



# [12] 发明专利说明书

专利号 ZL 02108025.9

[45] 授权公告日 2006年6月7日

[11] 授权公告号 CN 1258751C

[22] 申请日 2002.3.22 [21] 申请号 02108025.9

[30] 优先权

[32] 2001.3.23 [33] JP [31] 2001-086163

[32] 2001.3.23 [33] JP [31] 2001-086164

[32] 2001.3.23 [33] JP [31] 2001-086165

[32] 2001.3.23 [33] JP [31] 2001-086166

[71] 专利权人 雅马哈株式会社

地址 日本静冈县

[72] 发明人 田邑元一

审查员 杜 轶

[74] 专利代理机构 隆天国际知识产权代理有限公司

代理人 陈 红 楼仙英

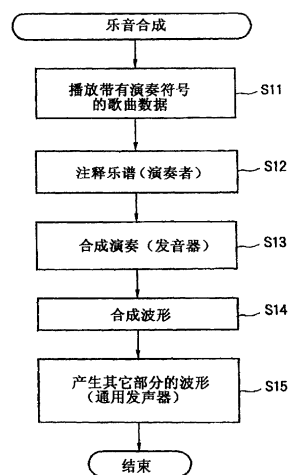
权利要求书 4 页 说明书 56 页 附图 29 页

## [54] 发明名称

乐音合成方法和用于合成乐音的设备

## [57] 摘要

本发明提供乐音合成方法和用于合成乐音的设备。该方法包括接收沿时间轴连续播放乐音的包流，所述包流包含先行包和接续包。每个包中含有对存储在第一存储装置中的特定矢量数据进行有效识别的信息。根据包含在先行包中的识别信息预测接续包中的矢量数据。在接收到接续包之前，把预测的矢量数据从第一存储装置移送到第二存储装置，以便将预测的矢量数据高速缓存到第二存储装置中。当接收到接续包并且包含在接续包中的识别信息命中高速缓存的矢量数据时，从第二存储装置中检索高速缓存的矢量数据并由此合成乐音。



1. 一种乐音合成方法，借助于第一存储装置和第二存储装置合成乐音，其中，该第一存储装置存储代表各种乐音的波形的矢量数据，所述的第一存储装置按第一存取速率存取，该第二存储装置以快于该第一存取速率的第二存取速率高速缓存该矢量数据，用于迅速提供代表将要合成的乐音波形的矢量数据，该方法包括以下步骤：

接收沿时间轴连续播放乐音的包流，每个包均含有：识别信息，该信息可有效识别存储在该第一存储装置中用于合成的矢量数据；以及时间信息，该信息可有效指定沿该时间轴合成的时间；

10 在规定的時間之前，將识别的矢量数据从该第一存储装置传输到该第二存储装置，由此可在合成之前，将识别出的矢量数据及时地高速缓存在该第二存储装置中；和

当达到规定的时间时，从该第二存储装置中检索高速缓存的矢量数据，以便将由检索到的矢量数据定义的波形合成为乐音。

15 2. 根据权利要求 1 所述的方法，还包括以下步骤：判断是否已将由接收到的包识别出的矢量数据高速缓存到该第二存储装置中，从而在判断步骤判断该第二存储装置还没有存储所识别的矢量数据时，在传输步骤中对所识别的矢量数据进行传输。

20 3. 根据权利要求 1 所述的方法，还包括以下步骤：检测该第二存储装置何时没有足够的空闲区域来接收将要传输到该第二存储装置中的新矢量数据，以及从该第二存储装置中释放旧矢量数据，以产生接收新矢量数据的空闲区域，所释放的旧矢量数据包括已在先前的合成中用过的矢量数据和在先前的合成中一直未用的矢量数据。

25 4. 根据权利要求 3 所述的方法，其中使用矢量数据合成具有预定长度的乐音波形，所述矢量数据包括用于合成预定波形长度前部的有实义的矢量数据，和用于合成预定波形长度剩余部分的无实义的矢量数据，其中所述释放步骤释放无实义的矢量数据比释放有实义的矢量数据更容易。

5. 根据权利要求 1 所述的方法，其中，该第一存储装置设置成硬盘驱动器的形式，而该第二存储装置设置成随机存取存储器芯片的形式。

30 6. 根据权利要求 1 所述的方法，还包括以下步骤：

对应于先前的包保持合成中使用的矢量数据，以便在另一次合成中，对应于后面的包再次使用所述数据。

7. 根据权利要求 1 所述的方法，其中，所述传输步骤周期性地传输足以成连续方式传输的预定数量的矢量数据。

5 8. 根据权利要求 1 所述的方法，还包括以下步骤：

指出用于在合成时间之前准备合成乐音的矢量数据，指出的矢量数据包括表示已确定或预测将在合成中使用的波形的矢量数据；

在所指出的矢量数据还没有被高速缓存时，把具有相关计数的所指出的矢量数据从该第一存储装置传输到第二存储装置中，用于高速缓存，其中  
10 所述相关计数被设定为初始值；

在已经高速缓存了所指出的矢量数据时，递增所指出的矢量数据的相关计数的当前值；

在到达合成时间时，利用所指出并高速缓存的矢量数据进行乐音合成；

15 然后，递减在合成中使用的所指出并高速缓存的矢量数据的相关计数的当前值；以及

根据所述相关计数的当前值检测在该第二存储装置中的可释放的矢量数据。

9. 根据权利要求 8 所述的方法，其中该指出步骤根据在接收到的包  
20 中所含的识别信息指出矢量数据，从而对于多个接收到的包可以共同指出矢量数据，而该传输步骤将共同指出的矢量数据一次传输到该第二存储装置。

10. 根据权利要求 9 所述的方法，还包括以下步骤：设定在包的接收  
25 和相应乐音合成之间的时间间隔，从而使所述指出步骤能够在设定的时间间隔内指出对于多个接收到的包共有的矢量数据。

11. 根据权利要求 10 所述的方法，其中所述指出步骤根据在接收到的包中所含的识别信息两次或更多次地指出同一矢量数据，所述递增步骤在每次指出同一矢量数据时使高速缓存的矢量数据的相关计数的当前值  
30 递增，而所述递减步骤在每次使用同一矢量数据合成乐音时，使高速缓存的矢量数据的相关计数的当前值递减。

12. 一种用于合成乐音的设备，包括：

第一存储装置，存储代表各种乐音波形的矢量数据，该第一存储装置以第一存取速率存取；

5 第二存储装置，以快于该第一存取速率的第二存取速率高速缓存该矢量数据；

控制器，控制从该第一存储装置到该第二存储装置的矢量数据的传输，用于合成音乐，其中所述控制器包括：

10 接收部分，用于接收沿时间轴连续播放乐音的包流，每个包均含有可有效识别在该第一存储装置中存储的用于合成的矢量数据的识别信息、和有效指定沿该时间轴的合成时间的信息；

传输部分，在指定时间之前，将识别的矢量数据从该第一存储装置传输到该第二存储装置，从而在合成前，将识别的矢量数据及时地高速缓存在该第二存储装置中；和

15 检索部分，在达到指定时间时，从该第二存储装置中检索高速缓存的矢量数据，以将检索到的矢量数据定义的波形合成为乐音。

13. 根据权利要求 12 所述的设备，其中该控制器还包括：

指出部分，用于指出准备乐音合成的矢量数据，所指出的矢量数据包括合成时使用的代表确定波形或预测波形的矢量数据；

20 指令部分，用于根据矢量数据指示合成的开始，所述矢量数据高速缓存在该第二存储装置中，而且可以包含或不包含所指出的矢量数据；

另一传输部分，在没有高速缓存所指出的矢量数据时，将具有相关计数的所指出的矢量数据从该第一存储装置传输到该第二存储装置，所述相关计数被设定为初始值；

25 递增部分，用于在已经高速缓存了所指出的矢量数据时，使所指出的矢量数据的相关计数的当前值递增；

执行部分，响应于开始指令，利用所指出并高速缓存的矢量数据执行乐音的合成；

递减部分，使在合成中使用的、所指出并高速缓存的矢量数据的相关计数的当前值递减；以及

30 检测部分，按照相关计数的值，检测在该第二存储装置中可释放的矢

量数据。

14. 根据权利要求 12 所述的设备，其中该控制器还包括：

保持部分，其对应于先前的包保持在一合成中使用的矢量数据，用于对应于后面的包在另一合成中再次使用。

## 乐音合成方法和用于合成乐音的设备

### 5 技术领域

本发明涉及一种声音数据传输方法，声音数据传输设备，和根据从较低速度的存储介质，例如，以较低存取速率存储数据的硬盘中读取的信息，产生乐音、语音和任何其他声音的设备程序。更确切地说，本发明涉及波形的产生或合成，所产生或合成的波形能根据演奏者不同的演奏风格或常规乐器的发音特性准确地表现乐音的音调。本发明不仅广泛应用于电子乐器，而且还适用于能够产生乐音、语音或所有其他声音的所有领域中的装置、设备或方法。所述装置、设备包括例如机械乐器、计算机、电子游戏设备和其它多媒体装置。在本说明书中，乐音波形并不限于纯音乐感觉的波形，还可以包括语音或其它任何声音的波形。

15

### 背景技术

“波形存储器读取”技术是众所周知的技术。该技术能够存储用任何编码系统例如 PCM（脉冲编码调制）、DPCM（差分 PCM）和 ADPCM（自适应差分 PCM）编码的波形数据（例如波形采样数据），并根据所需要的音乐定音读取存储的数据，以产生或合成乐音。此外，公知的还有各种类型的“波形存储器读取系统”技术。可以用最传统的“波形存储器读取系统”技术产生从发音期开始到发音期结束的单音波形。一个实例是存储从发音期开始到发音期结束的整个声音波形的波形数据。另一个实例是用复杂变化在声音硬起音（attack）部分，存储部分波形数据，和用少变化在保持部分存储连续部分的特定环状波形（loop waveform）。在本说明书中，

25

术语“环状波形”指的是在一个环状中反复读取的波形。

公知的作为存储波形数据的装置有 ROM、RAM、硬盘、CD-ROM 等。硬盘和 CD-ROM 每单位存储容量的单位价格较低，并且，适合存储大量数据。然而，硬盘和 CD-ROM 的存取速度较慢且不稳定，不能在刚好输出乐音信号的时刻瞬间读取所需波形数据。为此，而提出了以下技术。

日本待审专利申请公开号平成 6-308964（对应于美国专利 5463183）中公开了预先向 RAM 传送存储在硬盘中的多个波形数据起始部分的技术。即，当给出发声指令时，开始执行从硬盘中先行读取合适波形的数据后续部分的操作，并且同时播放先前存储在 RAM 中的数据起始部分。在完成了播放数据开始部分的操作后，播放波形数据的后续部分。

日本待审专利申请公开号昭 63-181188 公开了通过顺序读取从而播放波形数据的技术。将顺序发声的波形数据预先定义为序列数据。该技术预定了与时序对应的每个波形数据的开始读取时间，从而使启动时间早于添加音符时刻的时间。

15

### 发明内容

然而，根据日本待审专利申请公开号平成 6-308964 的技术，需要预先向 RAM 传送所有波形数据的起始部分，因此妨碍了精密 RAM 的有效使用。由于按照日本待审专利申请公开号昭 63-181188 的技术，是以非实时产生乐音信号的假设为基础的，所以在实时提供音乐序列数据的情况下不可能应用该技术。

本发明是在考虑了上述问题的情况下做出的。因此，本发明的第一个目的是提供一种声音数据传输方法，声音数据传输设备，和有效地以及大量地使用 RAM 和同时使用硬盘等装置实时产生乐音信号用的设备程序。

25 按照日本待审专利申请公开号平成 6-308964 所述技术，需要预先向

RAM 传输所有波形数据的开始部分。按照日本待审专利申请公开号昭 63-181188 所述的技术，总是要求硬盘存取每一个事件，因此，增加了来自硬盘的噪音并会缩短硬盘的寿命。此外，硬盘频繁存取会消耗应分配给其他程序的额外资源。

5 本发明是在考虑了上述问题的情况下做出的。因此，本发明的第二个目的是提供一种声音数据传输方法，声音数据传输设备，和用于减少硬盘不必要存取的设备程序。

通常，“波形存储器读取系统”技术存储的是一种从声音发声期开始到发声期结束这一完整波形的波形数据，或存储作为硬起音部分的一种声音中的分音波形数据。这种技术必须存储大量与各种演奏风格（或发音）  
10 对应的不同波形数据，因此，需要大储存容量来存储这些不同的波形数据。

上述用于存储完整波形之波形数据的系统能如实地表达因各种演奏风格（或发音）特别是常规乐器产生的音调（timbre）。然而，这种系统可以仅根据存储的波形数据来简单地播放音乐，并提供有限的可控性和  
15 编辑性。例如，很难根据音乐特性数据提供与所需演奏风格（或发音）相应的例如波形数据之时间轴控制的特征控制。

本发明是在考虑了上述问题的情况下做出的。因此，本发明的第三个目的是提供一种波形合成方法，声音数据传输设备，和能够容易、简单和实时产生适合各种演奏风格（或发音）的高质量波形数据的设备程序。

20 按照日本待审专利申请公开号平成 6-308964 所述技术，需要预先向 RAM 传输各种音调的全部波形数据的起始部分，因此不利于 RAM 的充分利用。此外，按照日本待审专利申请公开号昭 63-181188 所述的技术，总是要求硬盘存取每一个事件，因此，增加了来自硬盘的噪音，并会缩短硬盘的寿命。此外，硬盘频繁存取会损失机器的有限资源。

25 本发明是在考虑了上述问题的情况下做出的。因此，本发明的第四个

目的是提供一种声音数据传输方法，声音数据传输设备，和节约硬盘存取等的设备程序。

为了实现上述第一目的，本发明的特征在于具备以下构成特点。同时应注意括号内的实例。

5       按照本发明第一方案所提及的构成特点，涉及声音数据传输方法，该方法使用了用于存储乐音波形的声音数据(矢量数据)的低速存储装置(硬盘 109)，和用于高速缓存声音数据的高速存储装置(高速缓冲存储器 44)，并将存储在低速存储装置中的一部分声音数据输送到高速存储装置。所述方法的特征在于具有：接收音调类别(specification)(MIDI 程序变化  
10 信息)的音调类别接收程序(process)，所述音调类别与将要传输到高速存储装置的声音数据有关；预测候选声音数据的声音数据预测程序(步骤 S30)，所述候选声音数据是指后面将根据接收到的音调类别指定的数据；和将预测的候选声音数据从低速存储装置传输到高速存储装置的程序(步骤 S43)。

