各例参照）およびスキャンモードテーブル（図１０および図１１～図１９の各例参照）を用いて、複数の鍵を同時に押鍵し、あるいは順次に押鍵して押鍵したままとし、あるいはホールドペダルを踏んで押鍵を記憶させておくという簡単な動作で、音楽的に意味の深い、種々のアルペジオ演奏を行うことができる。また、図２に示すグルーブレート操作子１５１を操作することにより、‘ノリ’の程度を簡単に変更することができる。

【０１１３】

次に、本発明のアルペジエータの他の実施形態について説明する。ただし、これ以降では、上述した実施形態と共通的な部分については図示および説明は省略し、上述の実施形態との相違点のみ、図示および説明を行う。図６１は、図２のパネルに付加されるベロシティボリウムを示した図、図６２は、図２に示すパネルに図６１に示したベロシティボリウムが付加された構成において、図４９に示すベロシティ生成ルーチンに代えて採用されるベロシティ生成ルーチンのフローチャートである。

【０１１４】

この実施形態では、図６１に示すようなベロシティボリウム１５４がパネル１５（図２参照）に付加される。このベロシティボリウム１５４は、手でつまんで回転させることができ、反時計まわりにいっぱいに回転させると‘０’、時計まわりにいっぱいに回転させると‘１２７’、それらの中間位置ではその位置に応じた値が出力され、ＲＡＭ１３（図１参照）中のＶＥＬＯＣＩＴＹ＿ＶＯＬＵＭＥに格納される。このＶＥＬＯＣＩＴＹ＿ＶＯＬＵＭＥは、図６２のフローチャートで説明するように、アルペジオ演奏におけるベロシティデータの生成にかかわっている。

【０１１５】

図６２に示すベロシティ生成ルーチンでは、図４９のベロシティ生成ルーチンと同様に、リズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭが参照され（ステップ６２＿１）、そのリズムパターン番号ＰＲＭ＿ＲＨＹＴＨＭに対応するリズムパターンテーブルＰＨＹ＿ＸＸＸ＿ＴＢＬ（図３、および図４～図９の各例参照）の現在のステップＣＵＲ＿ＲＨＹ（図３８参照）のベロシティ係数ＲＨＹ＿ＸＸＸ＿ＴＢＬ［ＣＵＲ＿ＲＨＹ］．ｖｅｌｏＣｏｅｆが読み出されて、ＣＯＥＦに格納される（ステップ６２＿２）。

【０１１６】

その後、図４９に示すベロシティ生成ルーチンと異なり、ＶＥＬＯＣＩＴＹ＿ＶＯＬＵＭＥ＝０か否かが判定される（ステップ６２＿３）。ＶＥＬＯＣＩＴＹ＿ＶＯＬＵＭＥ＝０の場合は、図４９のベロシティ生成ルーチンと同様であり、ステップ６２＿５に進んで、ベロシティ係数ＣＯＥＦと、押鍵により生成されたベロシティＶＬとの間で、ＶＬ２＝ＶＬ×ＣＯＥＦ／１２７の計算が行われ、さらにステップ６２＿６において、ＶＬ－（ＶＬ－ＶＬ２）×ＰＲＭ＿ＧＲＯＯＶＥ／１００の計算が行われて、その計算の結果求められたベロシティが再度ＶＬに格納される。

【０１１７】

一方、ステップ６２＿３において、ＶＥＬＯＣＩＴＹ＿ＶＯＬＵＭＥが‘０’ではないと判定されると、ステップ６２＿４に進み、そのＶＥＬＯＣＩＴＹ＿ＶＯＬＵＭＥがＶＬに格納される。すなわち、この場合は、押鍵により生成されたベロシティは無視され、その押鍵により生成されたベロシティに代わり、図６１に示すベロシティボリウム１５４により設定されたＶＥＬＯＣＩＴＹ＿ＶＯＬＵＭＥが採用される。

【０１１８】

このように、押鍵ベロシティの代わりに固定のベロシティを用いるようにすると、押鍵強さに関係なく、一定の強さを持つアルペジオ演奏が実現できる。すなわち、演奏状態に左右されず安定した再現性のあるアルペジオ演奏を行うことができる。図６３は、図２のパネルに付加されるスタイルスイッチを示した図、図６４は、図２に示すパネルに図６３に示したスタイルスイッチが付加された構成において、図５４に示すエディットルーチンに

代えて採用されるエディットルーチンのフローチャートである。

【０１１９】

図６３に示すスタイルスイッチ１５５は、ジャンル毎に複数（この例では３個）のボタン１５５ａ，１５５ｂ，１５５ｃから成り、各ボタン１５５ａ，１５５ｂ，１５５ｃを押すと、それぞれワルツ（ＷＡＬＴＺ）、レゲエ（ＲＥＧＧＡＥ）、三味線（ＳＨＡＭＩＳＥＮ）のイメージに合致した、リズムパターン番号とスキャンモード番号とのセットが選択されるようになっている。

【０１２０】

図６４に示すエディットルーチンでは、先ず、ステップ６４＿１において、スタイルスイッチ１５５（図６３参照）のいずれかのボタン１５５ａ，１５５ｂ，１５５ｃが押されることによりスタイルが変更されたか否かが判定される。スタイルが変更されたことが認識されると、その変更後のスタイルに応じて（ステップ６４＿２）、ワルツの場合はＰＲＭ＿ＲＨＹＴＨＭ（図２０参照）、ＰＲＭ＿ＳＭＯＤＥ（図２１参照）にそれぞれ‘２’（図６参照）、‘６’（図１７参照）が格納され、レゲエの場合は、ＰＲＭ＿ＲＨＹＴＨＭ，ＰＲＭ＿ＳＭＯＤＥにそれぞれ‘５’（図９参照）、‘７’（図１８参照）が格納され、三味線の場合はＰＲＭ＿ＲＨＹＴＨＭ，ＰＲＭ＿ＳＭＯＤＥにそれぞれ‘１’（図５参照）、‘８’（図１９参照）が格納される。図６４のエディットルーチンの他のステップ６４＿４～６４＿１０は、図５４に示すエディットルーチンのステップ５４＿１～５４＿７と同様であるため、説明は省略する。

