

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6492629号  
(P6492629)

(45) 発行日 平成31年4月3日(2019.4.3)

(24) 登録日 平成31年3月15日(2019.3.15)

(51) Int.Cl.

F 1

G 1 O H 1/18 (2006.01)  
G 1 O L 13/02 (2013.01)G 1 O H 1/18 1 O 1  
G 1 O L 13/02 1 1 O Z

請求項の数 11 (全 16 頁)

(21) 出願番号 特願2014-259483 (P2014-259483)  
 (22) 出願日 平成26年12月22日 (2014.12.22)  
 (65) 公開番号 特開2016-118721 (P2016-118721A)  
 (43) 公開日 平成28年6月30日 (2016.6.30)  
 審査請求日 平成29年12月14日 (2017.12.14)

(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  
 (72) 発明者 森山 修  
 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 安田 勇太

最終頁に続く

(54) 【発明の名称】歌唱生成装置、電子楽器、方法、およびプログラム

## (57) 【特許請求の範囲】

## 【請求項 1】

互いに異なる音高が割り当てられた複数の演奏操作子の操作状態を特定する特定部と、  
前記操作状態に基づいて、第1及び第2のいずれかの状態に移行する状態移行部と、  
前記第1の状態に移行した場合に、前記操作状態に基づいて歌詞を決定する歌詞決定部  
と、

前記第2の状態に移行した場合に、前記操作状態に基づいて音高を決定する音高決定部  
と、

前記決定した歌詞を前記決定した音高で発音させる発音データを生成する発音生成部  
<sup>、</sup>

を備えた歌唱生成装置。

## 【請求項 2】

前記操作状態は、操作された複数の前記演奏操作子の組み合わせ、操作された前記演奏操作子の数、のうちのいずれかの操作の状態を含む、

請求項1に記載の歌唱生成装置。

## 【請求項 3】

前記歌詞決定部は、前記第1の状態に移行した場合に、前記移行の判定に用いた前記操作状態に基づいて歌詞を決定し、

前記音高決定部は、前記第2の状態に移行した場合に、前記移行の判定に用いた前記操作状態に基づいて音高を決定する、

請求項 1 または 2 に記載の歌唱生成装置。

**【請求項 4】**

前記第 2 の状態に移行した場合に、前記歌詞決定部により既に決定された歌詞を、前記音高決定部により決定された音高で歌唱させるための歌唱データを生成する歌唱生成部、を更に備え、

前記発音生成部は、前記歌唱生成部にて生成された歌唱データに基づいて発音させる発音データを生成する、

請求項 3 に記載の歌唱生成装置。

**【請求項 5】**

前記歌詞決定部はさらに、前記決定された歌詞を順次メモリに記憶させ、10

前記歌唱生成部は、前記音高決定部にて音高が決定される毎に、前記メモリに記憶されている歌詞を予め記憶された順に読み出すとともに、前記読み出された歌詞を前記決定された音高で歌唱させる歌唱データを生成する、請求項 4 に記載の歌唱生成装置。

**【請求項 6】**

前記状態移行部は、予め定められた期間内に操作が行われた前記演奏操作子の数に基づいて、前記第 1 の状態及び前記第 2 の状態のいずれかの状態に移行する、

請求項 1 乃至 5 のいずれかに記載の歌唱生成装置。

**【請求項 7】**

前記状態移行部は、予め定められた期間内に前記演奏操作子が複数同時に操作された場合に前記第 1 の状態に移行し、前記予め定められた期間内に前記演奏操作子を単数操作した場合に前記第 2 の状態に移行する、請求項 6 に記載の歌唱生成装置。20

**【請求項 8】**

前記操作状態は、演奏操作子の操作が終了したタイミングを含み、

前記状態移行部は、前記操作の終了を判別する毎に、第 1 及び第 2 のいずれか一方の状態から他方の状態に移行する、

請求項 1 乃至 5 のいずれかに記載の歌唱生成装置。

**【請求項 9】**

請求項 1 乃至 8 のいずれかに記載の歌唱生成装置と、

複数の演奏操作子と、

前記歌唱生成装置からの発音データに基づいて歌唱音を発音する発音部と、30

を備えた電子楽器。

**【請求項 10】**

装置が、

互いに異なる音高が割り当てられた複数の演奏操作子の操作状態を特定する特定処理と、

前記操作状態に基づいて、第 1 及び第 2 のいずれかの状態に移行する状態移行処理と、前記第 1 の状態に移行した場合に、前記操作状態に基づいて歌詞を決定する歌詞決定処理と、

前記第 2 の状態に移行した場合に、前記操作状態に基づいて音高を決定する音高決定処理と、40

前記決定した歌詞を前記決定した音高で発音させる発音データを生成する発音生成処理と、

を実行する歌唱生成方法。

**【請求項 11】**

コンピュータに、

互いに異なる音高が割り当てられた複数の演奏操作子の操作状態を特定する特定処理と、

前記操作状態に基づいて、第 1 及び第 2 のいずれかの状態に移行する状態移行処理と、前記第 1 の状態に移行した場合に、前記操作状態に基づいて歌詞を決定する歌詞決定処理と、50

前記第2の状態に移行した場合に、前記操作状態に基づいて音高を決定する音高決定処理と、

前記決定した歌詞を前記決定した音高で発音させる発音データを生成する発音生成処理と、

を実行させるプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、演奏操作にて歌詞を指定して歌唱生成を行う歌唱生成装置、電子楽器、方法、およびプログラムに関する。 10

【背景技術】

【0002】

キー・ボードや操作パネル、鍵盤の鍵等の演奏操作子から楽曲の歌詞を予め入力し、その後モードを切り換えて再び演奏操作子の操作等で決定される音高に基づいて、予め記憶されている人の歌声の波形データの中から予め入力された歌詞に対応する波形データを読み出して歌唱生成を行うようにした電子楽器が知られている（例えば特開平6-324677号公報に記載の技術）。このような電子楽器により、人の歌声を発声させることが可能となる。

【0003】

また、片方の手で発音内容（歌詞）を、もう片方の手で発音音高を入力することにより歌唱生成を行う電子楽器も知られている。 20

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開平6-324677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、モードを切り換えて歌詞入力と音高入力を従来技術では、リアルタイムの歌詞入力を行うことができないという課題があった。 30