15       优选的是，存储在低速存储装置中的声音数据包括与发声起始部分(硬起音部分)对应的第一声音数据和与除发声开始部分之外的后续部分(主体部分)对应的第二声音数据，而按照音调类别预测的声音数据是第一声音数据。所述方法的特征在于还具有：接收任何第一类声音数据类别的第一类声音数据类别接收程序；预测候选的第二声音数据的第二声音数  
20 据预测程序(步骤 S30)，所述候选的第二声音数据是指随后将根据第一声音数据类别指定的第二声音数据；和用于将预测的第二声音数据从低速存储装置传送到高速存储装置的程序(程序 43)。

      优选的是，存储在低速存储装置中的声音数据还包括在第一和第二数据之后使用的第三声音数据(释放(release)部分或接合部分)。所述方  
25 法的特征在于还具有：接收所有第二声音数据类别的第二类声音数据类别

接收程序；预测候选的第三声音数据的第三声音数据预测程序（步骤 S33），所述候选的第三声音数据是指随后将根据第二声音数据类别指定的第三声音数据；和用于将预测的第三声音数据从低速存储装置传送到高速存储装置的程序（程序 43）。

- 5 优选的是，本发明所述声音数据传输方法还具有提供识别信息（设要素 dwStatus 为 ‘USED’）的程序，在接收任何声音数据类别时，这些信息都保存在高速存储装置中，而且这些信息能指示出释放未经预测的候选声音数据的可能性。

- 10 优选的是，本发明所述声音数据传输方法还具有，在音调类别接收程序之前，预先将每个音调的基本声音数据（替换页）传送到高速存储装置的程序。当未将在第一、第二、或第三数据类别接收程序中指定的声音数据传送到高速存储装置中时，用基本声音数据代替指定的声音数据。

为了实现上述第二目的，本发明的特征在于具备以下构成特点。同时应注意括号内的实例。

- 15 按照本发明第二方面的构成特点，提供一种声音数据传输方法，该方法使用了用于存储音乐波形声音数据（矢量数据）的低速存储装置（硬盘 109），和用于高速缓存声音数据的高速存储装置（高速缓冲存储器 44），并将存储在低速存储装置中的一部分声音数据输送到高速存储装置以便进行波形合成。该方法的特征在于具有：准备将要用到的确定的声音数据  
20 或准备将要用到的预测声音数据的准备程序（把矢量 ID 或包从预测控制部分 41 或演奏风格合成部分 101C 施加到预取部分 42）；通过使用预先已命令准备并存储在高速存储装置中的声音数据，发出开始合成命令的合成开始命令程序；当未将命令准备的声音数据存储于高速存储装置中时，将声音数据从低速存储装置传送到高速存储装置，并根据传送的声音数据  
25 在初始值上设定计数值（要素 dwCount）的传输程序（步骤 S42 和 S43）；

当己将命令准备的声音数据存储的高速存储装置中时,增加声音数据计数  
值(要素 dwCount)的增加程序:根据来自合成开始命令程序的开始命令,  
使用声音数据开始进行波形合成并减去(FILLED 状态:S64 中的预取)  
与所用声音数据有关的计数值(要素 dwCount)的波形合成程序;检测存  
5 储在高速存储装置中的声音数据,而且根据与每个声音数据对应的计数值  
预定在下一个开始命令中不使用的声音数据,并将检测的声音数据设定成  
可擦除状态(当要素 dwCount 达到“0”时,将 FILLED 状态 S64 转到 USED  
状态 S63)的设置程序。

优选的是,本发明所述声音数据传输方法的特征在于,还具有在播放  
10 开始时刻之前从事件序列中(音乐演奏数据),预取事件数据的预取程序,  
其中准备程序根据预取的事件数据确定与准备命令有关的声音数据,从而  
使多个使用同样声音数据的事件数据共用传输程序。

优选的是,本发明的声音数据传输方法的特征在于还具有设定播放开  
始时间的程序,它用从提供事件数据时刻开始的延迟时间设定播放开始  
15 间。通过这种方式,准备程序能够根据在延迟时间内连续提供的事件数据  
确定与准备命令有关的共享声音数据,因此,使用共享声音数据的多个事  
件数据共用传输程序。

优选的是准备程序根据单个事件数据执行相同声音数据适合的多个  
准备命令;而且波形合成程序在将声音数据作为单个事件数据使用时用来  
20 确定计数值。

为了实现上述第三目的,本发明的特征在于具有以下构成特点。同时  
应注意括号内的实例。

按照本发明第三方面所述的构成特点,涉及一种波形合成方法,该方  
法的特征在于具有:接收由多个包(packet)组成的包流(packet stream)  
25 的输入程序,所述多个包至少包含与存储在低速存储装置(硬盘 109)中

的矢量数据有关的类别信息（矢量 ID），和与波形发生有关的时间信息；把由包含在这些包中的类别信息指定的矢量数据从低速存储装置传送到高速存储装置的传输程序，其中传输程序在由包含在每个包内的时间信息指示的时间之前开始；和在对应每个包内时间信息的时刻执行的波形合成程序，在该程序中能播放存储在高速存储装置中的，并由每个包中的类别信息指定的矢量数据，和根据播放的矢量数据合成波形。

优选的是，本发明所述波形合成方法的特征在于还包括确定由接收到的包指定的矢量数据是否已经存入高速存储装置中的程序，其中执行传输程序的条件是，指定的矢量数据目前还没有传送到高速存储装置中。

10 优选的是，传输程序包括：确定是否有足够的容量存储最新传送到高速存储装置的矢量数据的程序；和在没有足够空闲容量的情况下，将波形生成时，释放已经使用过的旧矢量数据或未使用（未成功预测）的矢量数据的存储区域以便形成空白区（设要素 dwStatus 为 ‘FREE’）的释放程序。

15 为了实现上述第四目的，本发明的特征在于具有以下构成特点。同时应注意括号内的实例。

按照本发明第四方面所提及的构成特点，涉及声音数据传输方法，该方法使用了用于存储乐音波形的声音数据（矢量数据）的低速存储装置（硬盘 109），和用于高速缓存声音数据的高速存储装置（高速缓冲存储器 44），并将存储在低速存储装置中的一部分声音数据输送到高速存储装置以便进行波形合成。本发明方法的特征在于具有：根据发声命令（加音符）将声音数据从低速存储装置传送到高速存储装置的传输程序（步骤 S43）；和在使用了数据合成波形后将声音数据以可在将来使用的状态保留在高速存储装置内的保持程序（设要素 dwStatus 为 ‘USED’）。

25 优选的是，传输程序将声音数据从低速存储装置传送到高速存储装置

中有特定容量的多个单元内（存储块的多个单元内）。

优选的是，传输程序包括：检测是否有足够的空间存储最新传送到高速存储装置的声音数据的程序；在没有足够的空白空间的情况下，释放先前保留的声音数据，以建立空白区的程序（设要素 dwStatus 为 ‘USED’）；

5 和将最新传输的声音数据传送到空白区的程序。

优选的是，用声音数据对每个指定的部分（模式）进行波形合成，和所述方法的特征在于还包括：设定高速存储装置中数据管理方式的程序，该程序使得在每部分开始时使用的声音数据与其他声音数据相比不容易释放（有这种变化：在链接表开始之前添加要高速缓存的矢量数据和将其  
10 他矢量数据插在链接表中间）。

本发明的又一目的是提供一种乐音合成方法。该方法借助于第一存储装置和第二存储装置合成乐音，其中，该第一存储装置存储代表各种乐音的波形的矢量数据，所述的第一存储装置按第一存取速率存取，该第二存储装置以快于该第一存取速率的第二存取速率高速缓存该矢量数据，用  
15 于迅速提供代表将要合成的乐音波形的矢量数据，该方法包括以下步骤：接收沿时间轴连续播放乐音的包流，每个包均含有：识别信息，该信息可有效识别存储在该第一存储装置中用于合成的矢量数据；以及时间信息，该信息可有效指定沿该时间轴合成的时间；在规定的  
20 时间之前，将识别的矢量数据从该第一存储装置传输到该第二存储装置，由此可在合成之前，将识别出的矢量数据及时地高速缓存在该第二存储装置中；和当达到规定的时间时，从该第二存储装置中检索高速缓存的矢量数据，以便将由检索到的矢量数据定义的波形合成为乐音。

上述方法还包括以下步骤：判断是否已将由接收到的包识别出的矢量数据高速缓存到该第二存储装置中，从而在判断步骤判断该第二存储装置  
25 还没有存储所识别的矢量数据时，在传输步骤中对所识别的矢量数据进行

传输。

上述方法还包括以下步骤：检测该第二存储装置何时没有足够的空闲区域来接收将要传输到该第二存储装置中的新矢量数据，以及从该第二存储装置中释放旧矢量数据，以产生接收新矢量数据的空闲区域，所释放的旧矢量数据包括已在先前的合成中用过的矢量数据和在先前的合成中一直未用的矢量数据。

上述方中使用矢量数据合成具有预定长度的乐音波形，所述矢量数据包括用于合成预定波形长度前部的有实义的矢量数据，和用于合成预定波形长度剩余部分的无实义的矢量数据，其中所述释放步骤释放无实义的矢量数据比释放有实义的矢量数据更容易。

上述方法中，该第一存储装置设置成硬盘驱动器的形式，而该第二存储装置设置成随机存取存储器芯片的形式。

上述方法还包括以下步骤：对应于先前的包保持合成中使用的矢量数据，以便在另一次合成中，对应于后面的包再次使用所述数据。

上述方法中，所述传输步骤周期性地传输足以成连续方式传输的预定数量的矢量数据。

上述方法还包括以下步骤：指出用于在合成时间之前准备合成乐音的矢量数据，指出的矢量数据包括表示已确定或预测将在合成中使用的波形的矢量数据；在所指出的矢量数据还没有被高速缓存时，把具有相关计数的所指出的矢量数据从该第一存储装置传输到第二存储装置中，用于高速缓存，其中所述相关计数被设定为初始值；在已经高速缓存了所指出的矢量数据时，递增所指出的矢量数据的相关计数的当前值；在到达合成时间时，利用所指出并高速缓存的矢量数据进行乐音合成；然后，递减在合成中使用的所指出并高速缓存的矢量数据的相关计数的当前值；以及根据所述相关计数的当前值检测在该第二存储装置中的可释放的矢量数据。

上述方法中，该指出步骤根据在接收到的包中所含的识别信息指出矢量数据，从而对于多个接收到的包可以共同指出矢量数据，而该传输步骤将共同指出的矢量数据一次传输到该第二存储装置。

上述方法还包括以下步骤：设定在包的接收和相应乐音合成之间的时间间隔，从而使所述指出步骤能够在设定的时间间隔内指出对于多个接收到的包共有的矢量数据。

上述方法中，所述指出步骤根据在接收到的包中所含的识别信息两次或更多次地指出同一矢量数据，所述递增步骤在每次指出同一矢量数据时使高速缓存的矢量数据的相关计数的当前值递增，而所述递减步骤在每次使用同一矢量数据合成乐音时，使高速缓存的矢量数据的相关计数的当前值递减。

本发明的又一目的是提供一种用于合成乐音的设备。该设备包括：第一存储装置，存储代表各种乐音波形的矢量数据，该第一存储装置以第一存取速率存取；第二存储装置，以快于该第一存取速率的第二存取速率高速缓存该矢量数据；控制器，控制从该第一存储装置到该第二存储装置的矢量数据的传输，用于合成音乐，其中所述控制器包括：接收部分，用于接收沿时间轴连续播放乐音的包流，每个包均含有可有效识别在该第一存储装置中存储的用于合成的矢量数据的识别信息、和有效指定沿该时间轴的合成时间的信息；传输部分，在指定时间之前，将识别的矢量数据从该第一存储装置传输到该第二存储装置，从而在合成前，将识别的矢量数据及时地高速缓存在该第二存储装置中；和检索部分，在达到指定时间时，从该第二存储装置中检索高速缓存的矢量数据，以将检索到的矢量数据定义的波形合成为乐音。

上述设备中，该控制器还包括：指出部分，用于指出准备乐音合成的矢量数据，所指出的矢量数据包括合成时使用的代表确定波形或预测波形的矢量数据；指令部分，用于根据矢量数据指示合成的开始，所述矢量数

- 据高速缓存在该第二存储装置中,而且可以包含或不包含所指出的矢量数据;另一传输部分,在没有高速缓存所指出的矢量数据时,将具有相关计数的所指出的矢量数据从该第一存储装置传输到该第二存储装置,所述相关计数被设定为初始值;递增部分,用于在已经高速缓存了所指出的矢量数据时,使所指出的矢量数据的相关计数的当前值递增;执行部分,响应于开始指令,利用所指出并高速缓存的矢量数据执行乐音的合成;递减部分,使在合成中使用的、所指出并高速缓存的矢量数据的相关计数的当前值递减;以及检测部分,按照相关计数的值,检测在该第二存储装置中可释放的矢量数据。
- 5
- 10 上述设备中,该控制器还包括:保持部分,其对应于先前的包保持在一合成中使用的矢量数据,用于对应于后面的包在另一合成中再次使用。

#### 附图说明

- 图 1 是按照本发明所述波形发生器硬件结构范例的方框图;
- 图 2 是表示在波形发生器内执行的“波形数据库建立程序”的实施例流程图;
- 15 图 3 表示产生与演奏风格模式相应的实际波形的元器件实例示意图;
- 图 4 是表示“根据数据库完成乐音合成程序”的实例流程图;
- 图 5 是与图 4 所述一样,以给定的硬件形式表示完成相同波形合成程序的实施例方框图;
- 20 图 6 是表示在上述演奏合成部分中演奏合成程序的流程方框图;
- 图 7 是详细表示在演奏合成部分中完成的演奏合成程序的实施例流程图;
- 图 8 是用于解释当演奏风格模式与振幅分量或定音分量相对应时的链接程序示意图;
- 25 图 9 是用于说明当硬起音部分的波形与主体部分波形相接时波形变薄