【０１２１】

このように、本実施形態では、例えばワルツのボタン１５４ａを押したときには、ワルツに最もふさわしいと一般的に考えられるリズムパターンとスキャンモードが同時に選択されて設定される。したがって演奏者は、リズムパターンとスキャンモードとの組合せをいちいち考えることなく、自分の選びたいスタイルをワンタッチで選ぶことができる。

【０１２２】

なお、上記実施形態ではアルペジオ演奏に使用する音色に関しては特に触れなかったが、選択されたスタイルに対応してそのスタイルの演奏に適した音色を自動的に選択するようにしてもよい。例えば、ワルツ、レゲエのスタイルが選択された場合にはワルツ、レゲエの楽曲を演奏する際によく用いられる楽器の音色をそれぞれ選択し、三味線のスタイルが選択された場合には三味線の音色を選択するようにしてもよい。尚、上記各実施形態では、ホールド情報はデータ入力端子１７（図１参照）を経由して入力されるが、パネル１５にホールドオン情報とホールドオフ情報とを切換え入力するスイッチを設けてもよく、あるいはアルペジエータ自体にホールド情報生成用ペダル、ないしそのペダル接続用のジャックを備えてもよい。また、上記各実施形態では、鍵盤および音源が外部に接続されているが、鍵盤もしくは音源のうちの一方もしくは双方を内蔵し、あるいは一体的に構成してもよく、それに加え、さらに外部と送受信できるようにしてもよい。

【０１２３】

また、上記各実施形態では、グルーブレート設定用操作子１５３を１個のみ備え、その１個のグルーブレート設定用操作子１５３で設定したグルーブレートＰＲＭ＿ＧＲＯＯＶＥによってアルペジオ演奏の各ステップのデュレーションとベロシティとの双方を変更しているが、それらの双方を変更できる構成になっている必要はなく、それらのうちの一方のみ変更できても、音楽的に十分意味のある変更が可能である。あるいは、より微妙な調整を演奏者に委ねるために、その１個のグルーブレート設定用操作子１５３を、デュレーション変更用とベロシティ変更用とに切換えて操作できるように構成したり、デュレーション変更用とベロシティ変更用とに役割が分担された２つの操作子を備えてもよい。

【０１２４】

また、上記各実施形態では、リズムパターンテーブルの各ステップに、ステップタイム、デュレーション、ベロシティ係数の三者がセットになって記録されているが、ステップタイムのみ、デュレーションのみ、あるいはベロシティのみのいずれかのリズムパターンテーブルを持ち、他の二者は固定値を用いるようにしてもよい。リズムパターンテーブルと

して、デュレーションのみのテーブルを持った場合であっても、時間的に等分された単純なアルペジオ演奏だけでなくさまざまなリズムのアルペジオ演奏が可能となり、ステップタイムのみのテーブルを持った場合であっても、スタッカートやテヌートの効果を、アルペジオ演奏の各ステップ毎に独立して付加できるので、リズムの多彩なニュアンスやアーティキュレーションを付けることができ、ベロシティ係数のみのテーブルを持った場合であっても、アルペジオ演奏の各ステップ毎に強弱を付けることができさまざまなアクセントのアルペジオ演奏が実現するなど、それぞれ音楽的に意味のあるアルペジオ演奏を実現できる。もちろん、上記三者のうちのいずれか二者のリズムパターンテーブルを持ってもよい。

【０１２５】

また、リズムパターンテーブルに関し、上記二者ないし三者のリズムパターンテーブルを持つにあたり、それら二者ないし三者のリズムパターンテーブルをそれぞれ独立して持ち（例えばステップタイムのみのテーブルとデュレーションのみのテーブルとベロシティのみのテーブルとに分けて持ち）、演奏者の好みや、演奏スタイルに応じてそれらを組合せて用いるようにしてもよい。

【０１２６】

なお、リズムパターンテーブルに記憶するデータの種類としてステップタイム、デュレーション、ベロシティー係数以外の種類のデータを記憶するようにしてもよい。例えば、各ステップ毎に音色指定データを記憶すれば各ステップ毎に音色を切り換えることができ、より一層高度なアルペジオ演奏が可能となる。この場合、この音色が、本発明にいう楽音の特質として観念される。さらに、上記各実施形態においては、リズムパターンテーブル、およびスキャンモードテーブルはＲＯＭ１４（図１参照）に固定的に記憶されている旨説明したが、ＲＯＭ１４に代わり、あるいはＲＯＭ１４とともに、ＲＡＭ１３にリズムパターンテーブルないしスキャンモードテーブルを置いてもよい。その場合、外部から新たなリズムパターンテーブルないしスキャンモードテーブルをロードして、あるいは、パネル１５にリズムパターンテーブルないしスキャンモードテーブルを設定できるような操作子を配置しておいて演奏者がその操作子を操作することにより新たなリズムパターンテーブルないしスキャンモードテーブルを定義して、自分の好みに一層適合したアルペジオ演奏を行うことができる。

【０１２７】

【発明の効果】

以上説明したように、本発明のアルペジエータによれば、より音楽的なアルペジオ演奏を簡単な操作で行うことができる。

【図面の簡単な説明】

【図１】本発明のアルペジエータの一実施形態の回路構成図である。

【図２】パネルの構成を示した図である。

【図３】リズムパターンテーブルの基本パターンを示す図である。

【図４】４分音符用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図５】１６分音符用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図６】ワルツ用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図７】シャッフル用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図８】ディミニッシュ用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図９】レゲエ用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図１０】スキャンモードテーブルの基本パターンを示す図である。

【図１１】上昇スキャン用のスキャンモードテーブルおよびそのステップ数を示した図である。

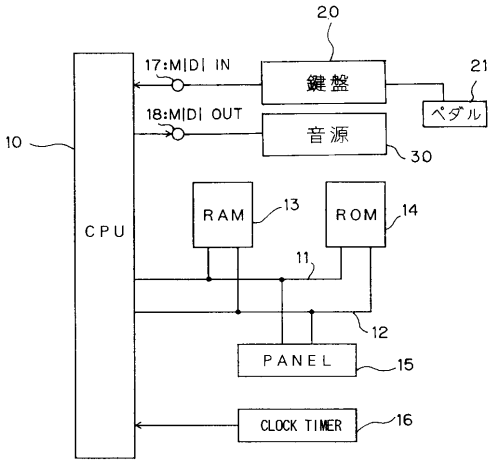
【図１２】下降スキャン用のスキャンモードテーブルおよびそのステップを示した図であ