【0006】

また、両手を使って歌詞入力と音高入力を従来技術では、リアルタイムの歌唱入力は可能であるが、歌唱操作のみで両手が塞がってしまうため、フェーダーなど、手で操作する演奏操作子を歌唱中に操作することが困難であるという課題があった。

【0007】

そこで、本発明は、リアルタイムの歌唱入力を片手で行えるようにすることを目的とする。

【課題を解決するための手段】

【0008】

態様の一例では、互いに異なる音高が割り当てられた複数の演奏操作子の操作状態を特定する特定部と、前記操作状態に基づいて、第1及び第2のいずれかの状態に移行する状態移行部と、前記第1の状態に移行した場合に、前記操作状態に基づいて歌詞を決定する歌詞決定部と、前記第2の状態に移行した場合に、前記操作状態に基づいて音高を決定する音高決定部と、前記決定した歌詞を前記決定した音高で発音させる発音データを生成する発音生成部と、を備える。 40

【発明の効果】

【0009】

本発明によれば、リアルタイムの歌唱入力を片手で行うことが可能となる。

【図面の簡単な説明】

【0010】

【図1】歌唱生成装置の第1、第2、第3の実施形態のハードウェア構成例を示す図である。

【図2】各実施形態におけるメイン処理の例を示すフローチャートである。

【図3】第1、第2の実施形態における鍵盤処理の詳細例を示すフローチャートである。

【図4】第1、第3の実施形態における歌詞決定処理の詳細例を示すフローチャートである。

【図5】第1、第3の実施形態における歌唱生成処理の詳細例を示すフローチャートである。

【図6】第2の実施形態における歌詞決定処理の詳細例を示すフローチャートである。

【図7】第2の実施形態における歌唱生成処理の詳細例を示すフローチャートである。

10

【図8】第3の実施形態における鍵盤処理の詳細例を示すフローチャートである。

【図9】歌唱生成装置の第4の実施形態のハードウェア構成例を示す図である。

【図10】第4の実施形態における鍵盤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】

【0011】

以下、本発明を実施するための形態について図面を参照しながら詳細に説明する。まず、本発明の第1の実施形態を説明する。第1の実施形態における歌唱生成処理では、演奏者は、演奏操作子としての鍵盤上の複数(2つ以上)の鍵を、Nミリ秒未満(「同時」と見なせる短い時間内で、例えば4ミリ秒未満)のうちに押鍵する。この鍵の組み合わせにより、歌詞(ここでは、発音内容の単位、たとえば「あ」「か」などのかな一文字や、発音記号などを指す)が指定され、記憶される。この動作を、歌詞決定処理と呼ぶ。その後、演奏者は、複数鍵の押鍵からNミリ秒以上経ってから、单一の鍵を押鍵する。单一鍵の押鍵からNミリ秒経っても他の押鍵が無い場合は、その单一鍵に割り当てられた音高で歌詞の発音が指示される。この動作を、歌唱生成処理と呼ぶ。单一鍵の押鍵からNミリ秒未満の内に他の押鍵があった場合は、これは複数鍵の同時押鍵の扱いになるため、歌詞決定処理の制御となり、記憶されている歌詞が上書きされる。以上の鍵盤の鍵の制御処理を鍵盤処理と呼ぶ。

20

【0012】

図1は、歌唱生成装置の第1の実施形態(後述する第2、第3の実施形態も同じ)のハードウェア構成例を示す図である。図1に示される歌唱合成装置100は、ROM(リードオンリーメモリ：読み出し専用メモリ)102、RAM(ランダムアクセスメモリ)103、および鍵盤104が、CPU(中央演算処理装置)101に接続される構成を備える。また、CPU101には、歌唱合成ユニット105が接続され、歌唱合成ユニット105には発音ユニット106が接続される。同図に示される構成は上記歌唱合成装置100を実現できるコンピュータの一例であり、そのようなコンピュータはこの構成に限定されるものではない。

30

【0013】

ROM302は、コンピュータを制御する歌唱合成プログラムを含む各プログラムを記憶するメモリである。RAM303は、各プログラムの実行時に、ROM302に記憶されているプログラム又はデータを一時的に格納するメモリである。CPU101は、各プログラムを、ROM302からRAM303に読み出して実行することにより、当該コンピュータ全体の制御を行う。

40

【0014】

鍵盤104は、複数の鍵を備え、演奏者がピアノ式の鍵盤演奏をできるようにする演奏入力装置である。

【0015】

歌唱合成ユニット105は、CPU101から指示された歌詞に対応する波形データを内部の波形メモリから選択し、その歌詞の波形データをCPU101から指示された音高に対応する読み出し速度(サンプリング間隔)で読み出し、歌唱データを合成し出力する。

【0016】

50

発音ユニット106は、歌唱合成ユニット105が出力するデジタルの歌唱データをアナログの音響信号に変換し、特には図示しないが、内部の増幅器で増幅し、スピーカから放音する。

#### 【0017】

図2は、第1の実施形態（第2、第3、第4の実施形態でも同じ）における歌唱合成処理のメイン処理の例を示すフローチャートである。この処理は、CPU101が、ROM102に記憶されている歌唱合成処理プログラムを実行する動作である。

#### 【0018】

CPU101はまず、RAM103の内容の初期化を行った後（ステップS201）、演奏者が特には図示しない電源スイッチを操作してステップS202で演奏が終了したと判定するまで、ステップS203の鍵盤処理を繰り返し実行する。  
10

#### 【0019】

図3は、第1実施形態（第2の実施形態でも同じ）における鍵盤処理の詳細例を示すフローチャートである。

#### 【0020】

まず、CPU101は、鍵盤104をスキャンし、押鍵されている鍵の集合をRAM103上の鍵集合変数c\_keysに代入する（ステップS301）。

#### 【0021】

次に、CPU101は、今回鍵集合変数c\_keysと前回処理時の鍵集合変数の内容が保持されているRAM103の前回鍵集合変数p\_keysの差集合を演算し、その結果をRAM103上の集合変数kに代入する（ステップS302）。すなわち、前回の鍵盤処理時から今回の鍵盤処理時までに新たに押鍵された分が集合変数kに代入される。  
20

#### 【0022】

次に、CPU101は、RAM103上の状態を示す状態変数stateの値がIDLE（準備完了状態）であるかACTIVE（押鍵動作中状態）であるかを判定する（ステップS303）。

#### 【0023】

状態変数stateの初期値は、図2のステップS201においてIDLE状態になっているため、CPU101は、最初は処理をステップS304に進める。ここでは、CPU101は、集合変数kの元の数を判定する。  
30