的示意图；

图 10 是用于说明当主体波形与释放波形相接时波形变薄的示意图；

图 11 是用于说明当转弯的硬起音波形与释放波形相接时波形变薄的示意图；

5 图 12 是用于说明当正常的硬起音波形与具有环形部分的释放波形相接时波形变薄的示意图；

图 13 是用于说明当一个演奏风格模式结束而在随后的另一个演奏风格模式启动前的演奏风格模式所波形连接示意图；

图 14 是用于说明在演奏合成区内产生的包流示意图；

10 图 15 表示在波形合成区内工作的所有结构示意图；

图 16 是用于简要表示所有波形合成的过程方框图；

图 17 是用于解释矢量加载器的方框图；

图 18 是用于解释矢量运算器的方框图；

图 19 是用于解释矢量记录器的方框图；

15 图 20 表示矢量数据之数据结构的实施例示意图；

图 21 表示在乐谱注释部分 101B 内建立的音乐演奏数据的内容示意图；

图 22 表示把包从演奏合成部分送到波形合成部分的时间示意图；

图 23 是表示高速缓存控制部分所有结构的方框图；

20 图 24 是用于在预测控制部分内进行预测运算的状态过渡图；

图 25 是表示在预取部分完成的加载过程的流程图；

图 26 是表示在预取部分完成的数据接收过程的流程图；

图 27 表示构成高速缓存页面的操作示意图；

图 28 是演奏合成部分和波形合成部分之间的信号流向图；

25 图 29 表示在高速缓冲存储器中页面标题的链接结构示意图；

图 30 是高速缓冲存储器内页面之间的状态过渡图；和  
图 31 是表示上述实施例之时间控制的时基图。

### 具体实施方式

#### 5 1. 按照本发明实施例所述的硬件结构

下面将参照附图更详细地描述本发明的实施例。

图 1 是表示本发明所述波形发生器硬件结构的范例方框图。该结构实例使用了一台计算机。在该结构中，计算机执行指定的程序（软件）以实现本发明所述的波形发生。当然，波形发生并不限于计算机软件的形式，  
10 还可以用由 DSP（数字信号处理器）处理的微程序的形式。此外，波形发生并不限于这些程序种类的形式。而是还可以采用的结构包含分离电路、集成电路、大规模集成电路等给定的硬件形式。波形发生器可以应用于各种产品，例如电子乐器、卡拉 OK 设备、电子游戏设备、其他多媒体装置、个人计算机等。

15 在图 1 所示的硬件结构实例中，CPU101 作为计算机的主要控制部分。通过总线 BL（数据或地址总线等），将 CPU101 与只读存储器（ROM）102、随机存取存储器（RAM）103、面板开关 104、面板显示器 105、驱动器 106、波形输入部分 107、波形输出部分 108、硬盘 109、和通信接口 111 相连。CPU101 根据指定的程序完成例如“建立波形数据库”、“乐音合成（以  
20 所建数据库为基础的软声源）”等处理（下面将进行描述）。这些程序由网络通过通信接口 111 或例如 CD、MO 等安装在驱动器 106 上的外部存储媒体 106A 提供并存储在硬盘 109 中。在执行时，由硬盘 109 将程序加载到 RAM103 上。此外，可以优选将程序记录在 ROM102 中。ROM102 存储由 CPU101 执行或参照各种程序、数据等。RAM103 作为工作存储器使用，  
25 用于暂时存储有关音乐演奏的各种信息，或者暂时存储 CPU101 执行程序

- 时产生的各种数据。RAM103 还用作存储当前执行的程序或相关数据的存储器。赋予 RAM103 中的特定地址区域以相关的功能，并用作寄存器、标志、表格、存储器等。面板开关 104 的结构包括各种用于发布指令以便对乐音进行采样、对采样的波形数据等进行编辑、和输入各种信息等的运行装置。运行装置包括用于输入数字数据的数字键座、用于输入字符数据的键盘、面板开关等。此外，最好还包括各种用于选择、设定或控制定音、音调、效果等的运行装置。面板显示器 105 显示从面板开关 104 输入的各种信息或采样波形数据等。例如，用液晶显示器（LCD）、CRT 等作为面板显示器 150。
- 10 波形输入部分 107 包括用于将模拟乐音信号（外部波形输入信号，例如，从麦克风输入的信号）转换（采样）成数字式数据的 A/D 转换器。把数字波形数据输入到 RAM103 或硬盘 109 中作为原始波形数据（该波形数据作为要产生的波形数据的资料）。CPU101 进行“建立波形数据库”的处理以便根据输入的原始波形数据建立按照本发明所述的“波形数据库”。
- 15 此外，CPU101 进行“以数据库为基础的乐音合成”的处理，以便利用“波形数据库”形成与音乐演奏信息相对应的适合任何乐音信号的波形数据。当然，可以同时产生多个乐音信号。通过总线 BL 将适合产生乐音信号的波形数据送到波形输出部分 108 并适当缓冲。波形输出部分 108 根据指定的输出采样频率输出缓冲的波形数据，对数据进行数—模转换，并将其发送到发声系统 108A。最后，发声系统 108A 使得从波形输入部分 108 输出的乐音信号发出声音。硬盘 109 存储多种与音乐演奏有关的数据，例如波形数据、使对应于音乐演奏的使波形同步的数据（即后面将描述的演奏风格表、代码本等的数据）、由各种音调参数构成的音调数据等。硬盘 109 还存储由 CPU101 执行的用于控制各种程序等的的数据。
- 20 驱动器 106 驱动可拆盘（外存储媒体 106A），该可拆卸盘可存储多种
- 25

与音乐演奏相关的数据，例如，波形数据、使与音乐演奏对应的使波形同步的数据（后面将要描述的演奏风格表、代码本等的的数据）、由各种音调参数构成的音调数据等，以及由 CPU101 执行的用于控制各种程序等的的数据。受驱动器 106 驱动媒体 106A 不仅可以是软盘（FD），还可以是高  
5 密盘（CD-ROM 或 CD-RAM），磁光盘（MO），DVD（数字式通用盘），和其他类型的可拆卸媒体。当将存储控制程序的外部存储媒体 106A 安装到驱动器 106 上时，最好将其内容（控制程序）直接加载到 RAM103 中，而不必将其复制到硬盘 109 上。因为能容易地添加控制程序或使其升级，使用外部存储媒体 106A 或通过网络能很容易地提供控制程序。

10 通信接口 111 连接到通信网络（图中未示），例如 LAN 网、因特网、电话线等。通信接口通过通信网络与服务性计算机（图中未示）相连，以便将控制程序、各种数据或音乐演奏信息从服务性计算机等输入到波形发生器。当 ROM102 或硬盘 109 未存储控制程序或数据时，用通信接口 111 从服务性计算机上下载控制程序或各种数据。随同客户一起工作的波形发  
15 生器通过通信接口将命令发送到服务性计算机上，以便下载控制程序或各种数据。当接收该命令时，服务性计算机通过通信接口将请求的控制程序或数据存储于硬盘 109 中，从而完成下载。此外，所述接口最好包括用于接收 MIDI 音乐演奏信息的 MIDI 接口。显然，音乐演奏键盘或演奏操作装置可以与总线 BL 相接，以便实时提供音乐演奏信息。此外，最好使用  
20 存储了所需歌曲音乐演奏信息的外部存储媒体 106A 来提供音乐演奏信息。

## 2. 波形数据库建立程序

### 2.1 程序的概述

图 2 是表示在上述波形发生器中完成的“波形数据库建立程序”的实  
25 施例流程图。为了适应各种演奏（或发音），该程序将以用各种演奏风格

(或发音)演奏的声波为基础建立矢量数据。

在步骤 S1, 程序将准备用于存储演奏风格表和代码本(下面将提及)的数据库。例如, 用硬盘 109 作为数据库的媒体。所述程序根据常规乐器的各种演奏模式收集波形数据(步骤 S2)。即, 通过波形输入部分 10 7  
5 从外部的波形输入信号(例如麦克风)中获取各种常规乐器的实际演奏声音。程序把适合这些演奏声音的波形数据存储到硬盘 109 上指定的区域中。待输入的演奏声音的波形数据可以包括所有的音乐演奏或其特征段, 例如给定的分句、一个音符、一个硬起音部分或释放部分。

因此, 波形数据是根据常规乐器的各种演奏模式特征从演奏声音中获  
10 得的。然后, 程序将该波形数据分段成特定部分, 将其定调和为其起文件名(步骤 S3)。即, 程序将把输入的原始波形数据分成代表波形变化的单个波形(例如, 硬起音波形, 主体部分的波形, 释放部分的波形, 接合部分的波形等)([1]分段)。由程序确定在一个或多个循环中每个独立的波形数据的定音([2]定调)。由程序还为每个独立的波形数据指定文件  
15 名([3]起文件名)。当程序输入部分演奏波形数据例如硬起音部分或释放部分时可以省略波形分离([1]分段)。

然后程序借助频率分析分离各分量(步骤 S4)。即, 程序通过 FFT(快速傅里叶变换)分析把在步骤 S3 中分别产生的一部分波形数据分成多个分量。在该实施例中, 波形数据分成谐波(harmonic)分量和非谐波  
20 (non-harmonic)分量。程序还从每个分量(谐波分量, 非谐波分量等)中抽取适合各要素的特征, 例如波形、定音和振幅。通常, 这意味着特征分离。(当将波形数据分成谐波分量和非谐波分量时, 对非谐波分量来说, 因为其没有定音, 所以不需要定音分离)。例如, “波形”(音调)要素是一个抽取的特性, 它仅包括具有正常定音和振幅的波形形状。“定音”  
25 要素包括抽取的定音相对于参考定音的变化特性。“振幅”要素包括抽取

的振幅包络特性。

在步骤 S5，建立矢量数据。即，程序间断地或按需要可以连续地从各要素抽取多个采样值，所述要素可以是例如在分离的谐波分量、非谐波分量中的波形（音调）、定音和振幅等。用唯一矢量（识别信息）指定相关采样值顺序，并将其与采样值的时间位置数据一起存在代码本中。这样的采样数据在下文中将称为矢量数据。该实施例建立了波形（音调）要素和振幅要素在非谐波分量中的矢量数据。每个分量要素的矢量数据可随着沿时间轴的进程而改变。程序随后将建立演奏风格模式（下面将详细说明）的数据，并将演奏风格模式存储在演奏风格表中。由此，将已建立的演奏风格模式和矢量数据写到数据库中的演奏风格表和代码本上（步骤 S6），从而将数据存到数据库内。如上所述，矢量数据本身并不是原始的波形数据，而是代表输入的原始波形形状的波形中的各要素的独立数据。这个矢量数据最终成为演奏风格模式的单位数据。代码本存储经过压缩的并代表抽取波形形状变化的一部分波形数据。演奏风格表存储演奏风格模式数据（将在下文中详细说明）。即，该包把以压缩形式存储的矢量数据存储成具有波形形状的原始波形数据，和用来指定存储在代码本中的矢量数据之 ID 数据所需的各种数据。

在上述特征分离过程中（步骤 S4），程序还使用除振幅、定音和波形要素之外的时间要素来进行特征抽取。在下文中将抽取时间要素的矢量数据称为“时间矢量数据”。分别产生的分音波形数据的时间周期中的该时间要素用作原始波形数据的时间长度。因此，如果用系数“1”表示时间周期的原始时间长度（可变的），则“波形数据库建立程序”不需要分析或测量该时间长度。在这种情况下，代码本不必一定要存储时间要素的数据（即，时间矢量数据），这是因为在任何时间周期里总是取相同的值“1”。当然，本发明并不限于此。也可以采用分析和测量实际时间长度的实例，

并将其作为“时间矢量数据”存储在代码本中。

- 然后，判断数据库是否已完全建立（步骤 S7）。即，通过大量收集从外部波形输入信号获得的常规乐器不同演奏模式的演奏声音原始波形数据，该程序能据此判断是否已完全获得各种演奏风格模式的数据和矢量数据。该判断不总是自动进行的。最好跟随以下指令进行操作，该指令为：是否能根据用户的转换操作继续进行处理。当判断结果是已经收集到原始波形数据和以此为基础的矢量数据已完全建立时，程序将终止（步骤 S7 中的 YES）。为了继续收集原始波形数据和建立以原始波形数据为基础的矢量数据（步骤 S7 中的 NO），程序将转向步骤 S2 并重复步骤 S2—S7。
- 5
- 10 最好实际使用为判断是否已完全建立矢量数据（步骤 S7）而建立的矢量数据来产生乐音。即，在断定矢量数据完全建立（步骤 S7 中的 YES）的情况下，随后从图 2 的流程中退出控制。此后，可能出现通过使用所建立的矢量数据播放音乐的情况，但这不能令人满意。在这种情况下，优选在步骤 S2 之中和之后进行处理来添加矢量数据。即，不管何时，只要需要
- 15 便建立矢量数据和将其添加到数据库中。

在上述“波形数据库建立程序”中，优选设置添加或删除任何演奏风格模式或编辑演奏风格模式数据等的功能。

## 2.2 演奏风格模式数据结构

下面将具体描述演奏风格模式数据。

- 20 演奏风格模式存储在演奏风格表中，所述演奏风格表构成硬盘 102 上的数据库。将演奏风格 ID 和演奏参数相结合可以指定一种演奏风格模式。演奏风格 ID 包含乐器信息和模式部分的名称。例如演奏风格 ID 的定义如下。当用 32 位（0—31 位）的序列描述一种演奏风格 ID 时，用 6 位表示乐器信息。例如，当 6 位序列是“000000”时，乐器信息表示中音
- 25 萨克斯。当序列是“001000”时，乐器信息表示小提琴。优选使用三个

高等级位的 6 位序列对乐器类型进行粗分类, 和使用三个低等级位对乐器进行细分类, 例如用 32 位序列中的另 6 位表示模式部分的名称。例如, 当模式部分的名称表示正常硬起音 (NormalAttack) 时, 用 6 位序列 “000000”; 转弯起音 (BendAttack) 用 “000001”; 装饰音音符起音 (GraceNoteAttack) 用 “000010”; 正常短音主体 (NormalShortBody) 为 “001000”; 颤音主体 (VibBody) 用 “001001”; 正常长音主体 (NormalLongBody) 用 “001010”; 正常释放 (NormalRelease) 用 “010000”; 正常接合 (NormalJoint) 用 “011000”; 或装饰音音符接合 (GraceNoteJoint) 用 “011001”。当然, 模式部分的名称并不限于这一结构。

如上所述, 可利用演奏风格 ID 和演奏参数的结合来确定每种演奏风格模式。即, 演奏风格模式是根据演奏风格 ID 确定的。按照演奏参数变化地设定演奏模式的内容。演奏参数具有与演奏风格模式相应的波形数据特性, 或者可控制所述与演奏风格模式相应的波形数据。特定类型的演奏参数对每个演奏风格模式都是有效的。例如, 可以赋予中音萨克斯正常硬起音 (NormalAttack) 多种演奏参数, 例如紧跟在硬起音之后的独立定音或音量等。中音萨克斯上弯起音 (BendUpAttack) 模式可以设置多种演奏参数, 例如在上弯起音 (BendUpAttack) 结束时的独立定音, 上弯起音 (BendUpAttack) 时刻转弯深度的初始值, 从上弯起音 (BendUpAttack) 开始到结束的时间周期 (加音符的时间), 在硬起音之后瞬间的音量, 在上弯起音 (BendUpAttack) 期间默认曲线的瞬时伸展和收缩等。此外, 中音萨克斯正常短音主体 (NormalShortBody) 模式可以用该模式的独立定音, 正常短音主体 (NormalShortBody) 的停止时间和开始时间, 正常短音主体 (NormalShortBody) 开始或结束时的力度等来表示。演奏风格模式并不需要包括与所有有效演奏参数相应的数据。