る。
【図１３】上昇下降スキャン用のスキャンモードテーブルおよびそのステップを示した図である。
【図１４】ランダムスキャン用のスキャンモードテーブルおよびそのステップを示した図である。
【図１５】和音スキャン用のスキャンモードテーブルおよびそのステップを示した図である。
【図１６】押鍵順スキャン用のスキャンモードテーブルおよびそのステップを示した図である。
【図１７】ワルツ用のスキャンモードテーブルおよびそのステップを示した図である。
【図１８】レゲエ用のスキャンモードテーブルおよびそのステップを示した図である。
【図１９】三味線用のスキャンモードテーブルおよびそのステップを示した図である。
【図２０】リズムパターンパラメータを示す図である。
【図２１】スキャンモードパラメータを示す図である。
【図２２】グルーブレートパラメータを示す図である。
【図２３】ノートバッファを示す図である。
【図２４】プレイバッファを示す図である。
【図２５】カレントプレイノートナンババッファを示す図である。
【図２６】ホールドバッファを示す図である。
【図２７】オーダバッファを示す図である。
【図２８】オーダライトカウンタを示す図である。
【図２９】オーダポジションカウンタを示す図である。
【図３０】リズムパターンテーブルポジションカウンタを示す図である。
【図３１】スキャンモードテーブルポジションカウンタを示す図である。
【図３２】クロックカウンタを示す図である。
【図３３】ネクストクロックバッファを示す図である。
【図３４】ノートオフリザベーションバッファを示す図である。
【図３５】エンドフラグを示す図である。
【図３６】ベースノートバッファを示す図である。
【図３７】トップノートバッファを示す図である。
【図３８】ゼネラルプログラムのフローチャートである。
【図３９】初期化処理ルーチンのフローチャートである。
【図４０】リセットスキャナルーチンのフローチャートである。
【図４１】クロック処理ルーチンのフローチャートである。
【図４２】スキャンノートオンルーチンのフローチャートである。
【図４３】スキャンアップルーチンのフローチャートである。
【図４４】スキャンオーダルーチンのフローチャートである。
【図４５】スキャンコードルーチンのプログラムである。
【図４６】スキャンベースルーチンのプログラムである。
【図４７】スキャンコードウィズアウトベースルーチンのプログラムである。
【図４８】ノートオフ予約ルーチンのプログラムである。
【図４９】ベロシティ生成ルーチンのフローチャートである。
【図５０】図４９に示すベロシティ生成ルーチンで生成されるベロシティの説明図である。
【図５１】ネクストクロック更新ルーチンのフローチャートである。
【図５２】スキャナ更新ルーチンのフローチャートの前半部分である。
【図５３】スキャナ更新ルーチンのフローチャートの後半部分である。
【図５４】エディットルーチンのフローチャートである。
【図５５】ノートオン処理ルーチンのフローチャートである。
【図５６】ノートオフ処理ルーチンのフローチャートである。

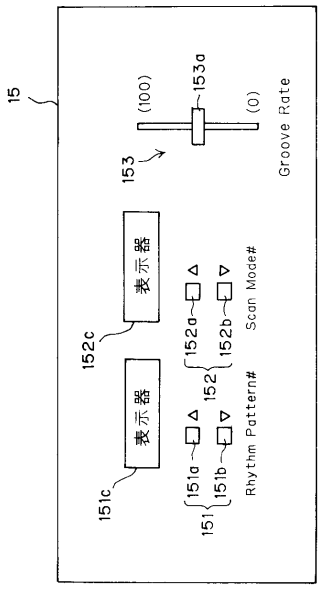
【図５７】デリートオーダルーチンのフローチャートである。
【図５８】パックオーダルーチンのフローチャートである。
【図５９】ホールドオフ処理ルーチンのフローチャートである。
【図６０】リメークオーダルーチンのフローチャートである。
【図６１】図２のパネルに付加されるベロシティボリウムを示した図である。
【図６２】図４９に示すベロシティ生成ルーチンに代えて採用されるベロシティ生成ルーチンのフローチャートである。
【図６３】図２のパネルに付加されるスタイルスイッチを示した図である。
【図６４】図５４に示すエディットルーチンに代えて採用されるエディットルーチンのフローチャートである。

【符号の説明】
１０　ＣＰＵ
１１　アドレスバス
１２　データバス
１３　ＲＡＭ
１４　ＲＯＭ
１５　パネル
１６　クロックタイマ
１７　データ入力端子
１８　データ出力端子
２０　鍵盤
２１　ペダル
３０　音源
１５１　リズムパターン設定用操作子
１５２　スキャンモード設定用操作子
１５３　グルーブレート設定用操作子
１５４　ベロシティボリウム
１５５　スタイルスイッチ

【図１】



【図２】



【図３】

[Rhythm Pattern Table]

RHY_XXX_TBL[n]

Step Time	Duration	Velo. Coef.
Step Time	Duration	Velo. Coef.
Step Time	Duration	Velo. Coef.
Step Time	Duration	Velo. Coef.
:	:	:
Step Time	Duration	Velo. Coef.