#### 【0024】

集合変数kの元の数が0のときは前回から何も変わっていないので、CPU101は、次のステップS305をスキップし、次回の鍵盤処理のためにステップS301にて算出された今回鍵集合変数c\_keysの内容を前回鍵集合変数p\_keysに格納した後に（ステップS313）、そのまま図3のフローチャートで示される図2のステップS203の今回の鍵盤処理を終了する。

#### 【0025】

集合変数kの元の数が1以上であれば、CPU101は、押鍵開始時の時刻を保持するRAM103上の押鍵開始時刻変数t\_startに現在時刻を格納する。また、CPU101は、状態変数stateに押鍵が開始されたことを示す状態値ACTIVEを格納する。さらに、CPU101は、押鍵開始後現在までに押鍵されている鍵の情報を保持する押鍵済集合変数on\_keysに、ステップS302で今回新たに押鍵された鍵の集合が格納されている集合変数kの内容を格納する（以上、ステップS205）。その後、CPU101は、次回の鍵盤処理のためにステップS301にて算出された今回鍵集合変数c\_keysの内容を前回鍵集合変数p\_keysに格納した後に（ステップS313）、図3のフローチャートで示される図2のステップS203の今回の鍵盤処理を終了する。  
40

#### 【0026】

図3のフローチャートで示される図2のステップS203の鍵盤処理は、例えば1ミリ秒ごとに繰り返し実行されている。この結果、次回の鍵盤処理では、状態変数state  
50

の値判定はステップ S 2 0 5 により A C T I V E となっているため、 C P U 1 0 1 は、ステップ S 3 0 3 からステップ S 3 0 6 に処理を進める。ステップ S 3 0 6 では、押鍵済集合変数 o n \_ k e y s に、新たに押鍵された鍵の集合変数 k の内容が加えられる。

#### 【 0 0 2 7 】

続いて、 C P U 1 0 1 は、現在時刻から押鍵開始時刻変数 t \_ s t a r t の値を減算し、その結果が前述した N ミリ秒以上であるか N ミリ秒未満であるかを判定する（ステップ S 3 0 7）。

#### 【 0 0 2 8 】

ステップ S 3 0 7 の判定が N ミリ秒未満を示していれば、同時押鍵とみなされる。この場合、次回の鍵盤処理のためにステップ S 3 0 1 にて算出された今回鍵集合変数 c \_ k e y s の内容を前回鍵集合変数 p \_ k e y s に格納した後に（ステップ S 3 1 3）、そのまま図 3 のフローチャートで示される図 2 のステップ S 2 0 3 の今回の鍵盤処理を終了する。押鍵開始時刻変数 t \_ s t a r t が示す押鍵開始時刻から現在時刻までが N ミリ秒未満であればこの処理が繰り返し実行され、押鍵済集合変数 o n \_ k e y s に押鍵開始後現在までに押鍵されている鍵の情報が蓄積されてゆく。

#### 【 0 0 2 9 】

やがて、押鍵開始時刻から時間が経過して、ステップ S 3 0 7 の判定が N ミリ秒以上を示すと、 C P U 1 0 1 は、ステップ S 3 0 7 から S 3 0 8 に処理を進める。ステップ S 3 0 8 において、 C P U 1 0 1 は、押鍵済集合変数 o n \_ k e y s の元の数が 1 であるか 2 以上であるかを判定する。

#### 【 0 0 3 0 】

押鍵済集合変数 o n \_ k e y s の元の数が 2 以上であれば、同時に押鍵された複数の鍵によって歌詞を決定する歌詞決定処理が実行される（ステップ S 3 0 9）。この処理の後、 C P U 1 0 1 は、状態変数 s t a t e の内容を I D L E 状態に戻し、押鍵済集合変数 o n \_ k e y s の内容を空集合 とする（ステップ S 3 1 2）。そして、 C P U 1 0 1 は、次回の鍵盤処理のためにステップ S 3 0 1 にて算出された今回鍵集合変数 c \_ k e y s の内容を前回鍵集合変数 p \_ k e y s に格納した後に（ステップ S 3 1 3）、図 3 のフローチャートで示される図 2 のステップ S 2 0 3 の今回の鍵盤処理を終了する。

#### 【 0 0 3 1 】

以上のようにして、演奏者は、複数鍵の押鍵により歌詞決定処理を実行させた後、今度は前述したように単数鍵の押鍵での音高指定による歌唱生成処理を実行させる。この場合には、図 3 の鍵処理において、まず単数鍵が押鍵された状態でステップ S 3 0 4 の判定が Y E S となり、ステップ S 3 0 5 により押鍵開始時刻変数 t \_ s t a r t に現在時刻が格納され、状態変数 s t a t e の値が押鍵を示す値 A C T I V E にされ、押鍵済集合変数 o n \_ k e y s に今回押鍵された鍵の内容を示す集合変数 k の内容が格納される。その後、鍵盤処理が何回か実行される間ステップ S 3 0 7 の判定が N ミリ秒未満となり、やがて N ミリ秒以上となってステップ S 3 0 7 からステップ S 3 0 8 に移行し、単数鍵の押鍵であるためステップ S 3 0 8 の判定が「 1 」となる。

#### 【 0 0 3 2 】

この場合、 C P U 1 0 1 は、押鍵済集合変数 o n \_ k e y s の内容を再生鍵集合変数 p l a y \_ k e y s に格納し（ステップ S 3 1 0）、歌唱合成ユニット 1 0 5 に対して歌唱音の発音を指示する（ステップ S 3 1 1）。これにより、歌唱合成ユニット 1 0 5 では、新たに押鍵された単数鍵による音高で、ステップ S 3 0 9 で決定された歌詞に対応する歌唱データが生成され発音される。歌唱生成処理の後、単数鍵による押鍵状態をクリアするために、状態を示す変数 s t a t e に待ち状態を示す I D L E 状態に戻し、押鍵済集合変数 o n \_ k e y s の内容を空集合 とする（ステップ S 3 1 2）。そして、 C P U 1 0 1 は、次回の鍵盤処理のためにステップ S 3 0 1 にて算出された今回鍵集合変数 c \_ k e y s の内容を前回鍵集合変数 p \_ k e y s に格納した後に（ステップ S 3 1 3）、図 3 のフローチャートで示される図 2 のステップ S 2 0 3 の今回の鍵盤処理を終了する。