演奏风格模式可以存储与某些断续演奏参数值相应的数据。例如，中音萨克斯正常硬起音（NormalAttack）模式可以存储与部分数据相应的数据，而不是所有绝对定音值或在硬起音之后的瞬间音量值。

5 如上所述，可以用演奏风格 ID 和演奏参数来确定演奏风格模式。因此，例如，中音萨克斯正常硬起音（NormalAttack）模式可以确定除了表示萨克斯管正常硬起音的多个数据（后面将要描述的基本数据）之外的与所需演奏参数相应的数据。提琴转弯起音（Bendattack）模式可以确定除表示提琴转弯的硬起音部分多个数据（下面将描述的基本数据）之外的与所需演奏参数相对应的数据。

### 10 2.3 演奏风格表数据结构

演奏风格表根据每个演奏风格模式，存储和产生对应于每个演奏风格模式的相关演奏风格模式波形所需的数据。例如，演奏风格表存储矢量 ID 和典型点值序列（用于确定典型采样点的数据，以便修正多个采样序列，）以便确定每个分量要素的矢量数据（波形要素、定音要素（定音包络线）、  
15 振幅要素（振幅包络线）等）。或者，演奏风格表存储每个分量要素的信息，这些信息例如矢量数据（波形要素、定音要素（定音包络线），振幅要素（振幅包络线））的开始或停止时间位置。即，演奏风格表存储各种以正常形状播放波形所需的数据（下文中也称作“要素数据”），所述波形来自数据库中以压缩形式作为矢量数据存储的波形。下面的描述用中音  
20 萨克斯正常起音（NormalAttack）模式来解释存储在与演奏风格模式相应的演奏风格表中的特定数据实例。

数据 1：演奏风格模式的采样长度。

数据 2：加音符的时间位置。

数据 3：在谐波分量中振幅要素的矢量 ID 和典型点值序列。

25 数据 4：在谐波分量中定音要素的矢量 ID 和典型点值序列。

数据 5: 在谐波分量中波形 (音调) 要素的矢量 ID。

数据 6: 在非谐波分量中振幅要素的矢量 ID 和典型点值序列。

数据 7: 在非谐波分量中波形 (音调) 要素的矢量 ID。

数据 8: 在谐波分量中波形 (音调) 要素块 (block) 的开始位置。

5 数据 9: 在谐波分量中波形 (音调) 要素块的结束位置 (在谐波分量中波形 (音调) 要素环 (loop) 的开始位置)。

数据 10: 在非谐波分量中波形 (音调) 要素块的开始位置。

数据 11: 在非谐波分量中波形 (音调) 要素块的结束位置 (在非谐波分量中波形 (音调) 要素环的开始位置)。

10 数据 12: 在非谐波分量中波形 (音调) 要素环的结束位置。

下面将参照图 3 描述上述数据 1—12。

图 3 示出了构成与演奏风格模式对应的实际波形部分的分量和要素的范例。在该图中, 从顶部到底部, 示出了在谐波分量中的振幅要素, 谐波分量中的定音要素, 谐波分量中的波形 (音调) 要素, 非谐波分量中的  
15 振幅要素, 和非谐波分量中的波形 (音调) 要素。图中的参考标记对应于上述数据号。

参考标记 1 代表与演奏风格模式对应的波形采样长度 (波期)。例如, 采样长度对应于以作为演奏风格模式基础的原始波形数据的整个时间长度。参考标记 2 表示加音符的时间位置, 在演奏风格模式中的任何时间位置  
20 上均能以可变的形式确定该加音符的时间位置。原则上, 与波形对应的演奏声音从这个加音符的时间位置开始发声。例如, 根据转调硬起音等演奏风格, 波形分量的上升时间可以先于加音符的时间。

参考标记 3 表示的矢量 ID 和典型点值序列显示出了以谐波分量的形式存储在代码本内的振幅要素之矢量数据。在图 3 中, 典型点用两个黑色  
25 方块表示。参考标记 4 表示的矢量 ID 和典型点值序列示出了谐波分量中

定音要素之矢量数据。参考标记 6 表示的矢量 ID 和典型点值序列显示出了非谐波分量中的振幅要素之矢量数据。用典型点值序列数据改变或控制由矢量 ID 指定的矢量数据（包括多个采样序列）和指示或指定某些常规采样点。根据指定的常规采样点，可以改变和校正它的时间位置（横轴）和水平轴（纵轴）。由此，可使其它采样点发生变化，并因此改变矢量形状。例如，常规的点值序列数据表示比所述采样量少的分散采样量，但并不限于此。常规的点值序列数据可以是采样之间的间断数据或是指定范围内的数据（多个连续的采样）。此外，常规的点值序列数据可以是差值、倍数等，而不是采样值本身。可以通过使该典型点沿着横轴和/或纵轴（时间轴）移位来改变每个矢量数据形状。即，可以改变包络波形形状。

参考标记 5 表示的矢量 ID 示出了谐波分量中波形（音量）要素的矢量数据。参考标记 7 表示的矢量显示出了非谐波分量中波形（音调）要素的矢量数据。参考标记 8 表示的是谐波分量中波形（音调）要素块的开始位置。参考标记 9 表示谐波分量中波形（音调）要素块的结束位置（或谐波分量中波形（音调）要素环的开始位置）。即，从参考标记 8 开始的三角形表示连续存储特性波形状的非环状波形部分。从参考标记 9 开始的连续矩形表示能反复读取的环状波形部分。非环状波形是具有演奏风格（或发音）等特性的高质量波形。环状波形是与比较单调的声音部分相对应的单位波形，而且它包括一个周期或适量周期的波形。参考标记 10 表示非谐波分量中波形（音调）要素块的开始位置。参考标记 11 表示非谐波分量中波形（音调）要素块的结束位置（或非谐波分量中波形（音调）要素环的开始位置）。参考标记 12 表示非谐波分量中波形（音调）要素环的结束位置。用上述数据 3—7 作为指示存储在每个分量要素代码本中的矢量数据的信息。用上述数据 2 和 8—12 作为从矢量数据恢复原始波形（在分离之前）的时间信息。

如上所述，演奏风格模式包括用于指示矢量数据和时间信息数据的数据。通过使用存储在演奏风格表中的演奏风格模式数据，和使用存储在代码本中的波形资料（矢量数据），可以自由地构筑波形。即，演奏风格模式是代表波形状态的数据，该波形可根据演奏风格（或发音）产生。各演奏风格模式可以包含不同的数据种类或数据量。除了上述数据外，还可以在演奏风格模式上设置其它信息等，例如，用于控制波形时间轴扩展和压缩的数据。

为了容易地理解所述解释，上述实例示出的一种演奏风格模式包括谐波分量中的所有要素（波形、定音、振幅）和非谐波分量中的要素（波形和振幅）。然而，本发明并不限于此。显然，演奏风格模式可以包括谐波分量中多种要素（波形、定音、振幅）的一种或非谐波分量中多种要素（波形和振幅）中的一种。例如，演奏风格模式可以包括谐波中的波形（音调）要素、谐波分量中的定音要素、谐波分量中的振幅要素、非谐波分量中的波形（音调）要素、非谐波分量中的振幅要素中的一种。由于对每种分量而言，演奏风格模式可以自由组结，所以很方便。

如上所述，按照各种常规乐器的不同演奏模式，所有波形数据并不保持演奏声音。波形数据库建立程序仅抽取改变波形形状所需的分音波形（例如，硬起音部分的波形，主体部分的波形，释放部分的波形，接合部分的波形等）。此外，按照分量、要素和典型点使用分层压缩技术存储波形数据作为硬盘 109 中的压缩数据。可以保存存储波形数据所需的硬盘 109 的存储容量。

### 3. 波形合成过程

图 1 中的波形发生器，允许计算机执行指定程序（软件）完成本发明所述波形合成过程来合成波形。图 4 是执行上述波形合成过程（“以数据库为基础的音乐合成程序”）的程序流程图。

然而，在该实施例中，由程序完成的各种功能彼此是独立的。下面将描述以图 5 所示功能方框图为基础的操作。本发明不限于这种程序。波形合成过程是在专用硬件设备上进行的。在这种情况下，图 5 是专用硬件设备的方框图，所述硬件设备能提供与图 4 中所示相同的波形合成过程。以下主要参照图 5 进行描述并且用括号表示图 4 中相应的步骤。

### 3.1 声音数据播放部分 101A

声音数据播放部分 101A 播放带有演奏符号的声音数据（步骤 S11）。首先，声音数据播放部分接收带有演奏符号的声音数据（音乐演奏信息）。原始乐谱包括音乐标记例如动态标记（渐强，渐弱等）、速度标记（快板，慢板等）、连线、持续、重音等。这些标记如果不改变的话，不会成为 MIDI 数据。为此，需将这些标记转换成“演奏符号”数据。将包含演奏符号的 MIDI 声音数据称为“带有演奏符号的歌曲数据”。演奏符号包括图表 ID 和图表参数。图表 ID 表示乐谱上的音乐标记。图表参数表示用图表 ID 指出的音乐标记的内容范围。例如当图表表示“颤音”时，将提供图表参数来指定颤音的速度、深度等。当图表 ID 表示“渐强”时，将提供图表参数来指定在渐强开始或结束时的音量、音量变化的时间长度等。

### 3.2 乐谱注释部分 101B

乐谱注释部分（演奏者）101B 完成乐谱注释过程（步骤 S12）。特别是，该部分将 MIDI 数据和上述包含在声音数据中的“演奏符号”（图表 ID 和图表参数）转换成演奏风格指定信息（演奏风格 ID 和演奏参数）。经转换的数据与时间信息一起输出到演奏合成部分（发音器）101C。通常，演奏者可以用相同的音乐标记进行不同的注释。他们可以用不同的演奏风格或发音进行演奏。此外，演奏者可以根据不同的演奏风格表演一系列的音符。乐谱注释部分是一个可了解表现乐谱标记（音乐标记、音符的分布等）的专家系统。

例如，乐谱注释部分使用以下标准来注释乐谱上的标记。颤音只可用于八分音符或更长的音符。断奏实际上增加了动感。持续的长短决定着音符的衰减。连奏并不减弱音符。时间值决定着八分音符颤音的速度。动态取决于定音。此外，存在不同注释标准，例如因提高或降低短句内的定音和减小动态的 dB 线性增益而引起的动态变化，和与持续或断奏相应的音符长度以及与硬起音部分的上弯标记相应的上弯宽度和曲线变化。根据这些标准，乐谱注释部分 101B 注释乐谱以把乐谱转换成音乐演奏数据。此外，乐谱注释部分 101B 根据用户的表演者指令，即，表演者进行演奏（演奏风格）的指令，换行上述乐谱注释过程。乐谱注释部分 101B 根据该表演者指令，通过改变注释方法来注释乐谱。例如，数据库内存储根据多个演奏者得到的不同的乐谱注释方法。乐谱注释部分 101B 根据用户的演奏者指令，通过选择不同的乐谱注释方法来注释乐谱。

使构成的歌曲数据（音乐演奏信息）预先包括显示乐谱注释结果的数据。显然，在输入的声音数据包含预先注释的乐谱结果数据时，将不再需要上述过程。通过适当的用户输入操作由乐谱注释部分 101B 完全自动地或半自动地进行注释乐谱（步骤 S12）。

现在参照图 21，下面将描述在乐谱注释部分 101B 建立的音乐演奏数据的内容。

象常规的 SMF（标准 MIDI 格式）一样，音乐演奏包括标题和多个音轨。每个音轨包含程序和事件数据（添加音符，除去音符等）。在该实施例中，每个音轨包含，其用于指定硬起音部分、主体部分、接合部分和释放部分模式的模式指定部分。模式指定部分是包含演奏风格 ID 和演奏参数的演奏风格指定信息。实际上，这些模式指定部分采用了未定义的系统专用信息、元事件、14 位控制变化等。此外，每个音轨包含时间差数据，该数据用于指定事件数据或模式指定部分之间的时间差。

可能存在这样一种情况，即每个音轨包含指定音调的程序变化，也即新数据库（演奏风格表和代码本）。在这种情况下，乐谱注释部分 101B 向波形合成部分 101D 提供作为预测数据的程序变化。由于波形合部分 101D 从代码本中读取下一次预测将用到的矢量数据，所以需确定程序变化，以便将预测范围限制在某种程度上（下面将详细说明）。预测数据还包含其它各种数据。根据歌曲数据，乐谱注释部分 101B 向波形合成部分 101D 提供“指示后续演奏风格指令的数据”作为预测数据。以上所述包含的程序变化是一种预测数据。

### 3.3 演奏合成部分 101C

演奏合成部分（发音器）101C 在由乐谱注释部分（演奏者）101B 转换的演奏风格指令（演奏风格 ID 和演奏参数）的基础上参照演奏风格表。该部分产生与演奏风格指令（演奏风格 ID 和演奏参数）相应的包流（也称为矢量流）和与演奏参数对应的数据流有关的矢量参数，并且将所述包流和矢量参数提供给波形合成部分 101D（步骤 S13）。提供给波形合成部分 101D 的数据是包含包时间信息、矢量 ID、典型点值序列等与定音要素和振幅要素有关的包流。就波形（音调）要素而言，所述包含矢量 ID、时间信息等（下面将详细说明）。

波形合成部分 101D 根据包流从代码本中检索矢量数据，按照矢量参数修正矢量数据，并根据修正的矢量数据合成波形（步骤 S14）。然后，波形合成部分 101D 完成其他部分的波形生成过程（步骤 S15）。多个演奏部分中的其他部分是指那些不进行演奏合成过程而是采用常规乐音波形合成过程的演奏部分。例如，用常规波形存储器声源系统为这些其他部分产生乐音。这种“其他部分的波形产生过程”可以在专用硬件声源（外部发声设备或可安装到计算机上的声卡）上完成。为了便于说明，该实施例仅使用了一个部分来产生与演奏风格（或发音）相应的乐音。当然，也

可以用多个部分来播放演奏风格。

图 6 是用于表示上述演奏合成部分 101C 的演奏合成过程之流程的方框图。尽管图 6 中示出的是分别存储演奏风格模式和代码本的情形，但是实际上也可以将两者都存在硬盘 109 上的数据库内。