RHY_XXX_SIZE

Step Size

【図４】

RHY_4_TBL

96	92	127

RHY_4_SIZE

1

【図５】

RHY_16_TBL

24	22	127
24	22	100
24	22	100
24	22	100

RHY_16_SIZE

4

【図６】

RHY_WALTZ_TBL

96	20	127
96	20	100
96	20	100

RHY_WALTZ_SIZE

3

【図７】

RHY_SHUFFLE_TBL

32	30	127
16	14	80
32	30	100
16	14	80

RHY_SHUFFLE_SIZE

4

【図８】

RHY_DIM_TBL

24	22	127
24	22	110
24	22	95
24	22	80
24	22	60
24	22	40
24	22	30
24	22	20

RHY_DIM_SIZE

8

【図９】

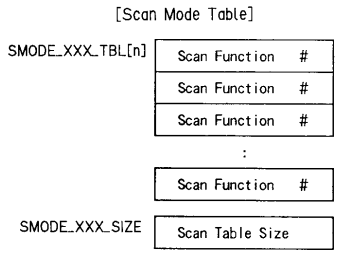
RHY_REGGAE_TBL

94	1	85
68	8	127
30	4	96
94	10	80
98	6	117

RHY_REGGAE_SIZE

5

【図１０】



【図１１】

SMODE_UP_TBL

SCAN_U

SMODE_UP_SIZE

1

【図１２】

SMODE_DOWN_TBL

SCAN_D

SMODE_DOWN_SIZE

1

【図１３】

SMODE_UP_DW_TBL

SCAN_UD

SMODE_UP_DW_SIZE

1

【図１４】

SMODE_RANDOM_TBL

SCAN_R

SMODE_RANDOM_SIZE

1

【図１５】

SMODE_CHORD_TBL

SCAN_C

SMODE_CHORD_SIZE

1

【図１６】

SMODE_ORDER_TBL

SCAN_O

SMODE_ORDER_SIZE

1

【図１７】