#### 【 0 0 3 3 】

10

20

30

40

50

図4は、第1の実施形態（第3の実施形態でも同じ）における図3のステップS309の歌詞決定処理の詳細例を示すフローチャートである。ここでは、CPU101は、押鍵済集合変数on\_keysの内容に基づいて、ROM102に記憶されている鍵盤・歌詞変換テーブル上の歌詞データを引き（参照し）、RAM103上の歌詞変数lyricに格納する（ステップS401）。鍵盤・歌詞変換テーブルは、例えば2鍵ずつの鍵の組合せごとに、その組合せによって「あ」「い」「う」等のどの歌詞が選択されるかを登録したデータである。

#### 【0034】

図5は、第1の実施形態（第3の実施形態でも同じ）における図3のステップS311の歌唱生成処理の詳細例を示すフローチャートである。ここでは、CPU101は、図4のステップS401で設定された歌詞変数lyricの内容と、図3のステップS310で設定された再生鍵集合変数play\_keysの内容とに基づいて、歌唱合成ユニット105に対して歌唱音の発音を指示し、歌唱合成ユニット105においては、歌唱データの出力波形Wを生成する（ステップS501）。

具体的には、CPU101は、歌詞変数lyricが示す歌詞の歌唱データの生成と、再生鍵集合変数play\_keysが示す音高での生成を指示する発音指示データを、図1の歌唱合成ユニット105に供給する。

#### 【0035】

その後、CPU101は、歌唱データの出力波形Wの発音を、歌唱合成ユニット105に対して指示する。この結果、歌唱合成ユニット105は、内部の波形データメモリに記憶している複数の歌唱データのうち、CPU101から指示された歌詞の歌唱データを選択し、その歌唱データを、CPU101から指示された音高に対応する読み出し速度で読み出して、必要に応じて波形サンプルデータの補間処理を実行し、最終的な出力波形Wを生成し、それを図1の発音ユニット106に出力する。これにより、出力波形Wの歌唱音声が放音される。

#### 【0036】

以上説明した第1の実施形態において、歌詞決定処理における鍵の組合せパターンについてであるが、歌詞決定処理のための歌詞指定の押鍵の後、歌唱生成処理のための押鍵を素早く行うために、歌詞指定のための鍵の組合せパターンと、発音したい音高の鍵は近い距離であることが好ましい。その一例として、歌詞指定の鍵の組合せパターンは図1の鍵盤104の1オクターブ内に収まるようにし、鍵盤104のどのオクターブにおいても同じ操作で同じ歌詞が指定できるようにすることが望ましい。

#### 【0037】

上記第1の実施形態において、単一鍵の押鍵からNミリ秒以上経ってから他の鍵の押鍵があった場合については、例えば以下のような制御処理が採用されてよい。

- ・ 単一鍵の押鍵からNミリ秒以上経ってから、その单一鍵が離鍵され他の单一鍵の押鍵があった場合は、その新規の押鍵の音で、歌詞を発音する（例えば、「かーかー」と発音する）。

- ・ 単一鍵の押鍵からNミリ秒以上経ってから、その单一鍵が離鍵されずに他の单一鍵の押鍵があった場合は、その新規の押鍵の音で、母音を繋げる形で歌唱する（例えば、「かーあー」と発音する）。

上記例は、逆でも良い。また、基本的に鍵盤操作のみでは常に現在の歌詞で発音しなおすようにし、サステインペダルを踏んでいる時のみ母音をつなげる、という制御動作が実行されてもよい。

#### 【0038】

なお、押鍵が歌詞指定か歌唱生成指定かの区別は、あるタイミングでの同時押鍵数が单一か複数かで区別可能である。よって、例えば、歌詞指定のために2つの鍵を同時に押鍵した後、離鍵せずに別の鍵を押鍵した場合、上記ではその押鍵は発音と見なされるが、その押鍵も歌詞指定の一部と見なし、3つの鍵の組み合わせによる歌詞に記憶されている歌詞を上書きするという制御動作が実行されてもよい。また、発音後に離鍵せずに別の鍵を押

10

20

30

40

50

鍵した場合、上記では先の発音から後の発音に独唱として遷移するように発音するが、先の発音を残したまま新たな発音をするという制御動作が実行されてもよい。

#### 【0039】

離鍵動作については、第1の実施形態では特に言及していないが、押鍵後所定時間で自然にリリースさせるか、あるいは演奏者の離鍵動作に対して所定の制御動作を実行させてもよい。

#### 【0040】

次に、本発明の第2の実施形態について説明する。図6は、第2の実施形態における歌詞決定処理の詳細例を示すフローチャートである。第2の実施形態では、RAM103上に歌詞の FIFO (ファーストインファーストアウト：先入れ先出し) 方式の歌詞キュー変数  $ly1ic\_queue$  を持つておく。そして、演奏者が複数鍵の同時押鍵を連続して実行するごとに、CPU101は、第1の実施形態と同様の図3のフローチャートで示される鍵盤処理におけるステップS309の歌詞決定処理を繰り返し実行し、この処理において、図6のステップS601に示されるように、各押鍵に対応する押鍵済集合変数  $on\_keys$  の内容でROM102内の鍵盤・歌詞変換テーブルを参照して決定される歌詞データを、第1の実施形態で示したRAM103上の歌詞変数  $ly1ic$  に上書きすることはせずに、歌詞キュー変数  $ly1ic\_queue$  に積んでゆく。

10

#### 【0041】

図7は、第2の実施形態における歌唱生成処理の詳細例を示すフローチャートである。上述の複数鍵の同時押鍵の連続処理の後、今度は演奏者が単数鍵の押鍵を連続して実行するごとに、CPU101は、図3の鍵盤処理におけるステップS311の歌唱生成処理を繰り返し実行する。この処理において、CPU101は、歌詞キュー変数  $ly1ic\_queue$  に積まれている先頭の歌詞データを  $pop$  し(取り出して先頭を取り去り)、それに対応する歌唱データを歌詞変数  $ly1ic$  に代入する(ステップS701)。その後は、第1の実施形態における図5のステップS501およびS502と同様の、ステップS702およびS703の処理を実行し、歌唱データの出力波形Wを生成して発音する。演奏者が単数鍵の押鍵を行うごとに、図7のフローチャートで示される歌唱生成処理が実行され、歌詞キュー変数  $ly1ic\_queue$  中の歌唱データが1つずつ  $pop$  されてゆく。このようにして、単数鍵による音高が指定されるごとに、最初に一括して指定された各歌詞に対応する各歌唱データを順次生成し発音させることができる。