- 5 演奏合成部分 101C 建立了根据演奏风格指令（演奏风格 ID 和演奏参数）和来自乐谱注释部分 101B 的时间信息数据供给波形合成部分 101D 的不同的包流。演奏合成部分 101C 并不使用针对每个音调的永久演奏风格模式。用户可以在当前使用的演奏风格模式上添加新的演奏风格模式或停止使用它们当中的某些模式。演奏合成部分 101C 还执行建立校正信息
- 10 以便校正选定要素数据和演奏参数值之差的程序，和平滑前后演奏风格模式的连接波形特征的平滑连接程序。

- 通常，乐谱注释部分 101B 向演奏合成部分 101C 提供数据，但是本发明并不限于此。如上所述，最好是根据演奏风格准备歌曲数据，其中所述歌曲数据已经由乐谱注释部分 101B 注释。或者，根据演奏风格准备歌曲
- 15 数据，其中用户通过注释乐谱为歌曲数据提供演奏风格 ID 和演奏参数。通过播放歌曲数据向演奏合成部分 101C 提供数据。

图 7 是详细表示演奏合成过程的实施例流程图。

- 演奏合成部分 101C 根据演奏风格 ID 和演奏参数从演奏风格表中选择演奏风格模式（步骤 S21）。即，演奏合成部分 101C 根据从乐谱注释部
- 20 分 101B 发出的演奏风格 ID（乐器信息和模式部分名称）和演奏参数选择一个演奏风格模式。在注释乐谱之前，乐谱注释部分 101B 检查数据库以便确定在与乐器信息指示的音调相应的演奏风格表中存在哪些模式部分。乐谱注释部分 101B 指定现存部分范围内的演奏风格 ID。当指定的是非现存部分时，优选代之以选择具有类似特征的演奏风格 ID。然后，演奏合
- 25 成部分 101C 选择与指定的演奏风格 ID 和演奏参数相应的多个要素数据

(步骤 S22)。即，演奏合成部分 101C 通过参照演奏风格表并根据指定的演奏风格 ID 和演奏参数来指定演奏风格模式，并且从所述模式中择出多个与演奏参数对应的要素数据。当演奏风格模式不包含与演奏参数完全匹配的要素数据时，演奏合成部分 101C 选择与和演奏参数相近的值对应的要素数据。

然后，该程序根据时间信息计算要素数据中每个位置的时间（步骤 S23）。即，根据时间信息将每个要素数据布置在绝对时间位置上。特别是，该程序根据来自指示每个相对的时间位置的要素数据的时间信息计算绝对时间。因此，由程序来确定每个要素数据的时间（见图 3）。此外，

10 程序根据演奏参数来校正每个要素数据的值（步骤 S24）。即，用程序校正选定的要素数据和演奏参数之间的偏差。例如，乐谱注释部分 101B 在硬起音音量（演奏参数）刚设定为“95”后发送中音萨克斯正常起音 [NormalAttack] 模式。当演奏风格表内包含在硬起音音量刚设定为“100”后的中音萨克斯正常起音 [NormalAttack] 模式时，演奏合成部分 101C 选

15 择在硬起音音量刚设定为“100”后的中音萨克斯正常起音 [NormalAttack] 模式的要素数据。然而，在硬起音后的瞬间音量保持在“100”。通过对选定要素数据的典型点修正，可以将硬起音之后瞬间的音量修正到“95”。进行该修正可以使选定的要素数据值近似于演奏参数值。此外，可以按照预定的微调值（乐器调音）进行修正，和按照乐器的音量变化特性进行音

20 量修正等。这些修正可以通过改变典型点值或各要素数据来实现。典型点值可以有很大变化。即，典型点是必需的和足以进行修正的数据。通过控制所述典型点，可以进行各种修正。

上述步骤 S23 通过使用例如演奏参数等修正信息，修正由时间信息指示的时间位置。例如，根据音乐演奏数据得出的时间位置和用上述时间信息指示的时间位置之间可能有差异。在这种情况下，程序选择的时间信息

25

所指示的时间位置应近似于根据音乐演奏数据得出的时间位置。程序根据音乐演奏数据修正所得到的时间位置信息。这样便可以得到所希望的音乐演奏数据的时间位置信息。此外，音乐演奏数据可以包含如触摸或速度等可变的控制因素。在这种情况下，可以根据音乐演奏数据并通过根据可变控制因素修正时间位置信息而对时间位置信息进行可变控制。修正信息包括该时间位置的修正信息。

此外，调节每个要素数据以便进行相邻演奏风格模式之间平滑连接的连接程序（步骤 S25）。即，在连接前后演奏风格模式时紧密连接各典型点，而使前后演奏风格模式的波形特性平滑的程序。对于谐波分量中的波形（音调）、振幅、和定音各要素或非谐波分量中的波形（音调）和振幅各要素而言，分别进行这种连接或链接程序。

这时，从前一演奏风格模式中的“链接开始”到后一演奏风格模式中的“链接结束”进行调节。即，该程序根据“近似率”在从链接开始”到“链接结束”的范围内调节典型点。所述“近似率”是控制近似量的参数，它是指在前后演奏风格模式之间形成连接时从它们中的一个得出的近似量。正如后面将要描述的那样，近似率取决于前后演奏风格模式的结合情况。当前后演奏风格模式相连时，波形不会很好地连接。在这种情况下，使来自前后演奏风格模式中任一模式的波形特性的矢量 ID 变薄，来实现平滑连接。为了实现这一跳跃，而设置了“演奏风格模式结合表”、作为参照的“跳跃执行参数范围表”和作为参照的“跳跃时间表”。

此外，借助于乐谱注释部分 101B 中的以下链接程序可以将波形特征平滑地连接。例如，所述程序能不考虑演奏风格模式，平滑地连接演奏参数中的间断部分（动态值、定音参数值等）。此外，在从颤音移到释放的情况下，通常极早地将颤音减半音便可以实现平滑连接。

下面将详细讨论上述链接程序。即，下面将简明地描述用于在前后演

奏风格模式之间进行平滑连接的每个要素数据的调节（参见步骤 S25）。参见图 8，下面将说明演奏风格模式与振幅要素或定音要素对应的情况的链接程序。

由于在前后演奏风格模式之间的链接程序中会出现间断的典型点值，所以在前后演奏风格模式之间的连接出现偏差。在这种情况下，需要首先确定作为指数的“近似率”，以确定前后演奏风格模式达到该值时连接点目标值应当近似。这时，动态连接点用作振幅，或定音连接点用作定音。在该实施例中，假设在图 8 所示的表格中给出了近似率。例如，当前面的演奏风格模式的矢量 ID 为“3”，而后面的演奏风格模式的矢量 ID 为“7”时，从表中得出的“近似率”是“30”。根据选定的“近似率”，从前一演奏风格模式的“链接开始”到“演奏风格模式结束”这一过程中的包络线形状逐渐转向目标值。此外，从后一演奏风格模式的“链接结束”至“演奏风格模式开始”这一过程中的包络线形状逐渐转向目标值。例如，当将“近似率”设定为“30”时，前一演奏风格模式的目标值是“30”。前一演奏风格模式近似于后一演奏风格模式的“30%”。（在该实施例中，前一演奏风格模式中的最后一个典型点下移了接近“30%”）。

相反，前一演奏风格模式近似于后一演奏模式的“70%”（ $100\% - 30\%$ ）。（在该实施例中，后一演奏风格模式中的第一典型点上移了接近“70%”）。从链接开始到链接结束，有多个典型点存在于前后演奏风格模式内，并且近似上移和下移上述近似值。如上所述，在前后演奏风格模式中的多个典型点上出现近似值。可以适当设定链接的开始和结束。优选的是，将链接的开始或结束设在与所期望的典型点相同的点上，以消除如图中所示在链接开始或结束时包络线形状出现弯形。即使是当没有将链接开始或结束设在与所希望的典型点相同的点上时，也显然允许近似，这样不会在包络线形状中出现弯形。

确定“近似率”并不限于上述实例所述方法。例如，可以根据在连接点之前和之后给出的演奏参数来确定近似率。此外，可以根据产生演奏风格 ID 或演奏参数之前的音乐演奏数据来确定近似率。另外，可以根据这些数据的结合来确定近似率。上述实例采用了以“近似率”为基础的单一近似典型点。根据该近似量，其它典型点也近似适当量。最好对多个典型点中的每个点确定各自的“近似率”并且根据每个“近似率”使多个典型点近似。

下面，针对演奏风格模式是波形（音调）要素的情况描述其链接程序。图 9—12 是表示当演奏风格模式是波形（音调）要素时链接程序的示意图。图 9 是用于说明当将硬起音部分的波形连接到主体部分波形上时波形变薄的示意图。图 10 是说明当将主体部分的波形连接到释放部分的波形上时波形变薄的示意图。在图 9 中，主体部分的波形包括 5 个环状波形 L1 到 L5，每个波形在特定的时间范围内以环状的形式播放。同样，图 10 中主体部分的波形包括六个环状波形 L1' 到 L6'。

有各种方法调节与波形有关（例如波形连接过程）的要素数据。例如，局部减薄波形以便平滑连接例如硬起音部分或接合部分的演奏风格模式和主体部分的演奏风格模式（或主体部分与释放部分或接合部分的演奏风格模式之间的连接）的方法。对于波形之间的连接来说，平滑转换是众所周知的。然而，如图 9 中所示，当时间  $t$  离连接环状波形 L1 的起始时间点较短时，需要在较短的时间  $t$  内快速进行平滑转换。即，快速的平滑转换表示在要连接的波形之间的极短时间内进行的平滑转换。进行这样的快速平滑转换会导致产生很大噪音的波形。这是不可取的。

为了避免快速的波形平滑转换，可以跳过（删除）一部分波形以加大要连接的波形之间的时间间隔。在这种情况下，硬起音部分、释放部分或接合部分的波形将构成一个方块，以防止波形跳越。因此，环状波形从主

体部分跳过。在图 9 和 10 中，环状波形 L1 和 L6' 被跳过。这些波形用黑色方块表示。例如，在图 9 中，完成的是与连接时间点具有较长时间差的第二环状波形 L2 和硬起音部分波形中最后一个波形之间的平滑转换。没有使用第一环状波 L1。同样，在图 10 中，完成的是环状波 L5' 和释放部分波形之间的平滑转换，其中没有使用环状波 L6'。

应该注意到，接合部分采用的是在任何演奏风格的声音之间（或声部之间）进行连接的波形区。

进行波形跳越以使硬起音部分的演奏风格模式和释放部分或接合部分的演奏风格模式之间形成平滑连接。图 11 和 12 是说明当硬起音部分的波形与释放部分的波形连接时，产生波形跳越的示意图。

在这种情况下，波形可以从硬起音部分或释放部分的演奏风格模式跳越也可以不跳越。转弯的硬起音部分是允许波形从相关演奏风格模式跳越的硬起音部分的实例。第二半转弯的硬起音部分包含几个环状波形。对于第一半包含几个环状波的释放部分，波形跳越也是可行的。波形从能产生波形跳越部分的演奏风格模式跳越。例如，当将转弯的硬起音部分连接到释放部分时，如图 11 所示，从转弯的硬起音部分一侧跳过环状波形。（图 11 表示从转弯的硬起音部分一侧跳过用黑色方块表示的一个环状波形）。当将正常的硬起音部分连接到具有环状波形的释放部分时，如图 12 所示，从释放部分一侧跳过环状波形。（图 12 表示从释放部分一侧跳过用黑色方块表示的一个环状波形）。

待跳越的环状波形实际上并不需要最靠近演奏风格模式之间的接合部。即，不必是第一个或最后一个环状波形。最好是根据给定的次序选择需从多个环状波形上跳过的环状波形。

如上所述，当在演奏风格模式的给定结合中不能在演奏风格模式的范围内进行成功连接时，便可进行波形跳越。为了实现所述跳越，而设置了

例如“演奏风格模式结合表”、作为参照的“跳越执行参数范围表”、和作为参照的“跳越时间表”。“演奏风格模式结合表”根据要连接的前后演奏风格模式的结合情况确定给定的参数。“跳越执行参数范围表”确定时间范围以便完成相对于上述每个参数的跳越。“跳越时间表”确定跳越时间。在连接时间点和第一个（或最后一个）环状波形 L1（或 L6'）之间存在时间差（图 9-12 中的时间  $t$ ）。当该时间差短于参考跳越时间时，将跳过相应的环状波形。

参照图 13, 下面将描述当演奏风格模式具有较短的采样时间而且当该演奏风格模式在启动之后先于另一个演奏风格模式结束时所进行的波形连接。在图 13 中, 波形（音调）要素的包流包括四个演奏风格模式, 即, 在图中从左到右按顺序排列的 A. 萨克斯上弯硬起音 [BendUpAttack], A. 萨克斯正常短音主体 [NormalShortBody], A. 萨克斯主体颤音 [VibratoBody], A. 萨克斯正常播放 [NormalRelease]。每一个演奏风格模式的采样时间（持续时间）用“长度”表示。在图 13 中, 上部所述“添加音符”和“断开音符”是 MIDI 数据的事件时间。在图的中部, A. 萨克斯上弯硬起音 [BendUpAttack] 等表示各演奏风格 ID 的产生时间。音符、跳动、深度等表示各演奏参数的产生时间。

A. 萨克斯上弯硬起音 [BendUpAttack] 模式从时间  $t_0$  开始。时间  $t_1$  是在该模式内添加音符的时间而且该时间应当对应于指定的添加音符的时间。根据演奏参数, 例如上述音符、跳动、深度等, 来控制模式包流的内容。A. 萨克斯正常短音主体 [NormalShortBody] 模式从硬起音模式之后紧接的时间  $t_2$  开始。时间  $t_3$  表示从接合部的中部开始颤音演奏风格的时间。该时间是根据例如设在歌曲数据中的颤音标记的开始时间确定的。时间  $t_5$  表示在 A. 萨克斯正常播放 [NormalRelease] 模式中断开音符的时间, 而且该时间应当对应于指定的断开音符时间。因此, A. 萨克斯正常播放

[NormalRelease]开始的时间  $t_4$  是指定的。

由于添加音符的事件发生在时间  $t_1$  而中断音符的事件发生在时间  $t_5$ ，所以，根据由包流产生的波形，实际的发声时间范围是从时间  $t_1$  至时间  $t_5$ 。在这种包流的情况下，从时间  $t_2$  到时间  $t_4$  的时间周期与 A. 萨克斯正常播放 [NormalRelease] 模式和 A. 萨克斯主体颤音 [VibratoBody] 模式的总采样时间常常不匹配。因此，需要适当的修正性测量。在这种情况下，通过反复运行相同的演奏风格模式，或通过改变演奏风格模式的采样时间或通过两种方法结合，将总的采样时间调整到上述时间周期。用这种方式，可以将模式调整到可进行波形连接。在上述实例中，反复运行 A. 萨克斯正常播放 [NormalShortBody] 模式从而使之与后面的 A. 萨克斯主体颤音 [VibratoBody] 模式进行波形连接。同样，反复运行 A. 萨克斯主体颤音 [VibratoBody] 模式，使之与后面的 A. 萨克斯正常播放 [NormalRelease] 模式进行波形连接。