SMODE_WALTZ_TBL

SCAN_B
SCAN_C_WO_B
SCAN_C_WO_B

SMODE_WALTZ_SIZE

3

【図１８】

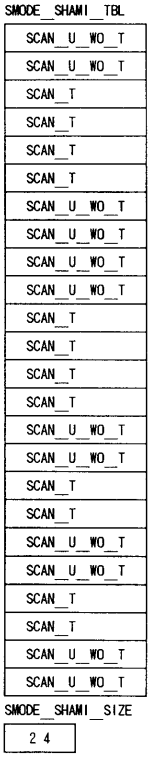
SMODE_REGGAE_TBL

SCAN_B
SCAN_C_WO_B
SCAN_C_WO_B
SCAN_B
SCAN_C_WO_B

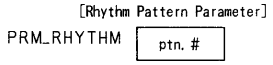
SMODE_REGGAE_SIZE

5

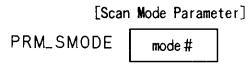
【図１９】



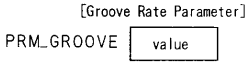
【図２０】



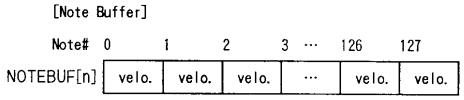
【図２１】



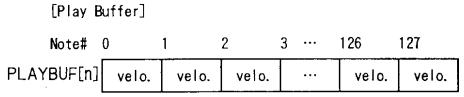
【図２２】



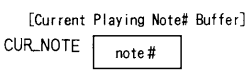
【図２３】



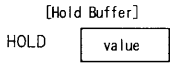
【図２４】



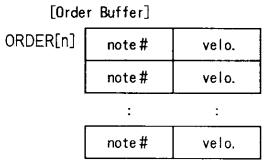
【図２５】



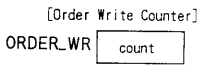
【図２６】



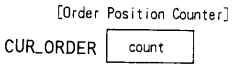
【図２７】



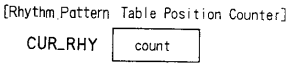
【図２８】



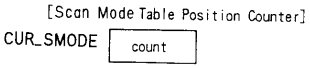
【図２９】



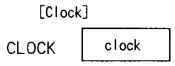
【図３０】