20

#### 【0042】

なお、歌詞キュー変数  $ly1ic\_queue$  が空の場合は、発音しない、最後の歌詞で発音する、最後の歌詞の母音で発音する、というような制御動作を実行してよい。また、歌詞キュー変数  $ly1ic\_queue$  があふれる場合は、キューの先頭の歌詞データを捨て最後に追加、キューの最後の歌詞データを上書き、というような制御動作を実行してよい。第1の実施形態は、第2の実施形態において、歌詞キュー変数  $ly1ic\_queue$  の最大長が1である場合に相当する。

30

#### 【0043】

次に、本発明の第3の実施形態について説明する。第3の実施形態は、前述した第1の実施形態に対して、歌詞決定処理と歌唱生成処理が必ず交互に実行されるようにしたものである。第3の実施形態では、複数鍵による音高指定が可能となるため、歌唱は独唱に限らず、合唱演奏が可能になる。

40

#### 【0044】

図8は、第3の実施形態における図2のステップS203の鍵盤処理の詳細例を示すフローチャートである。各集合変数については、図3で前述した第1の実施形態の場合と同様である。

#### 【0045】

まず、CPU101は、鍵盤104をスキャンし、押鍵されている鍵の集合をRAM103上の鍵集合変数  $c\_keys$  に代入する(ステップS801)。

#### 【0046】

50

次に、CPU101は、前回鍵集合変数 p\_key\_s の内容が空集合ではなく、かつ今回鍵集合変数 c\_key\_s の内容が空集合であるか否かを判定する(ステップ S802)。ステップ S802 の判定が真(以下「TRUE」と記載)である場合は、前回までの押鍵の全てが離鍵された状態を示し、偽(以下「FALSE」と記載)である場合、今回いずれかの鍵が押鍵されている状態を示している。

#### 【0047】

ステップ S802 の判定が FALSE の場合、CPU101 は次に、前回鍵集合変数 p\_key\_s の内容と今回鍵集合変数 c\_key\_s の内容が異なっているか否かを判定する(ステップ S803)。

#### 【0048】

前回鍵集合変数 p\_key\_s の内容と今回鍵集合変数 c\_key\_s の内容が同じ(ステップ S803 の判定が FALSE)である場合には、CPU101 は、次回の鍵盤処理のためにステップ S801 にて算出された今回鍵集合変数 c\_key\_s の内容を前回鍵集合変数 p\_key\_s に格納した後に(ステップ S813)、そのまま図 8 のフローチャートで示される図 2 のステップ S203 の今回の鍵盤処理を終了する。

#### 【0049】

前回鍵集合変数 p\_key\_s の内容と今回鍵集合変数 c\_key\_s の内容が異なっている(ステップ S803 の判定が TRUE である)場合には、CPU101 は、現在の段階が歌詞決定処理を実行する段階 LYRIC であるか歌唱生成処理を実行する段階 PLAY であるかを示す RAM103 上の段階変数 phase の内容がチェックされる。

#### 【0050】

この段階変数 phase の初期値は、図 2 のステップ S201 で歌詞決定処理を実行する段階 LYRIC に設定されているため、CPU101 は、最初はステップ S804 から S805 に移行する。ここでは、CPU101 は、押鍵済集合変数 on\_key\_s の内容に、今回鍵集合変数 c\_key\_s と前回鍵集合変数 p\_key\_s の差分集合の内容を加えた後(ステップ S805)、第 1 の実施形態における図 3 のステップ S309 と同様の歌詞決定処理を実行する(ステップ S806)。その後、CPU101 は、次回の鍵盤処理のためにステップ S801 にて算出された今回鍵集合変数 c\_key\_s の内容を前回鍵集合変数 p\_key\_s に格納し(ステップ S813)、図 8 のフローチャートで示される図 2 のステップ S203 の今回の鍵盤処理を終了する。

#### 【0051】

その後、演奏者が、歌詞決定処理のために押鍵していた鍵を離鍵すると、その直後に図 2 のステップ S203 の鍵盤処理が実行されるタイミングで、図 8 のステップ S802 の判定が TRUE となる。この場合、CPU101 は、段階変数 phase の内容が歌詞決定処理を実行する段階 LYRIC であるか歌唱生成処理を実行する段階 PLAY であるかを判定する。

#### 【0052】

歌詞決定処理が実行されたときには段階変数 phase の内容は LYRIC であるため(ステップ S804 S805)、CPU101 は、ステップ S809 から S810 に移行する。ここでは、CPU101 は、段階変数 phase の内容を歌唱発音処理を実行する段階 PLAY に変更する。その後、CPU101 は、次回の鍵盤処理のためにステップ S801 にて算出された今回鍵集合変数 c\_key\_s の内容を前回鍵集合変数 p\_key\_s に格納し(ステップ S813)、図 8 のフローチャートで示される図 2 のステップ S203 の今回の鍵盤処理を終了する。

#### 【0053】

この結果、次に演奏者が押鍵をしてから鍵盤処理が実行されるタイミングでは、ステップ S802 の判定が FALSE、ステップ S803 の判定が TRUE、ステップ S804 の判定が PLAY となる。この結果、CPU101 は、第 1 の実施形態における図 3 のステップ S310 および S311 と同様のステップ S807 および S808 を実行し、歌唱生成処理を実行する。その後、CPU101 は、次回の鍵盤処理のためにステップ S80

10

20

30

40

50

1 にて算出された今回鍵集合変数 `c_key_s` の内容を前回鍵集合変数 `p_key_s` に格納し（ステップ S 8 1 3）、図 8 のフローチャートで示される図 2 のステップ S 2 0 3 の今回の鍵盤処理を終了する。

#### 【 0 0 5 4 】

その後、演奏者が、歌唱生成処理のために押鍵していた鍵を離鍵すると、その直後に図 2 のステップ S 2 0 3 の鍵盤処理が実行されるタイミングで、図 8 のステップ S 8 0 2 の判定が TRUE となり、CPU 1 0 1 は、ステップ S 8 0 9 で、段階変数 `phase` の内容を再びチェックする。