如上所述，当反复运行演奏风格模式进行波形连接时，将为反复运行的演奏风格模式提供不同的时间长度。通过移动 A. 萨克斯正常短音主体 [NormalShortBody] 模式或 A. 萨克斯主体颤音 [VibratoBody] 模式中的典型点可以对时间长度进行可变控制。可以通过合适的方法，例如改变构成 A. 萨克斯正常短音主体 [NormalShortBody] 模式或 A. 萨克斯主体颤音 [VibratoBody] 模式的多个环状波形之间的平滑转换连接时间便可实现所述控制。通过设定可变的环数量和环的延续时间，环状波形能较容易地实现对整个环状播放波形时间长度的可变控制。另一方面，在非环状波形的情况下，并不能如此容易地可变控制其时间轴上现有的长度。因此，如上所述，提供可变控制以便在环形读取周期内按照由非环状波和环状波构成的声波顺序扩展和收缩波形数据的时间轴。在整个发声时间长度提供可变控制的发明对简单地控制临时扩展和收缩是非常有益的。为此，建议使

用早先由日本待审专利申请公开号平成 10-307586 的申请人提出的“时间的延长和压缩”控制（TSC 控制）。尤其是，该“TSC”控制是适用的，这是因为时间轴的长度是针对与特定演奏风格相应的非环状波形而变的。

图 14 示意性地示出了如此建立的包流的实例。在图 14 中，从顶部到底部示出了在谐波分量中振幅要素、定音要素、和波形（音调）要素等的包流，以及在非谐波分量中振幅要素和波形（音调）要素的包流。黑色方块表示谐波分量中振幅要素和定音要素的典型点以及非谐波分量中的振幅要素的典型点。连接这些点的曲线代表了由包含在包流之包内的矢量 ID 表示的矢量形状。白色的矩形 L 表示谐波分量和非谐波分量中波形（音调）要素的环状波形。另一个矩形 NL 表示非环状波形。阴影表示的非环状波形是特别有特点的一个波形。在该实施例中，用两个矢量来构筑正常起音（NormalAttack）模式中谐波分量和非谐波分量的每个波形（音调）要素。用一个矢量来构筑谐波分量的每个振幅要素和定音要素以及非谐波分量的振幅要素。

在该实施例中，振幅要素和定音要素在波形（音调）要素形成谐波分量或非谐波分量中非环状波形的部分不包含矢量。然而，优选通过在波形（音调）要素形成非环状波的部分向振幅要素和定音要素提供矢量来控制已产生的波形。在主体颤音（VibratoBody）模式中，用五个矢量来构筑谐波分量中的波形（音调）要素。用一个矢量构筑谐波分量中的每一个振幅要素和定音要素，以及非谐波分量中的波形（音调）要素和振幅要素。在此，将主体颤音 VibratoBody 重复三次。应当注意到，每一次重复使用不同的矢量形状。这是因为相对每次重复指定不同的演奏参数。按照不同的演奏参数，选择不同的要素数据，或完成不同程度的控制或时间轴的控制。在正常接合 NormalJoint 模式中，使用三个矢量来构筑谐波分量和非谐波分量中的每个波形（音调）要素。用两个矢量来构筑谐波分量中

每个振幅要素和定音要素，以及非谐波分量中的振幅要素。省略关于正常主体 NormalBody 模式的描述。

如上所述，演奏合成部分 101C 产生适合每个谐波分量和非谐波分量的包流。这些包流中的每一个都由多个包构成。每个包都含有包的矢量 ID 和时间信息。此外，所述包中含有在谐波分量中的振幅要素和定音要素以及非谐波分量中的振幅要素的情况下为每个典型点设定的数值。当然，本发明并不限于此。本发明优选的是提供除矢量 ID 和包的时间信息之外的其它信息。根据每个包的内容，为每个分量构筑包流。包流包含多个包和与每个包有关的时间信息（启动时间）。

10 显然，包流的数量等可以按照乐器的类型而不同。

### 3.4 波形合成部分 101D

#### 3.4.1 波形合成部分 101D 的所有操作

波形合成部分 101D 根据从演奏合成部分 101C 提供的每个分量的包流（包含矢量 ID、时间信息、修正信息等多个包的序列）进行波形合成。图 15 是用于解释波形合成部分 101D 内工作情况的完整结构示意图。图 16—19、22 和 23 详细地表示波形合成部分 101Dk 的每个操作。图 16 是简要表示波形合成所有流程的方框图。图 17 是用于解释矢量加载器的方框图。图 18 是用于解释矢量操作器的方框图。图 19 是用于解释矢量解码器的方框图。图 22 表示把包从演奏合成部分 101C 送到波形合成部分 101Dr 时间。图 23 是用于解释高速缓存控制部分 40 的方框图。

在图 15 中，演奏合成部分（发音器）101C 建立适合每个分量要素的包流。然后，对应于包基上的每个分量要素，将包流依次输入到设在波形合成部分 101D 内的指定包队列缓冲器 21—25 中（即，输入到包的单元中）。输入的包存储在包队列缓冲器 21—25 中并且按指定的顺序依次发送到矢量加载器 20 上。矢量加载器 20 使包中的矢量 ID 定位并通过高速

缓存控制部分 40 从代码本 26 中读取（加载）与所述矢量 ID 相应的原始矢量数据。将读出的矢量数据发送到指定的矢量解码器 31—35。这些矢量解码器是对应于每个分量要素设置的并且产生适合每个分量要素的波形。

- 5 此外，矢量解码器 31—35 对在矢量解码器 31—35 之间产生的适合每个分量元素的波形进行波形合成并产生适合每个分量（谐波分量和非谐波分量）的波形。将针对每个分量产生的波形发送到混合器 38。除了把包的输入输入到包队列缓冲器 21—25 之外，演奏合成部分（发音器）101C 进行诸如包流管理等各种控制和针对波形合成部分 101D 的播放控制。
- 10 包流管理涉及产生或删除单个矢量数据或在矢量数据之间进行连接。播放控制对产生理想波形、或播放或中断所产生的理想波形起着重要作用。

如上所述，顺序地向矢量加载器 20 提供构成包流的多个包，所述包流存储在包队列缓冲器 21 中。根据每个包中的矢量 ID，矢量加载器 20 通过高速缓存控制部分 40 从代码本 26 中读取与所述矢量 ID 相对应的矢

15 量数据。矢量加载器 20 随后将读取的矢量数据发送到矢量解码器 31（参照图 16）。此时，已读出的每个包可以包含修正信息（例如与典型点有关的修正信息）。在这种情况下，矢量加载器 20 根据修正信息改变读出的原始矢量数据。把经改变的矢量数据称之为不同于原始矢量数据的矢量信息数据。矢量加载器 20 把带有矢量信息数据的包以信息的形式输出到

20 矢量解码器 31—35。将所述包称为与从演奏合成部分 101C 输入的包不同的矢量包。用这种方式，矢量加载器 20 根据演奏合成部分（发音器）101C 输入的包矢量 ID 从代码本 26 中读取原始矢量数据。需要时，矢量加载器 20 根据修正信息修正矢量数据并且将矢量包传送到矢量解码器 31—35

（参见图 17）。存在各种与矢量数据典型点有关的修正信息，例如根据

25 随机数来改变时间信息的修正信息。

如图 18 所示，矢量解码器 31—35 控制各操作器的操作，例如，启动或关掉矢量操作器以便处理输入的矢量包，控制矢量操作器之间的连接或同步，控制时间，针对来自其他矢量 ID 流的每个矢量操作器输入设定参数转换等。矢量操作器 36 和 37 读取矢量信息数据，控制读取矢量信息数据的位置（速度输入），控制增益（增益输入）等。在矢量解码器 31—35 中控制矢量操作器 36 和 37 定义的各种参数。设置矢量解码器 31—35 使之对应于每个分量要素。相应的矢量解码器 31—35 从矢量包中读取矢量信息数据并按时间顺序产生所需的波形。

如图 19 所示，例如，矢量解码器 31 产生适合谐波分量中振幅要素的包络波形。矢量解码器 32 产生适合谐波分量中定音要素的包络波形。矢量解码器 33 产生适合谐波分量中波形（音调）要素的包络波形。矢量解码器 34 产生适合非谐波分量中振幅要素的包络波形。矢量解码器 35 产生适合非谐波分量中波形（音调）要素的包络波形。矢量解码器 33 产生的谐波分量波形带有谐波分量中振幅要素的包络波形和谐波分量中定音要素的包络波形。这些包络波形在矢量解码器 31 和 32 中产生。矢量解码器 33 随后向混合器 38 输出产生的波形。即，象增益控制（增益输入）的矢量操作器那样，向矢量解码器 33 提供谐波分量中振幅要素的包络波形。象控制矢量信息数据读取位置（速度输入）的矢量操作器那样，提供适合谐波分量中定音要素的包络波形。矢量解码器 35 产生的非谐波分量波形还有非谐波分量中振幅要素的包络波形。该包络波形在矢量解码器 34 中产生。矢量解码器 35 随后向混合器 38 输出产生的波形。即，向矢量解码器 35 提供非谐波分量中振幅要素的包络波形作为增益控制（增益输入）的控制命令。

当在每个分量和要素中按时间顺序产生波形时，矢量解码器 31—35 使波形形成时的波形同步。例如，当输入波形（音调）要素的矢量包和振

幅要素的矢量包时，根据以波形（音调）要素的矢量包为基础的波形产生时间并与之同步地产生以振幅要素的矢量包为基础的振幅波形。该振幅波形控制以波形（音调）要素矢量包为基础产生的波形的振幅。当输入波形（音调）要素的矢量包和定音要素的矢量包时，根据以波形（音调）要素

5 矢量包为基础产生波形的时间并与之同步地产生以定音要素的矢量包为基础的定音要素。

定音波形控制以波形（音调）要素矢量包为基础产生的波形的定音。当输入谐波分量中波形（音调）要素的矢量包和非谐波分量中波形（音调）要素的矢量包时，根据以谐波分量中波形（音调）要素的矢量包为基础的

10 谐波分量合成时间并与之同步地合成以非谐波分量中波形（音调）要素的矢量包为基础的非谐波分量。通过将合成的谐波分量和非谐波分量波形混合可产生理想的音乐波形。

可以将该实施例设计成在谐波分量和非谐波分量之间选择同步或不同步。只有当选择同步时，才可以优选根据以上述谐波分量中波形（音调）

15 要素矢量包为基础产生的谐波分量波形合成时间，并与之同步地合成以非谐波分量中波形（音调）要素矢量包为基础产生的非谐波分量波形。

如上所述，包流包括多个包。在矢量包包流的情况下，每个包都包含矢量数据。即，包流包括沿临时方向布置的矢量数据。根据振幅、定音、和波形（音调）要素，矢量数据包含不同的数据结构和含义。然而，从矢

20 量操作器 36 和 37 的角度看，矢量数据基本上相同。

### 3.4.2 矢量数据结构

图 20 是矢量数据结构的实施例示意图。例如，用单位（SEC）表示读取矢量数据的位置。当假定读取速度不变时，矢量数据上的一个采样值与输出波形的一个采样值匹配。用单位  $1/1200$ （音程）（取 2 的  $n$  次方）

25 表示读取速度。幂  $n=0$  意味着速度不变。幂  $n=1.0$  意味着速度加倍，例

如，波形（音调）要素高八度。幂  $n=-1.0$  意味着速度减半，例如，波形（音调）要素低八度（参见图 20 中的上图）。代码本 26 存储实际的矢量数据。例如，振幅要素或定音要素的矢量数据包括阵列式矢量点（VECTORPOINT）结构和典型点数据。

- 5        阵列式矢量点（VECTORPOINT）结构包含采样位置和按序排列的每个点值。例如，用单位（db）表示振幅要素的矢量数据值。当假设编号为 0 的 MIDI 音符是 0.0 时，用单位  $1/1200$ （音程）表示定音要素的矢量数据值。典型点数据是 DWORD 阵列并存储了作为典型点使用的阵列式矢量点（VECTORPOINT）结构的指数（参见图 20 中的下图）。显然，本发明  
10 并不限于上述实例。

### 3.4.3 高速缓存控制部分 40 的细节

#### （1）高速缓存控制部分 40 的整体结构

- 该部分参照图 23 描述了高速缓存控制部分 40 的整体结构。首先描述的是设置该高速缓存控制部分 40 的目的。由于硬盘 109 中存储着代码本  
15 26，所以可从硬盘 109 中读取矢量解码器 31—35 所需的矢量数据。然而，硬盘等提供的存取速度即慢又不稳定。因此不能在矢量解码器 31—35 处理相关矢量数据时及时读出矢量数据。作为一种方案，本实施例在波形合成部分 101D 设置了高速缓存部分 40 以便将待用（或是预计将使用的）的矢量数据加载到高速缓冲存储器中。

- 20        在图 23 中，参考标记 42 表示预取部分。预取部分从演奏合成部分 101C 送到波形合成部分 101D 的包中抽取矢量 ID。预取部分对硬盘 109 提供读取控制，以便使预取矢量数据与来自代码本 26 的矢量 ID 相关联。如上所述，这些包在包队列缓冲器 21 中构成包流，并且由矢量加载器 20 读取。同时，进行预取处理。

- 25        参考标记 41 表示预测控制部分。预测控制部分根据从乐谱注释部分

101B 提供的预测数据（程序改变等）和从预取部分 42 提供的预测条件来预测具有极高实用性的矢量数据。预测控制部分随后向预取部分 42 提供与预测的矢量数据有关的矢量 ID。因此，供给预取部分 42 的矢量 ID 可以包含那些与舍去的矢量数据有关的数据，所述舍去的矢量数据在将来可以不实际使用。在此，“预测条件”包括确定将要使用的矢量 ID 等（即，从演奏合成部分 101C 提供的矢量 ID）。用这种方式，演奏合成部分和预测控制部分 41 将矢量 ID 送给预取部分 42。预取部分 42 预取与两个矢量 ID 相关的矢量数据，同时按优先序排列确定将要使用的矢量数据负载（即，由演奏合成部分 101C 确定的矢量数据负载）。在下文中，把确定将要使用的加载矢量数据称为“指定的负载”；把不是确定要使用的加载矢量数据（只是预测将会使用）称为“预测负载”。参考标记 44 表示存储预取矢量数据的高速缓冲存储器。参考标记 43 表示读取控制部分。当从矢量加载器 20 接收矢量 ID 时，读取控制部分主要从高速缓冲存储器 44 中读取与该矢量 ID 相应的矢量数据，并将该矢量数据送到矢量加载器 20。参考标记 45 表示时间控制部分，其为预取等提供时间控制。