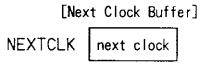
【図３１】



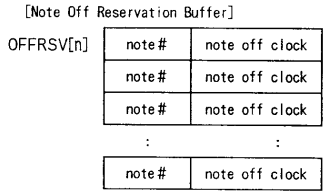
【図３２】



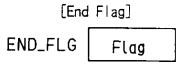
【図３３】



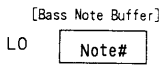
【図３４】



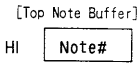
【図３５】



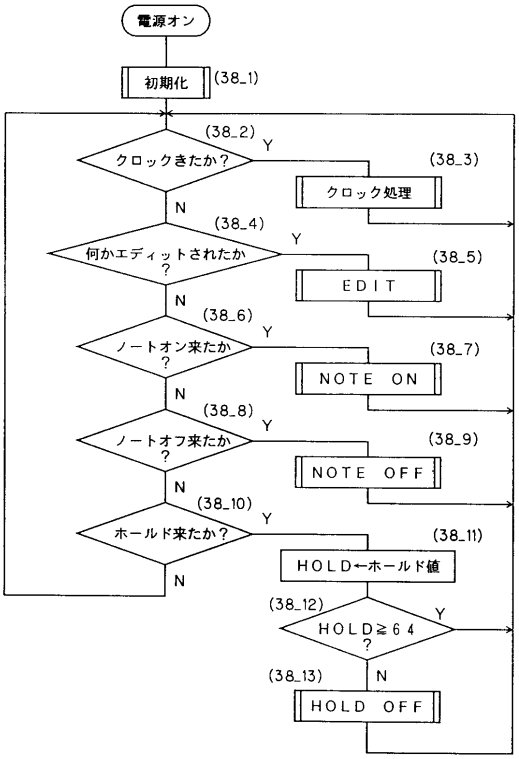
【図３６】



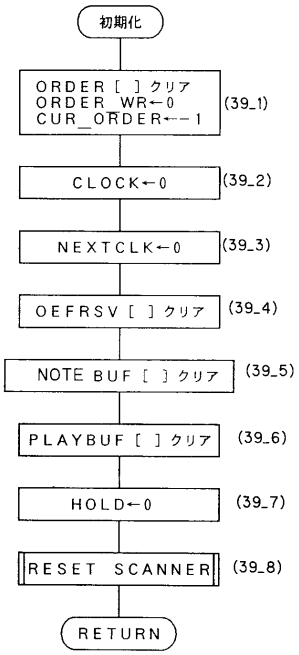
【図３７】



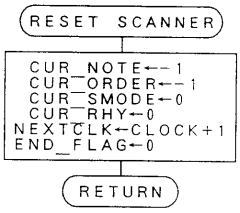
【図３８】



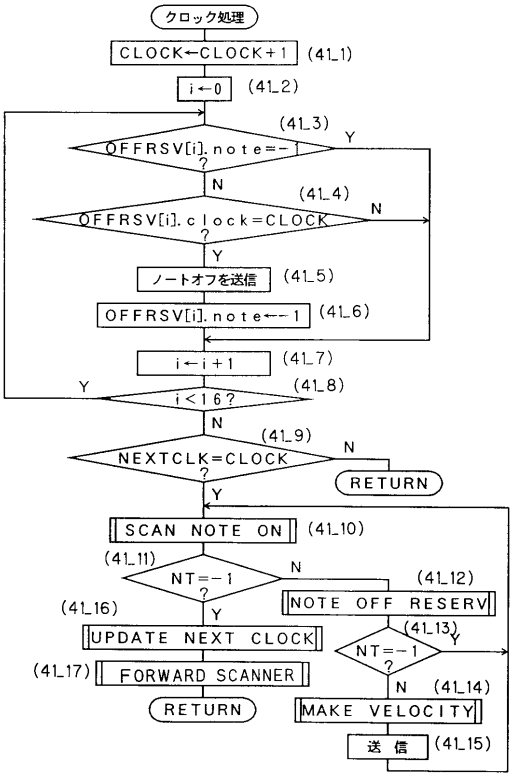
【図３９】



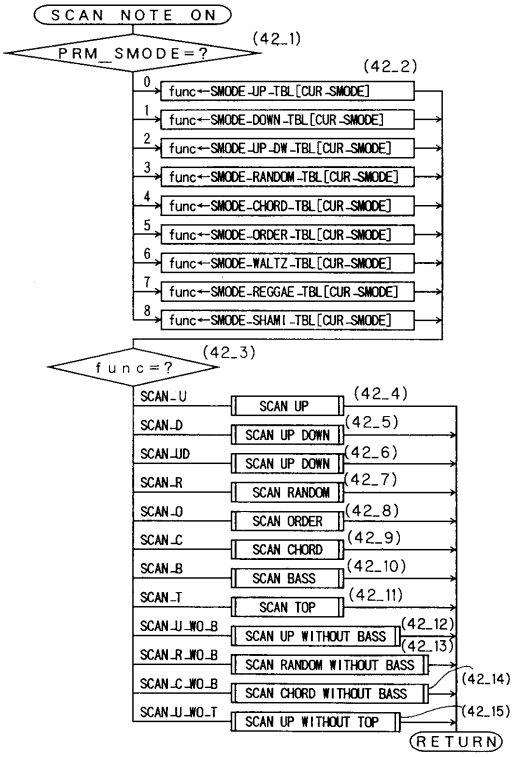
【図４０】



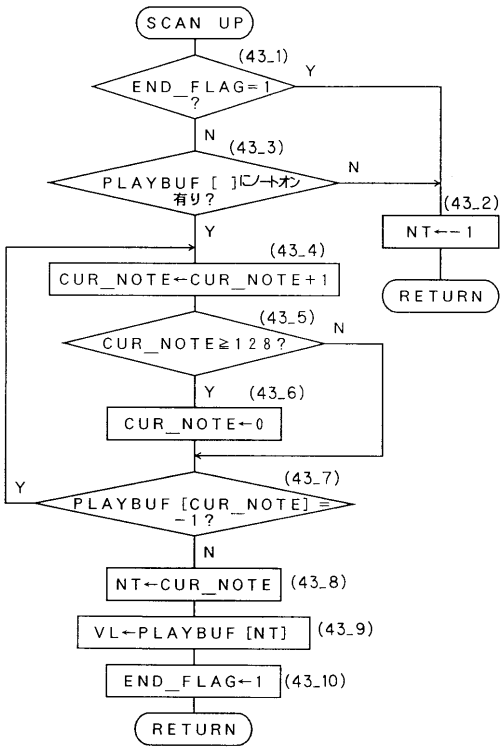
【図４１】



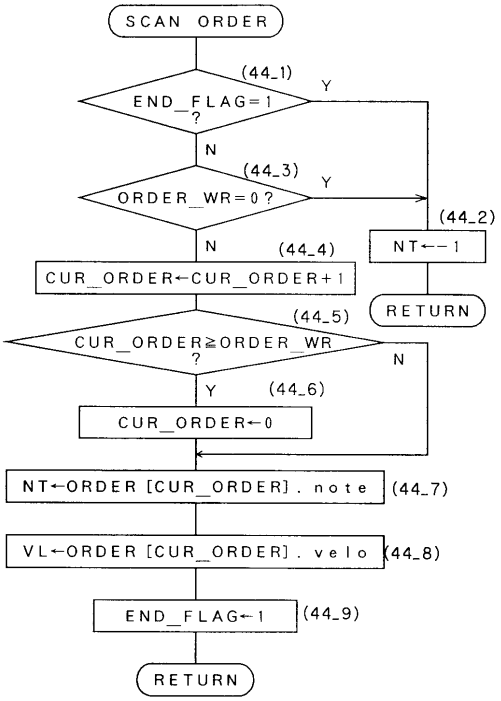
【図４２】



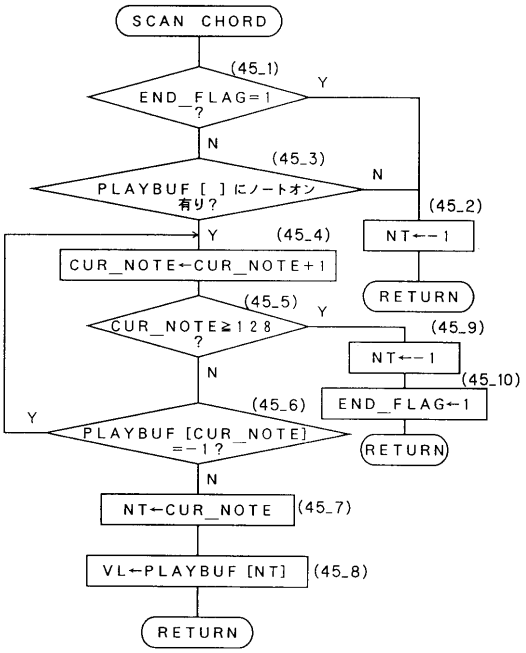
【図４３】



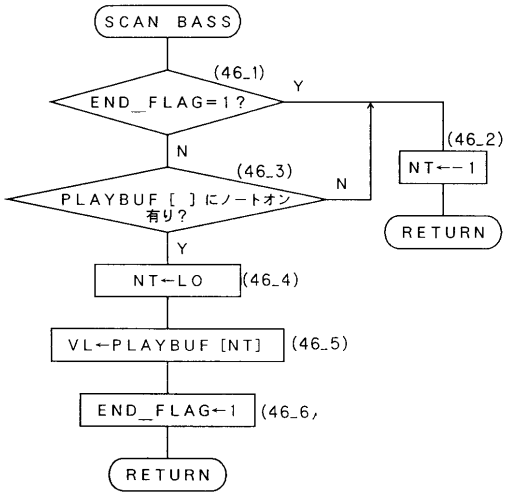
【図４４】



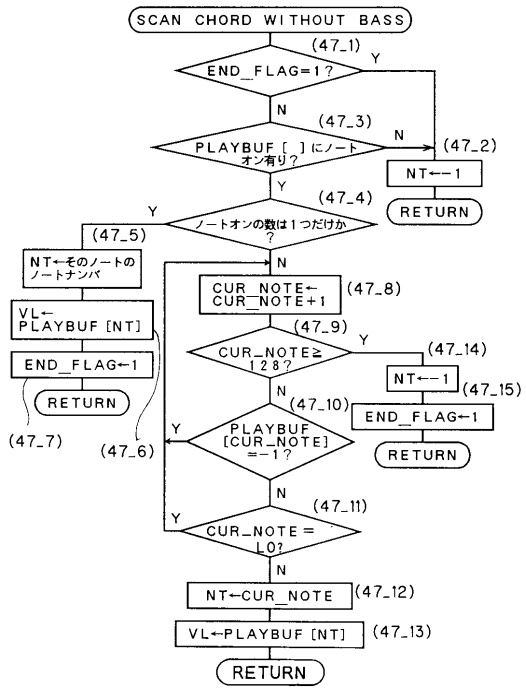
【図４５】



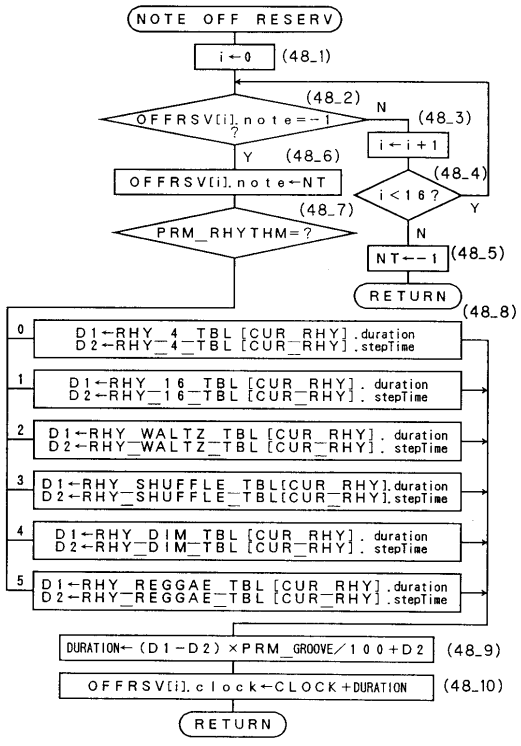
【図４６】



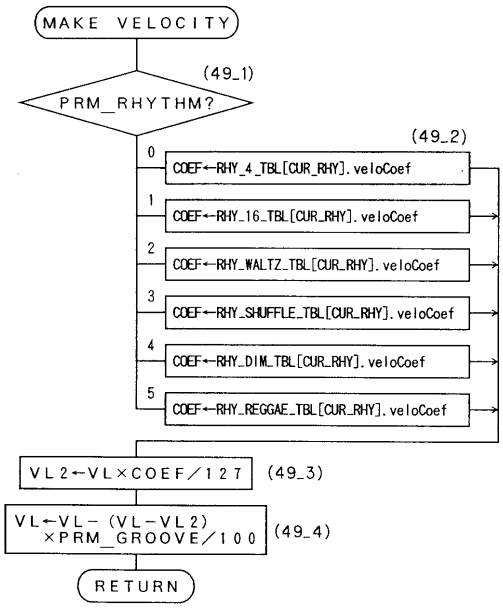
【図４７】



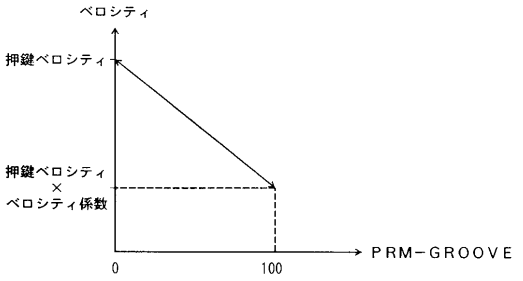