#### 【 0 0 5 5 】

歌唱生成処理が実行されたときには段階変数 `phase` の内容は PLAY であるため（ステップ S 8 0 4 S 8 0 7）、CPU 1 0 1 は、ステップ S 8 0 9 から S 8 1 1 に移行する。ここでは、CPU 1 0 1 は、段階変数 `phase` の内容を歌詞決定処理を実行する段階 LYRIC に変更する。また、CPU 1 0 1 は、押鍵済集合変数 `on_key_s` の内容を空集合 にする（ステップ S 8 1 2）。その後、CPU 1 0 1 は、次回の鍵盤処理のためにステップ S 8 0 1 にて算出された今回鍵集合変数 `c_key_s` の内容を前回鍵集合変数 `p_key_s` に格納し（ステップ S 8 1 3）、図 8 のフローチャートで示される図 2 のステップ S 2 0 3 の今回の鍵盤処理を終了する。

#### 【 0 0 5 6 】

以上のようにして、第 3 の実施形態では、演奏者が押鍵と離鍵を繰り返すごとに、歌詞決定処理を実行する段階 LYRIC と歌唱生成処理を実行する段階 PLAY がトグル的に交互に繰り返される。このため、演奏者は、歌唱生成処理を実行させるために押鍵を行う場合、第 1 の実施形態のように単数鍵の押鍵を行う必要は必ずしもなく、複数鍵での押鍵指定も行うことができ、合唱歌唱が可能となる。

#### 【 0 0 5 7 】

歌唱発音処理において同時に複数鍵を順に押鍵した場合は、押鍵された鍵毎に対応して発音を制御するようにしてもよい。この際も、前述のように、

- ・ 単一鍵の押鍵から N ミリ秒以上経ってから、その单一鍵が離鍵されずに他の单一鍵の押鍵があった場合は、その新規の押鍵の音で、母音を繋げる形で歌唱する（例えば、「かーあー」と発音する）。

としてもよい。また、後から押鍵した分の発音は、母音だけでなく、子音を含んでもよい。なお、第 3 の実施形態では、歌唱生成処理を実行する段階 PLAY から歌詞決定処理を実行する段階 LYRIC に戻るのは全て離鍵したタイミングであるが、最終的に全て離鍵する直前まで押鍵されていた鍵の組み合わせで歌詞を決定して、歌唱生成処理を実行する段階 PLAY に移るようにしてよい。

また、歌詞決定処理を実行する段階 LYRIC において、最初の押鍵時において、そのときに同時に押した鍵の組み合わせで歌詞を決定して、その後の押鍵では歌詞は変化しないようにし、全鍵の離鍵後に、歌唱生成処理を実行する段階 PLAY に移る（実際の発音は再押鍵後）ようにしてもよい。

さらに、歌詞決定処理を実行する段階 LYRIC で、最初に離鍵が為された場合にその離鍵した鍵の組み合わせで歌詞を決定、その後に歌唱生成処理を実行する段階 PLAY に移行してもよい。

#### 【 0 0 5 8 】

図 9 は、歌唱合成装置の第 4 の実施形態のハードウェア構成例を示す図である。この構成が、第 1、第 2、第 3 の実施形態の図 1 に示される構成と異なるのは、フットペダル 107 が備えられている点である。第 4 の実施形態では、歌詞決定処理を実行する段階と歌唱生成処理を実行する段階の切替えが、フットペダル 107 によって行われる。例えば、フットペダル 107 を踏んでいる時は歌詞決定処理が実行される段階、離しているときは歌唱生成処理が実行される段階とする（トグルによる切替えでもよい）。

#### 【 0 0 5 9 】

図 10 は、第 4 の実施形態における図 2 のステップ S 2 0 3 の鍵盤処理の詳細例を示す

10

20

30

40

50

フローチャートである。

**【0060】**

まず、CPU101は、鍵盤104をスキャンし、押鍵されている鍵の集合をRAM103上の鍵集合変数c\_keysに代入する(ステップS1001)。

**【0061】**

次に、CPU101は、今回鍵集合変数c\_keysの内容から前回鍵集合変数p\_keysの内容を差し引いた結果が空集合であるか否か、すなわち、前回鍵集合変数p\_keysの内容と今回鍵集合変数c\_keysの内容が同じであるか否かを判定する(ステップS1002)。

**【0062】**

ステップS1002の判定がTRUEである場合には、CPU101は、そのまま図10のフローチャートで示される図2のステップS203の鍵盤処理を終了する。

**【0063】**

ステップS1002の判定がFALSEである場合には、CPU101は、図9のフットペダル107の状態を確認する(ステップS1003)。

**【0064】**

フットペダル107が踏まれている状態である場合には、CPU101は、第3の実施形態におけるステップS805およびS806と同様のステップS1004およびS1005の処理で示される歌詞決定処理を実行する。その後、CPU101は、次回の鍵盤処理のためにステップS1001にて算出された今回鍵集合変数c\_keysの内容を前回鍵集合変数p\_keysに格納し(ステップS1008)、図10のフローチャートで示される図2のステップS203の今回の鍵盤処理を終了する。

**【0065】**

フットペダル107が踏まれていない状態である場合には、CPU101は、第3の実施形態におけるステップS807およびS808と同様のステップS1006およびS1007の処理で示される歌唱生成処理を実行する。その後、CPU101は、次回の鍵盤処理のためにステップS1001にて算出された今回鍵集合変数c\_keysの内容を前回鍵集合変数p\_keysに格納し(ステップS1008)、図10のフローチャートで示される図2のステップS203の今回の鍵盤処理を終了する。

**【0066】**

従来技術では、リアルタイムの歌詞の指定ができなかったり、歌唱演奏のみで両手が塞がってしまっていたりしたが、以上説明した第1、第2、第3、および第4の実施形態により、片手のみで歌唱演奏が可能になるため、もう片方の手で他の演奏操作子を操作することが可能になり、演奏の幅を広げることができる。また、例えば鍵盤を上半分と下半分に分け、それぞれの鍵域で別個の歌手データを使用し、異なる歌詞で同時に歌唱させることができるようになる。

第1～第4実施形態においては、演奏操作子として鍵盤の鍵を用いているが、これに限るものでない。例えば打楽器音を発音させる複数種のパッド、PC等に用いられるキーボードなど、演奏操作子として用いられるものであれば何でもよい。

**【0067】**

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)

演奏操作子により行われる演奏操作を判別し、当該判別結果に基づいて第1及び第2のいずれかの状態に移行する状態移行部と、

前記第1の状態に移行した場合に、前記判別された演奏操作に基づいて歌詞を決定する歌詞決定部と、

前記第2の状態に移行した場合に、前記判別された演奏操作に基づいた音高で歌唱させるための歌唱データ生成する歌唱生成部と、

前記歌詞生成部にて生成された歌唱データに基づいて発音させる発音データを生成する発音生成部と、

10

20

30

40

50

を備えた歌唱生成装置。

(付記 2 )