## （2）预测控制部分 41 的操作

下面将参照图 24 中的状态转换图描述预测控制部分 41 中的处理内容。预测控制部分 41 的状态取决于波形合成是否发生在矢量解码器 31—35 中，并且当进行合成，根据进行合成的模式来合成波形。最初时，在矢量解码器 31—35 中没有进行波形合成。预测控制部分 41 完成步骤 S30 的程序。在此，程序预测硬起音部分中的候选矢量数据。预测的矢量 ID 依次送到预取部分 42。然而，在步骤 S30，并不预测硬起音部分，除非乐谱表演部分 101B 提供作为预测数据的“程序变化”。这是因为不确定的程序变化会产生大量用于指定硬起音部分的矢量数据。在初始状态下，当从乐谱注释部分 101B 提供变化的程序时，预测控制部分瞬时启动适合

硬起音部分的矢量数据预测负载，所述硬起音部分对应于所述程序变化。

例如，假设指定“钢琴”作为变化的程序并且有 100 种适合硬起音部分的矢量数据与“钢琴”有关。在这种情况下，预测控制部分启动 100 种矢量数据的预测负载。当一个包满足硬起音部分时，矢量加载器 20 和  
5 矢量解码器 31—35 启动硬起音部分的波形合成程序。这时，预测控制部分 41 的状态移到步骤 S31。

在步骤 S31，在设定的硬起音部分矢量 ID 的基础上，根据主体部分的候选矢量数据进行预测加载。由于硬起音部分的定音是已知的，所以需要限制候选的矢量数据，使之与硬音部分的定音相适应。将矢量数据还限制  
10 到适合与设定的硬起音部分的包络波形对应的主体部分的范围内。在步骤 S31，预测控制部分根据这些条件预测性地加载缩小了的候选数据。在此，演奏合成部分 101C 实际上提供主体部分的包。当设定了主体部分的包时，预测控制部分 41 的状态转向步骤 S33。

在步骤 S33，预测控制部分根据设定的主体部分的矢量 ID，预测性地  
15 加载下一矢量数据的候选数据。如上所述，与主体部分有关的模式是其他主体部分、接合部分、或释放部分之一。因此，预测控制部分 41 预测性地加载这些矢量数据。用与步骤 S31 同样的方式，根据设定的主体部分的定音、包络波形等使将要预测性地加载的候选矢量数据缩小。

当演奏合成部分 101C 在执行步骤 S33 期间实际上传送包时，预测控制部分 41 的状态随所述包而变化。当提供的包与主体部分相关时，将保持步骤 S33 的状态。在该主体部分的基础上，针对其他主体部分、接合部分、或释放部分再度执行预测加载。当提供的包与接合部分相关时，预测控制部分 41 的状态转移到步骤 S32。

在步骤 S32，预测控制部分预测性地加载适合主体部分的矢量数据。  
25 预测性地加载的候选矢量数据使可能连接到固定接合部分的主体部分之矢

量数据变窄。当演奏合成部分 101C 提供与下一个主体部分相关的包时，预测控制部分 41 的状态转向步骤 S33 以便预测性地加载其他主体部分、接合部分、或释放部分的矢量数据。当演奏合成部分 101C 提供适合释放部分的包时，预测控制部分 41 的状态转向步骤 S30。

- 5        如上所述，在步骤 S30，预测控制部分预测性地加载硬起音部分的候选矢量数据。当已确定了前一释放部分的定音时，随后的硬起音部分的定音很有可能并不明显偏离。在合成释放部分期间，优选限制将要预测性地加载到释放部分定音附近的硬起音部分的矢量数据（例如，在±1个八度音的范围内）。

### 10        (3) 预取部分 42 的操作

#### (3.1) 加载程序（图 25）

下面将描述预取部分 42 的操作。首先，预取部分 42 在特定的周期执行图 25 的加载程序。当程序进入图中的步骤 S41 时，将判断是否接收目前不执行的指定加载请求。如果判断的结果是 YES，程序将转向步骤 S42  
15 并完成请求矢量数据的指定加载。在给定的读取时间过后，从硬盘 109 中读出这些矢量数据并将其存储到高速缓冲存储器 44 中。

当步骤 S41 的判断结果是 NO 时，程序转入步骤 S43 并完成预测加载。对于为预测加载而请求的矢量数据，程序需检测将读取命令发送到高速缓冲存储器 44 的矢量数据（按照其它加载请求已经加载到高速缓冲存储器  
20 44 中的矢量数据和计划加载的矢量数据）；和不向硬盘 109 发送读取命令的矢量数据。向硬盘 109 提供的是读取后者的命令。当需要预测性地加载很多矢量数据时，可在预测加载的中期将该程序再次调入。在这种情况下，只要请求的指定加载生效，相关的预测加载将中断并且通过步骤 S42 完成指定加载。在预取部分 42 中，指定加载在预测加载前进行。

#### 25        (3.2) 包接收程序（图 26）

每当演奏合成部分 101C 传送包时，预取部分 42 便执行图 26 中的包接收程序。当程序进入图中的步骤 S51 时，将从提供的包中抽取矢量 ID 以判断是否命中了与该矢量 ID 对应的矢量数据。在此，在以下情况下可以应用“击中的矢量数据”：（1）当已预测性加载到高速缓冲存储器 44 中的这些矢量数据中的任何矢量数据被击出时，（2）当矢量数据目前并没加载到高速缓冲存储器 44 中但是已计划作为预测加载时，或（3）一旦可以使用加载的和可用的矢量数据（使用状态）时。

在本实施例中，所用矢量数据并不立即释放，而是暂时保持在高速缓冲存储器 44 中作为处在“使 USED 状态”的矢量数据。在例如存储容量太小以致于不能存储其他矢量数据时，释放使用（USED）状态矢量数据区以便存储新的矢量数据。细节将在下面描述。

当没有击中矢量数据时，步骤 S51 的判断结果是 NO。程序转入步骤 S53。此时，将产生指定的加载请求，该请求对应于包含在上述包内的矢量 ID。当紧接着执行包接收程序（图 26）时，完成适合该矢量数据的指定加载。

当出现击中的矢量数据时，步骤 S51 的判断结果是 YES。程序转入步骤 S52。如果没有完成该矢量数据的预测加载，预测加载请求将变为指定加载请求。这是因为，当执行后面的包接收程序时，将优先加载所述矢量数据。当程序转向步骤 S54 时，程序把特定加载请求的矢量数据句柄（handle）返回到演奏合成部分 101C。这个句柄唯一地对应于每个矢量数据。当已经把特定加载请求的矢量数据存储到高速缓冲存储器 44 中时，使所述矢量数据的句柄返回。此外，产生新的句柄并将使返回到演奏合成部分 101C。

当程序转向步骤 S55 时，程序将取消预测的加载请求，该预测的加载请求是在先前出现的和不成功的。下面将描述“消除”的特定内容。当程

序转入步骤 S56 时，程序将改变预测控制部分 41 中的预测条件。即，程序使图 24 中每种状态下的候选预测加载变窄或根据由演奏合成部分 101C 提供的包改变预测控制部分 41 的状态。

从演奏合成部分 101C 的角度看，该包接收程序相当于图 28 中的“发出矢量 (GetVector) 命令”。即，演奏合成部分 101C 向波形合成部分 101D 提供包。该操作刚好等同于根据包含在包内的矢量 ID 发送“发出矢量数据”的 GetVector 命令。必须在矢量加载器 20 随后读出矢量数据之前准备该矢量数据。从这一点上看，指定所期望的矢量数据需要不多的句柄。

#### 10 (4) 高速缓冲存储器 44 中的数据结构

##### (4.1) 高速缓存页面 (图 27)

高速缓冲存储器 44 可以存储将要同时使用的矢量数据。这些矢量数据构成包并且转换成一个文件(或少于或等于矢量数据总量的多个文件)。转换是根据用户的指令或建立的歌曲数据在音乐演奏之前自动执行的。图 27 中示出了一个实例。在图 27 中，对从代码本 26 中读出的某些矢量数据进行指定加载。这些矢量数据包括时间信息。因此，可以抽取多个将要同时使用的矢量数据。从代码本中读出的矢量数据包含各自的标题。例如，把将要同时使用的矢量数据收集在一个文件中。给该文件起一个命令标题。这样能够缩短矢量加载器 20 存取每个矢量数据的时间。

20 该文件标题包含以下信息。

数据 ID: 存储识别文件类型用的 4 位识别字符“PACK”。

数据量: 表示文件数据量。

VQ 类型: 表示存储的矢量数据类型。

版本: 表示文件格式的版本。

25 按照本实施例，在个人计算机上运行的应用程序发挥波形发生器的作

用。系统存储器包含其高速缓冲存储器。当高速缓存矢量数据时，根据具有特定尺寸的高速缓存页面，而不根据单个矢量数据（具有自由尺寸）实施高速缓存控制。即，将单个高速缓存数据分成多个高速缓存页面，并且，将其高速缓存，以便在高速缓存页面单元内进行管理。系统存储器存储对  
5 应于每个高速缓存页面的高速缓存管理数据（页面标题）。即，硬盘根据尺寸固定的群集存储数据。高速缓存页面尺寸优选与群集尺寸或是该尺寸的整数倍相同。考虑到矢量数据的尺寸等，合适的是将高速缓存页面尺寸设定为 1—1 万个字节。

#### (4.2) 页面标题的数据结构

10 下面将描述供给每个高速缓存页面的标题数据结构。每个标题包括具有以下成员的 VDDLCSPAGE 结构。

dwPage: 为高速缓存页面指定的专用页号。

dwID: 该高速缓存页面内包含的矢量数据的矢量 ID。

dwSize: 该高速缓存页面的数据尺寸。

15 dwCount: 表示预取的数目。每当指定负载预取矢量数据时，该成员 dwCount 将递增“1”，而且每当矢量加载器 20 读取矢量数据时，该成员 dwCoun 将递减“1”。

dwStatus: 表示高速缓存页面的状态。高速缓存页面的状态是 FREE、ALLOCATED、USED、FILLED 和 LOCKED 之一。

20 lpBuf: 表示矢量数据实体（非标题部分）在高速缓存页面中的起始地址。

lpForward/lpBackward: 在本实施例中，多个 VDDLCSPAGE 结构形成了双向链接表。成员 lpForward 是一个可有效指向链路前方其它 VDDLCSPAGE 结构的指示字。成员 lpBackward 是一个可有效指向链路后方  
25 其它 VDDLCSPAGE 结构的指示字。

lpNext: 如上所述, 例如, 波形发生器可以设置在专用硬件上。在这种情况下, 把将要同时使用的多个矢量数据分成多个高速缓存页面。这些高速缓存页面形成“组”。成员 lpNext 是一个连续指示属于同一组的其它 VDDLCSPAGE 结构的指示字。

5       上述成员 dwCount 具有以下作用。当包流序列中的同一矢量数据被使用两次时, 将出现两次针对该矢量数据的指定加载请求。然而, 当矢量加载器 20 首次读取矢量数据时, 相应的高速缓存页面将设定为 USED 状态, 从而不能进行第二次读取。这个问题的解决方式是对应能连续读取高速缓存页面的次数进行计数。

10       所以, 如下所述, 可以多次地使用矢量数据。即, 在同一事件数据或多个事件数据的波形合成过程中, 可以多次地使用同一矢量数据。在任一情况下, 本实施例均使用成员 dwCount 来判断矢量数据的使用次数, 从而能够均匀地处理高速缓冲存储器 44 中的高速缓存页面。

#### (4.3) 页面标题的链接结构 (图 29)

15       图 29 表示用指示字 lpForward 和 lpBackward 构筑的双向链接表格结构。本实施例采用了图中所示的表格结构, 从而能自由地和快速地重新配置、添加或删除数据。

20       在图 29 中, A-1 是位于双向链接表格始端的页面标题。始端由特定指示字 lpTop 指示而终端由指示字 lpTail 指示。A-2 和 A-3 是与 A-1 属于同一组的页面标题。即, 该“组”相当于各高速缓存页面的数据, 所述高速缓存页面是通过分割一个文件形成的。页面标题 A-2 的起始地址由页面标题 A-1 的成员 lpNex 指示。页面标题 A-3 的起始地址由页面标题 A-2 的成员 lpNext 指示。

25       B-1 是链接到页面标题 A-1 下一级的页面标题。页面标题 B-1 的起始地址由页面标题 A-1 的成员 lpFoward 指示。页面标题 B-1 的成员

lpNext 指向属于同一组的页面 B-2 的起始地址。页面标题 B-1 的成员 lpForward 指向链接到下一组的页面 C-1 的起始地址。每个页面标题 C-1 和 C-2 的成员 lpNext 分别指向每个页面标题 C-2 和 C-3 的起始地址。在此，FREE 状态下的高速缓存页面构成一组。例如，将多个组 A 和 B 5 设定为 FILLED 状态。将 C 组设定为 FREE 状态。当高速缓存新的矢量数据时，从 FREE 或 USED 状态下的组中获得高速缓存页面。将该高速缓存页面设定成新矢量数据的高速缓存页面。

优选建立新的高速缓存页面和系统存储器上的相应页面标题，并用所述高速缓存页面作为新矢量数据的高速缓存页面。在这种情况下，希望将 10 高速页面的总量限制到特定值，以节约存储器资源。所述特定值可以根据系统存储量自动设定或由用户手工设定。当通过控制而总是将给定的恒定值作为 FREE 状态组的存储量时，能够将新矢量数据加速分配给高速缓存页面。

按照图 29 中的双向链接表格，参考成员 lpForward 可向前跟踪构成 15 表格组的起始页面标题 A-1、B-1 和 C-1。参考成员 lpBackward 可向后跟踪双向链接表格。为了检查是否已高速缓存给定的矢量数据，优选连续跟踪构成表格的起始页面标题和检查页面标题中的成员 dwID 是否与作为相关矢量数据的同一矢量 ID 匹配。如果匹配，与该页面标题有关的高速缓存页面是高速缓存所述矢量数据的那一组的起始高速缓存页面。当总 20 是用单个页面同时存储将要使用的所有矢量数据时，双向链接表格将仅包括页面标题 A-1，B-1，和 C-1。将无效数据存储在这些页面标题的成员 lpNext 中。

上述成员 dwStatus 表示重要等级顺序中诸如 LOCKED>FILLED>ALLOCATED>USED>FREE 等高速缓存页面状态。在按该重 25 要等级顺序将高速缓存页面排序的情况下，能够将高速缓存页面加速分配