【図４８】



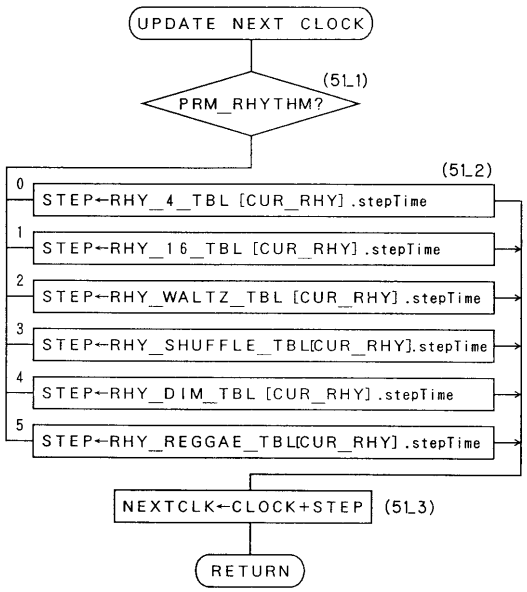
【図４９】



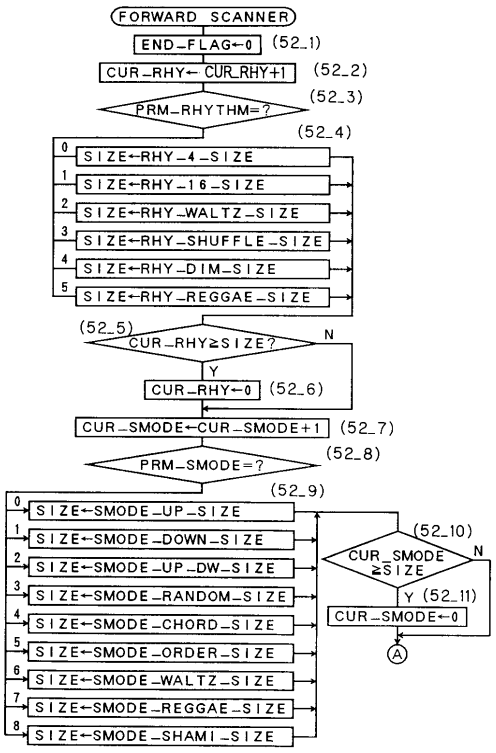
【図５０】



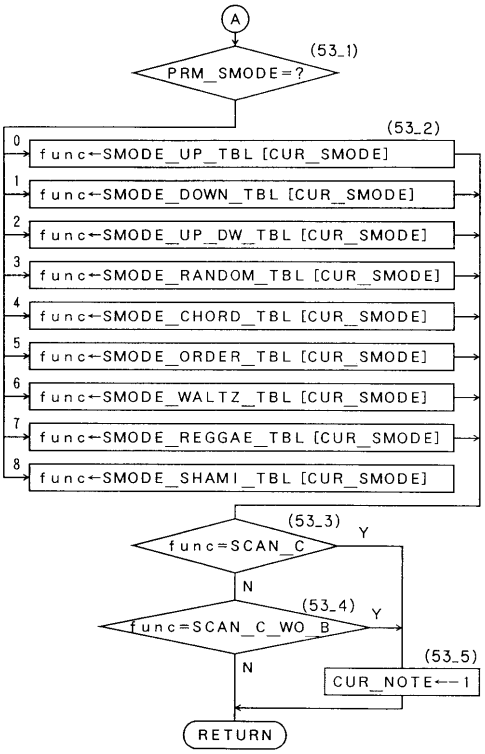
【図５１】



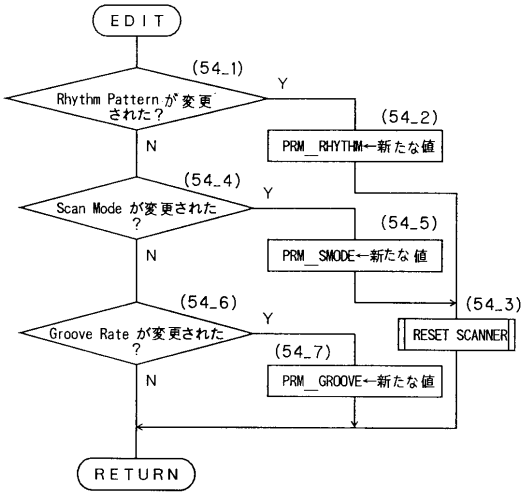
【図５２】



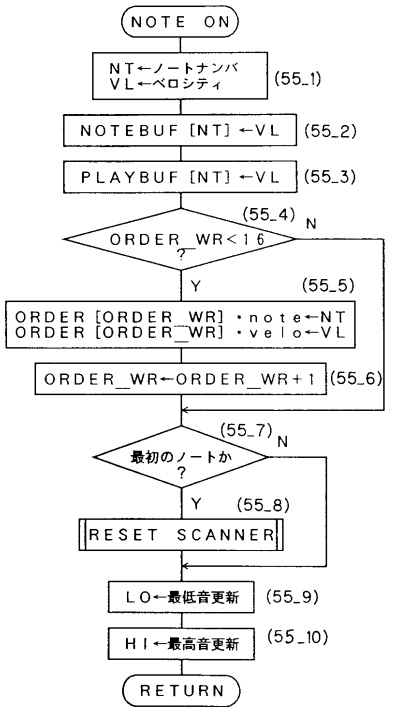
【図５３】



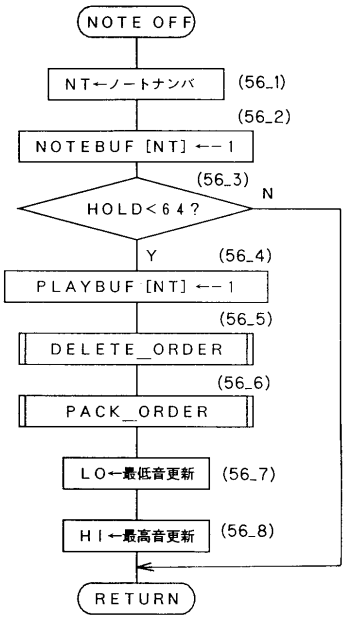
【図５４】



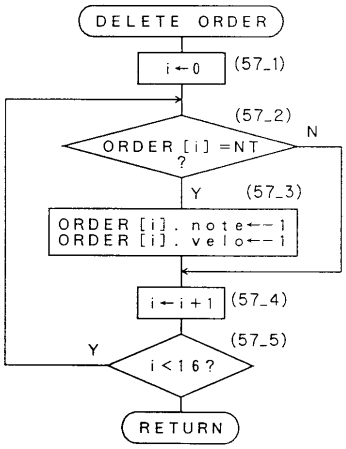
【図５５】



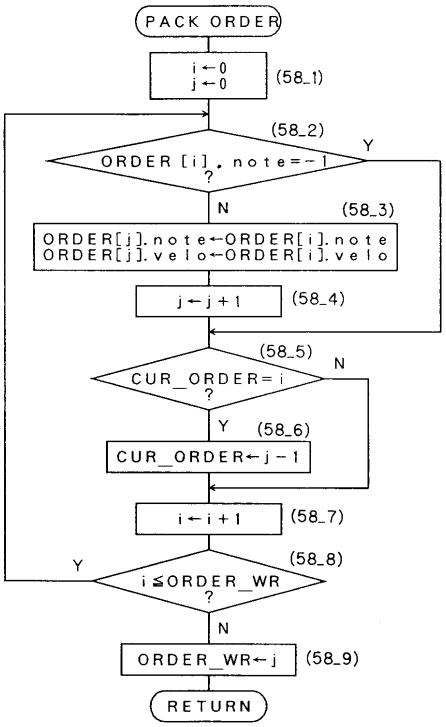
【図５６】



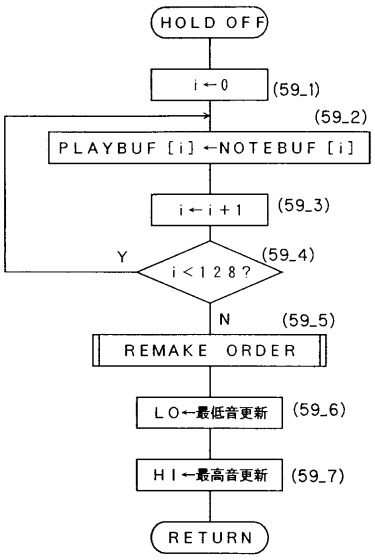
【図５７】



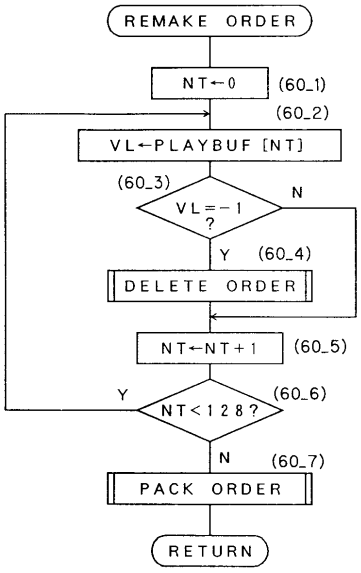
【図５８】



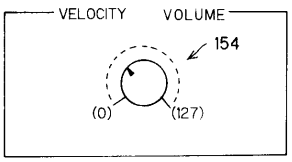
【図５９】



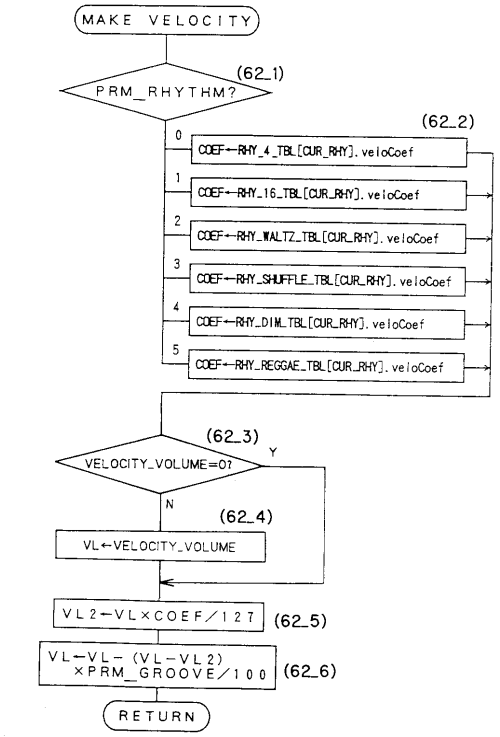
【図６０】



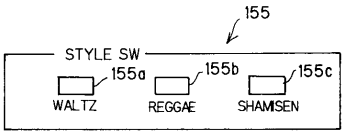
【図６１】



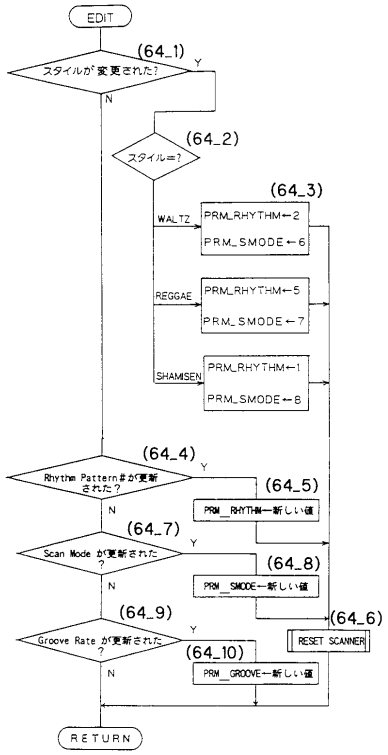
【図６２】


MAKE VELOCITY
PRM_RHYTHM?
(62_1)
(62_2)
0
COEF←RHY_4_TBL[CUR_RHY].veloCoef
1
COEF←RHY_16_TBL[CUR_RHY].veloCoef
2
COEF←RHY_WALTZ_TBL[CUR_RHY].veloCoef