前記歌詞決定部は、予め定められた期間内に前記演奏操作子が複数同時に操作された場合に前記第1の状態に移行し、前記演奏操作子を単数操作した場合に第2の状態に移行する処理を実行する、付記1に記載の歌唱生成装置。

(付記 3 )

前記歌詞決定部はさらに、前記決定された歌詞を順次メモリに記憶させ、

前記歌唱生成部は、前記音高決定部にて音高が決定される毎に、前記メモリに記憶されている歌詞を予め記憶された順に読み出すとともに、当該読み出された歌詞を前記決定された音高で歌唱させる歌唱データを生成する、付記1又は2に記載の歌唱生成装置。 10

(付記 4 )

演奏操作子により行われる演奏操作の内容及び当該演奏操作の終了を判別し、当該演奏操作子の終了を判別する毎に、第1及び第2のいずれか一方の状態から他方の状態に移行する状態移行部と、

前記第1の状態に移行した場合に、前記判別された演奏操作の内容に基づいて歌詞を決定する歌詞決定部と、

前記第2の状態に移行した場合に、前記判別された演奏操作の内容に基づいた音高で歌唱させるための歌唱データ生成する歌唱生成部と、

前記歌詞生成部にて生成された歌唱データに基づいて発音させるための発音データを生成する歌唱生成部と、 20

を備えた歌唱生成装置。

(付記 5 )

前記演奏操作子は、フットペダルを含み、前記状態移行部は、前記フットペダルが操作される毎に、前記第1及び第2の状態のいずれか一方の状態から他方の状態に移行させる、付記1に記載の歌唱生成装置。

(付記 6 )

演奏操作子により行われる演奏操作を判別し、当該判別結果に基づいて第1及び第2のいずれかの状態に移行する処理と、前記第1の状態に移行した場合に、前記判別された演奏操作に基づいて歌詞を決定する処理と、前記第2の状態に移行した場合に、前記判別された演奏操作に基づいた音高で歌唱させるための歌唱データ生成する処理と、を実行する制御部と、 30

前記歌詞生成部にて生成された歌唱データに基づいて発音させるための発音データを生成する歌唱生成部と、

を備えた歌唱生成装置。

(付記 7 )

制御部と歌唱生成部とを備えた歌唱生成装置に用いられる歌唱生成方法であって、

前記制御部が、

演奏操作子により行われる演奏操作を判別し、当該判別結果に基づいて第1及び第2のいずれかの状態に移行させ、

前記第1の状態に移行した場合に、前記判別された演奏操作に基づいて歌詞を決定し、

前記第2の状態に移行した場合に、前記判別された演奏操作に基づいた音高で歌唱させるための歌唱データ生成し、 40

前記歌唱生成部が、

前記生成された歌唱データに基づいて発音させるための発音データを生成する、歌唱生成方法。

(付記 8 )

生成された歌唱データに基づいて発音させるための発音データを生成する歌唱生成部を有する歌唱生成装置として用いられるコンピュータに、

演奏操作子により行われる演奏操作を判別し、当該判別結果に基づいて第1及び第2のいずれかの状態に移行するステップと、 50

前記第1の状態に移行した場合に、前記判別された演奏操作に基づいて歌詞を決定するステップと、

前記第2の状態に移行した場合に、前記判別された演奏操作に基づいた音高で歌唱させるための歌唱データ生成するステップと、

を実行させるプログラム。

(付記9)