3
COEF←RHY_SHUFFLE_TBL[CUR_RHY].veloCoef
4
COEF←RHY_DIM_TBL[CUR_RHY].veloCoef
5
COEF←RHY_REGGAE_TBL[CUR_RHY].veloCoef
(62_3)
VELOCITY_VOLUME=0?
Y
N
(62_4)
VL←VELOCITY_VOLUME
VL2←VL×COEF／127
(62_5)
VL←VL－(VL－VL2)
×PRM_GROOVE／100
(62_6)
RETURN

【図６３】


155
STYLE SW
155a
155b
155c
WALTZ
REGGAE
SHAMISEN

【図６４】


EDIT
(64_1)
スタイルが変更された?
Y
N
(64_2)
スタイル=?
(64_3)
WALTZ
PRM_RHYTHM←2
PRM_SMODE←6
REGGAE
PRM_RHYTHM←5
PRM_SMODE←7
SHAMISEN
PRM_RHYTHM←1
PRM_SMODE←8
(64_4)
Rhythm Pattern#が更新された?
(64_5)
PRM_RHYTHM←新しい値
(64_7)
Scan Mode が更新された?
(64_8)
PRM_SMODE←新しい値
(64_9)
Groove Rate が更新された?
(64_10)
PRM_GROOVE←新しい値
(64_6)
RESET SCANNER
RETURN

フロントページの続き

(56)参考文献 特公昭６３－０３２１９８（ＪＰ，Ｂ１）
特開昭５７－０７６５９８（ＪＰ，Ａ）
特開昭５９－０９４７９３（ＪＰ，Ａ）
特開平０５－１１３７９１（ＪＰ，Ａ）
特開平０５－２３２９３８（ＪＰ，Ａ）
特開平０５－０７３０３６（ＪＰ，Ａ）
特開昭６１－１８２０９６（ＪＰ，Ａ）
実公昭５７－００８０６４（ＪＰ，Ｙ２）

(58)調査した分野(Int.Cl.^{7}，ＤＢ名)
G10H 1/28