付記1に記載の歌唱生成装置と、

複数の演奏操作子と、

前記歌唱生成装置からの発音データに基づいて歌唱音を発音する発音部と、  
を備えた電子楽器。

10

【符号の説明】

【0068】

100 歌唱生成装置

101 CPU

102 ROM

103 RAM

104 鍵盤

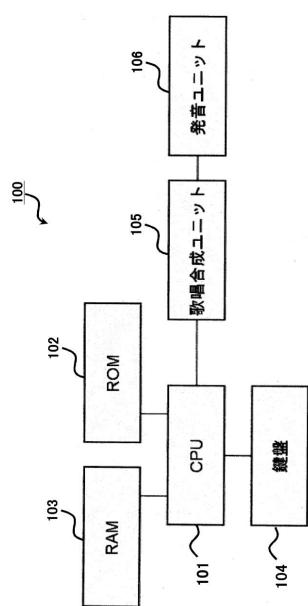
105 歌唱合成ユニット

106 発音ユニット

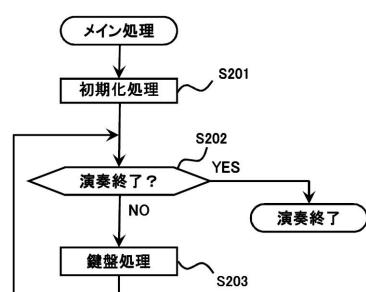
107 フットペダル 107

20

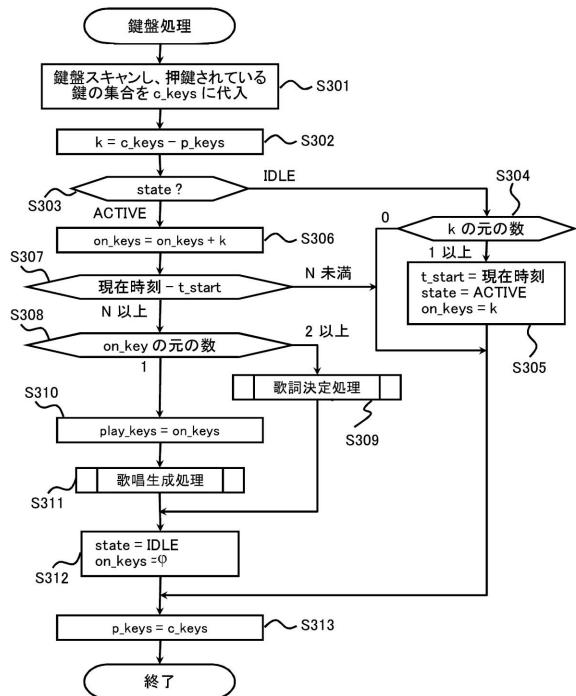
【図1】



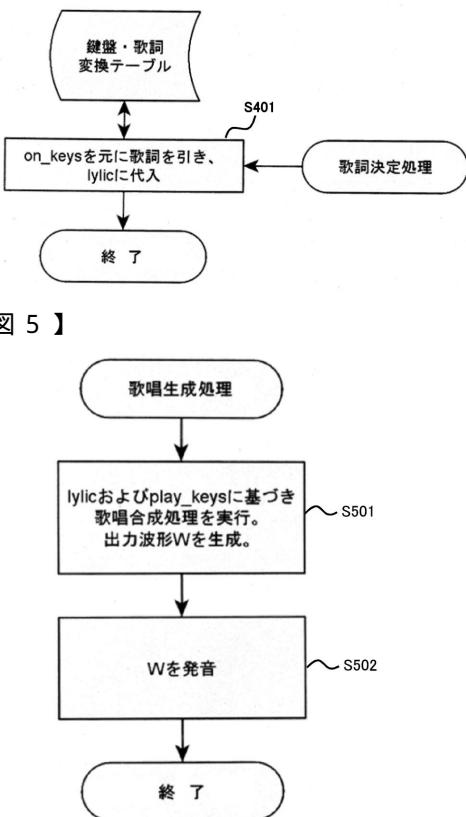
【図2】



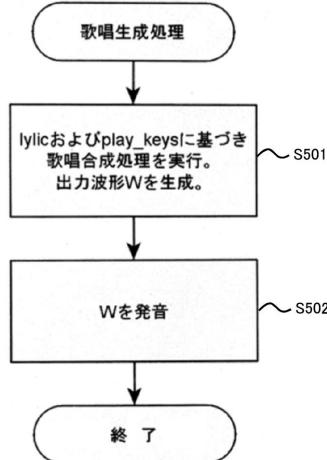
【図3】



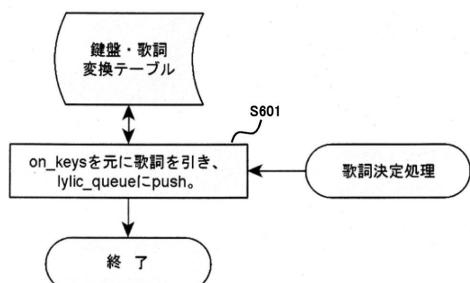
【図4】



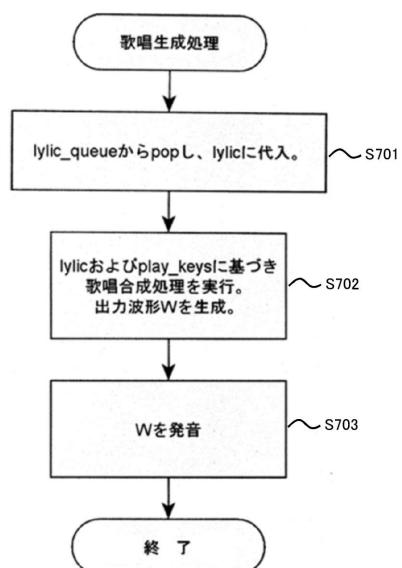
【図5】



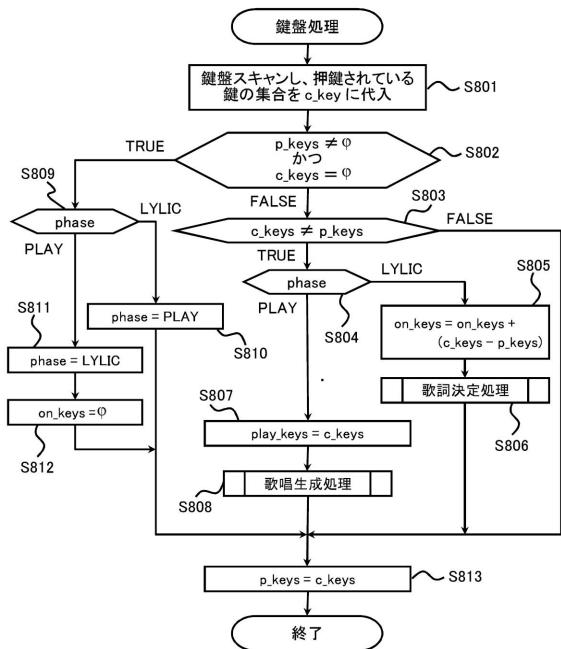
【図6】



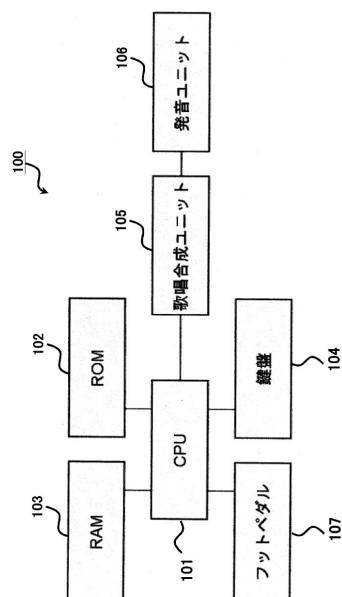
【図7】



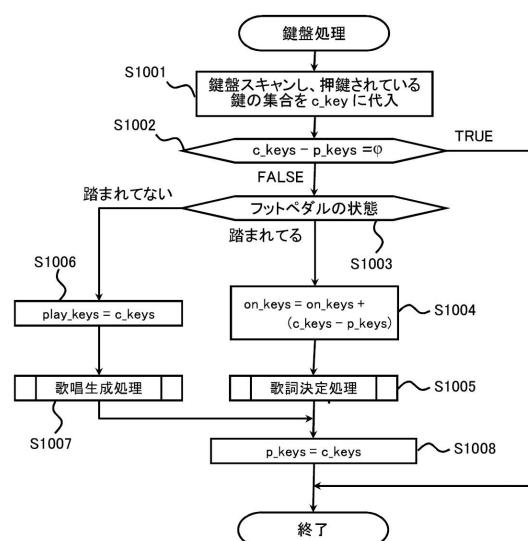
【図8】



【図9】



【図10】



---

フロントページの続き

(56)参考文献 特開平06-324677(JP,A)  
特開2011-027865(JP,A)  
実開平05-090593(JP,U)

(58)調査した分野(Int.Cl., DB名)

G 10 H	1 / 0 0	-	7 / 1 2
G 10 L	1 3 / 0 0	-	1 3 / 1 0