给新的矢量数据。此外，优选直接在重要等级上反映成员 dwCount 的值。即，优选把较重要的等级分配给较大值的成员 dwCount。在 USED 状态的高速缓存页面之间，优选将不太重要的等级分配给预先移到 USED 状态的高速缓存页面。

#### 5 (5) 读取控制部分 43 的操作

再次参照图 28，下面将描述读取控制部分 43 的操作。矢量加载器 20 向读取控制部分 43 发送命令，以便根据包流内容，即，根据每个包中的矢量 ID 读取高速缓冲存储器 44。该命令称为锁存 (LockVector)。该锁存 (LockVector) 命令带有预先响应发出矢量 (GetVector) 命令返回的句柄 (handle)。在正常工作状态下，矢量数据应存储在高速缓冲存储器 44 中。锁存 (LockVector) 命令使得与所述矢量数据相关的高速缓存页面起始地址指示字返回到矢量加载器 20。

根据上述操作，矢量加载器 20 和矢量解码器 31-35 适当读取高速缓冲存储器 44 的内容以便完成波形合成程序。用这种方式，使由矢量加载器 20 准备的高速缓存页面呈现已锁存 (LOCKED) 状态 (下面将详述)，从而防止其内容被其它程序修改。

然而，根据情况，当从矢量加载器接收到 LockVector 命令时，可以不将所需的矢量数据加载到高速缓冲存储器 44 中。按照本实施例，当个人计算机应用程序发挥波形发生器的作用时，这种情况将经常发生。这是由于因个人计算机操作系统的条件而较长时间地占据了硬盘 109。在这种情况下，需根据波形发生器的操作模式执行不同的程序。

当非实时进行波形合成时，最好停止后序程序，直到加载了指定的矢量数据为止。为此，读取控制部分 43 以完成硬盘 109 的同步读取操作。该同步读取操作意味着在读出期望的数据之前不会执行其他程序。

25 波形发生器实时工作时，可通过进行同步读取操作中断乐音。替换面

的指示字返回到矢量加载器 20。本实施例使用了各种用于忠实表现不同音调变化的矢量数据。当通过程序变化选出的音调限制在没有表现演奏风格的基音矢量时，其容量不会很大。RAM103 存储多个这样的矢量作为所述音调的缺陷矢量。这些矢量替换不能在高速缓冲存储器 44 中准备的矢量数据。

即使是完成了实时合成，也可能出现这样的情况，即，在确保波形合成程序 S14（波形合成部分 101D）有足够的时间延迟的情况下，使该程序能够加载所需的矢量并合成该范围内的波形。在这种情况下，与非实时波形合成一样，当发现没有加载所需的矢量数据时，通过立即读取相应的矢量数据，便可完成波形合成。当矢量加载器 20 和矢量解码器 31-35 用完高速缓存页面后，矢量加载器 20 向读取控制部分 43 发出与该高速缓存页面对应的附有句柄的释放命令。通过发送该释放命令可取消高速缓存页面的已锁存（LOCKED）状态和释放句柄。

#### （6）高速缓存页面的状态转换操作

下面将参照图 30 中的状态转换程序描述每个高速缓存页面状态转换的操作。

最初时，将所有高速缓存页面设置到空白（FREE）状态（S61）。即，高速缓存页面的标题包含设为空白（‘FREE’）的成员 dwStatus。这意味着不使用高速缓存页面并且不保证其内容。当将矢量数据加载到高速缓存页面中时，该高速缓存页面将成为已指定（ALLOCATED）状态（S62）。已指定（ALLOCATED）状态（S62）意味着不存储数据而是将数据保留以便从代码本 26 中进行数据读取操作。这时，在成员 dwCount 中存储“1”。

当将矢量数据存储在上述随后页面时，高速缓存页面的状态变为已填充（FILLED）状态。每当相对于已指定（ALLOCATED）或已填充（FILLED）状态下的同一矢量数据出现指定的加载请求时，成员 dwCount 将递增“1”。

每当在 ALLOCATED 或 FILLED 状态下取消预测加载请求时，成员 dwCount 将递减“1”。当成员 dwCount 在这些状态下的值变为“0”时，高速缓存页面设定为 USED 状态（S63）。

高速缓存页面变成已填充（FILLED）状态之后，矢量加载器 20 发出  
5 锁存矢量（LockVector）命令并将高速缓存页面设定到已锁存（LOCKED）  
状态（S65）。当矢量加载器 20 和矢量解码器 31—35 用完高速缓存页面  
时，释放（解开）已锁存（LOCKED）状态并且发出释放命令。高速缓存  
页面再次设定到已填充（FILLED）状态（S64）。这时，高速缓存页面的  
成员 dwCount 递减“1”。当成员 dwCount 成为“0”时，高速缓存页面  
10 马上成为已使用（USED）状态（S63）。

高速缓存页面成为已使用（USED）状态后，可以重复适合所述高速缓存  
页面的预测加载请求或指定加载请求（预取）。在这种情况下，高速缓存  
页面再次返回到已填充（FILLED）状态。随着所述程序的继续，高速  
缓冲存储器中已使用（USED）状态的高速缓存页面增加而 FREE 状态的高  
15 速缓存页面减少。在高速缓存页面不保持空白（FREE）状态后，预取  
的新矢量数据最先连续取消（SwapOut）已使用 USED 状态的高速缓存页面  
并且将该高速缓存页面返回到空白（FREE）状态（S61）。该返回到空白  
（FREE）状态的高速缓存页面变为已指定（ALLOCATED）状态以便加载所  
述的新矢量数据。

20 优选使用图 29 中所示双向链接表格作为迅速指定已使用（USED）状  
态下最早高速缓存页面的方法。当高速缓存页面设定成为已使用（USED）  
状态时，将该高速缓存页面添加到链接表的始端。当空白（FREE）状态  
的高速缓存页面不够时，高速缓存页面将从链接表格的终端分离并变成空  
白（FREE）状态。

25 （7） 时间管理部分 45

时间管理部分 45 控制整个波形合成部分 101D 的时间。下面将参照图 31 说明本实施例中的时间控制。

图 31 假设在时间  $t_{30}$  发出（例如，由用户按下播放按钮）开始播放的指令，并在时间  $t_{40}$  开始实际输出乐音。时间  $t_{30}$  和  $t_{40}$  之间的时间周期称为等待时间  $\beta$ 。例如，将等待时间  $\beta$  设为 2000 毫秒（msec），但不需要特别指定。

发出开始播放指令时，CPU101 在每个特定时间产生播放线索。播放线索允许读取歌曲数据。根据歌曲内容，由硬盘 109 执行预取程序。从读出给定歌曲数据的时间  $t_{30}$  到歌曲实际开始的时间  $t_{40}$  之间存在时间周期。该时间周期称为先期时间  $\gamma$ 。优选将先期时间  $\gamma$  的参考值设定为，例如，接近 4000 毫秒（msec），并且根据处理的情况使其在 1000 到 10000 毫秒（msec）的范围内变化。

如果在某种程度上能确保先期时间  $\gamma$ ，则可以间断地产生播放线索而且间断地读取（在给定的数据块中）歌曲数据。产生播放线索的间隔称为播放线索发生间隔  $\epsilon$ 。优选将播放线索发生间隔  $\epsilon$  的参考值设为，例如，20 毫秒（msec）并根据处理情况，使其能在 5—100 毫秒的范围内变化。

当在时间  $t_{32}$  读取歌曲数据时，如上所述，在步骤 S11 检索歌曲数据部分。在步骤 S12 注释乐谱。在步骤 S13 完成演奏合成。结果，在时间  $t_{34}$  预取矢量数据。从预取时间  $T_{34}$  至歌曲开始时间  $t_{40}$  之间的时间周期称为预取时间  $\alpha$ 。

从硬盘 109 读出的矢量数据最终写入高速缓冲存储器 44 中。此后，矢量加载器 20 通过读取控制部分 43 从高速缓冲存储器 44 中读出矢量数据，以便开始波形合成。从波形合成开始的时间  $t_{36}$  到歌曲开始的时间  $t_{40}$  的时间周期称为输出等待时间  $\delta$ 。将输出等待时间  $\delta$  的参考值最好设为，例如，近似于 300 毫秒（msec），并且根据处理情况使其在 10—1000

毫秒 (msec) 的范围内变化。

与播放线索相同, 还适合在指定的时间范围内间断地进行给定量的波形合成处理。将开始波形合成处理的间隔参考值最好设定为, 例如, 50 毫秒 (msec), 并根据处理情况使其在 10—500 毫秒的范围内变化。

- 5 如图 31 中所示, 在先期时间 $\gamma$ 、预取时间 $\alpha$ 和输出等待时间 $\delta$ 之间存在关系 $\gamma > \alpha > \delta$ 。在此, 从预取时间  $t_{34}$  至波形合成开始时间  $t_{36}$  的时间间隔“ $\alpha - \delta$ ”要足够长, 以便从硬盘 109 加载矢量数据。当加载的矢量数据量形成高峰值时, 能够避免因时间间隔“ $\alpha - \delta$ ”的加大引起的噪音。

#### 4. 改型

- 10 本发明并不限于上述实施例。例如, 可以如以下面所述做出各种变化。

(1) 当将上述波形发生器作为电子乐器使用时, 该电子乐器可以是各种形式, 例如弦乐器、管乐器、打击乐器等。在这种情况下, 本发明并不限于包括歌曲数据播放部分 101A、乐谱注释部分 101B、演奏合成部分 101C、波形合成部分 101D 等的单一电子乐器。当然, 本发明还适用于  
15 分离部件, 这些分离部件使用通信装置例如 MIDI 接口、各种网络等而相互连接。设备配置可包括个人计算机和应用软件。在这种情况下, 优选提供存储在例如光盘、半导体存储器等上的或是通过网络存储的处理程序。此外, 本发明可以用于自动演奏设备, 例如弹奏钢琴。

(2) 在上述实施例中, 单个高速缓存页面可以存储多个矢量数据。  
20 显然, 单个高速缓存页面也可以存储一个矢量数据。

(3) 上述实施例需判断将要通过递增或递减高速缓存页面标题中的成员  $dwCount$  转到已使用 (USED) 状态的高速缓存页面。然而, 当未使用的高速缓存页面增加时, 该方法将使得高速缓冲存储器 44 变得不够用。在这种情况下, 可以优选规定已填充 (FILLED) 状态的高速缓存页面  
25 面“优先”, 然后将具有低优先性的已填充 (FILLED) 状态的高速缓存

页面移到已使用 (USED) 状态, 随后移到空白 (FREE) 状态。当从矢量加载器 20 接收到高速缓存页面已移到空白 (FREE) 状态的锁存矢量 (LockVector) 命令时, 使用上述替换页。作为“优先”的实例, 可以使用成员 dwCount 的值 (前提是, 所述值越大, 预先产生的影响越大)。

5 此外, 优选考虑使用过的高速缓存页面的量或成员 dwCount 的最大值 (历史上的最大值)。

(4) 在上述实施例中, 用双向链接表格将已使用 (USED) 状态的高速缓存页面结合。当空白 (FREE) 状态的高速缓存页面变得不够时, 连续释放链接表格终端的高速缓存页面 (从最早开始) 并将高速缓存页面  
10 变成空白 (FREE) 状态。然而, 并不限于向空白 (FREE) 状态转换的顺序。例如, 相对于具有很多用途的高速缓存页面, 在成员 dwCount 历史中具有较大最大值的高速缓存页面或在每个模式开始时使用的矢量数据, 优选避免转到空白 (FREE) 状态。换句话说, 如果可能的话, 可以将这些高速缓存页面或矢量数据设计成保持在高速缓冲存储器 44 中的结构,  
15 此外, 最好在实际上不使用的矢量数据之前高速缓存实际使用的矢量数据 (由于没有预测)。

通过优先高速缓存在模式开始时使用的矢量数据, 硬盘 109 可以以增大的存取容量工作, 从而降低了使用替换页的需要。

特别是, 当需要将优先高速缓存的高速缓存页面转换到已使用 (USED)  
20 状态时, 优选将该高速缓存页面添加到链接表格的始端。当需要将其他高速缓存页面转换到已使用 (USED) 状态时, 优选将这些高速缓存页面添加到链接表格的中部。

(5) 在上述实施例中, 是用矢量数据作为声音数据的实例。然而, 声音数据并不限于矢量数据。当然, 可以优选使用指定乐音波形的各种波  
25 形数据或参数作为声音数据。

(6) 在上述实施例中, 如图 15-23 所示, 演奏合成部分 101C 直接向预取部分 42 输入包流。代替这一操作, 预取部分 42 可以读取暂时写入包队列缓冲器 21-25 中的包流。在这种情况下, 优选的是, 预取部分 42 在矢量加载器 20 工作之前的特定时间内读取和处理存储在包队列缓冲器中的包。

如上所述, 按照本发明的第一方面, 音响设备接收音调类别或第一到第三声音数据类别中的一个, 预测候选的和将随后指定的声音数据, 并预先将所述候选数据从低速存储装置转送到高速存储装置中。因此, 可以迅速处理第一至第三声音数据类别, 同时将大量声音数据存储于低速存储装置中。这样便可以用丰富的表现力实时播放波形合成声音数据。

如上所述, 根据本发明的第二方面, 合成方法能够根据相关的计数值判断出下面是否使用转换到高速存储装置的声音数据。当声音数据是后面实际使用的声音数据时, 不需要对低速存储装置例如硬盘等进行存取。这能降低对低速存储装置例如硬盘等进行存取的频率。

如上所述, 按照本发明的第三方面, 本方法利用致密存储容量能够实时产生与演奏风格或乐音的发音相关的高质量波形。即, 根据演奏风格参数修正所产生的矢量数据。这样便不需要存储与多种音乐演奏模式(演奏风格)相应的矢量数据, 因此, 节约了预先存储矢量数据的存储容量。此外, 可以根据演奏风格参数, 并依据共用演奏风格识别信息, 精细地控制演奏声音波形的特性, 由此提高了可控性。

用这种方式, 可以使用具有不同演奏风格(发音)特性的高质量波形。这些波形包括例如颤音和震音等进行了调制的波形, 拐弯音等经过定音调制的波形, 连奏波形, 具有经过定音变化的过渡音或装饰音的波形。把产生自由波形的高质量波形简单地结合可以还改善所用的高质量波形。用改善可控性和可编辑性的方式可以使实时产生的与演奏风格或发音有关的

高质量波形具有极佳效果。

如上所述，按照本发明的第四方面，在用声音数据进行波形合成后，将所用数据存储在未来能够使用状态下的上述高速存储装置中。如果在下面实际使用了声音数据，则不再需要对诸如硬盘等低速存储装置进行存取。这将能够降低对诸如硬盘等低速存取装置存取的频率。

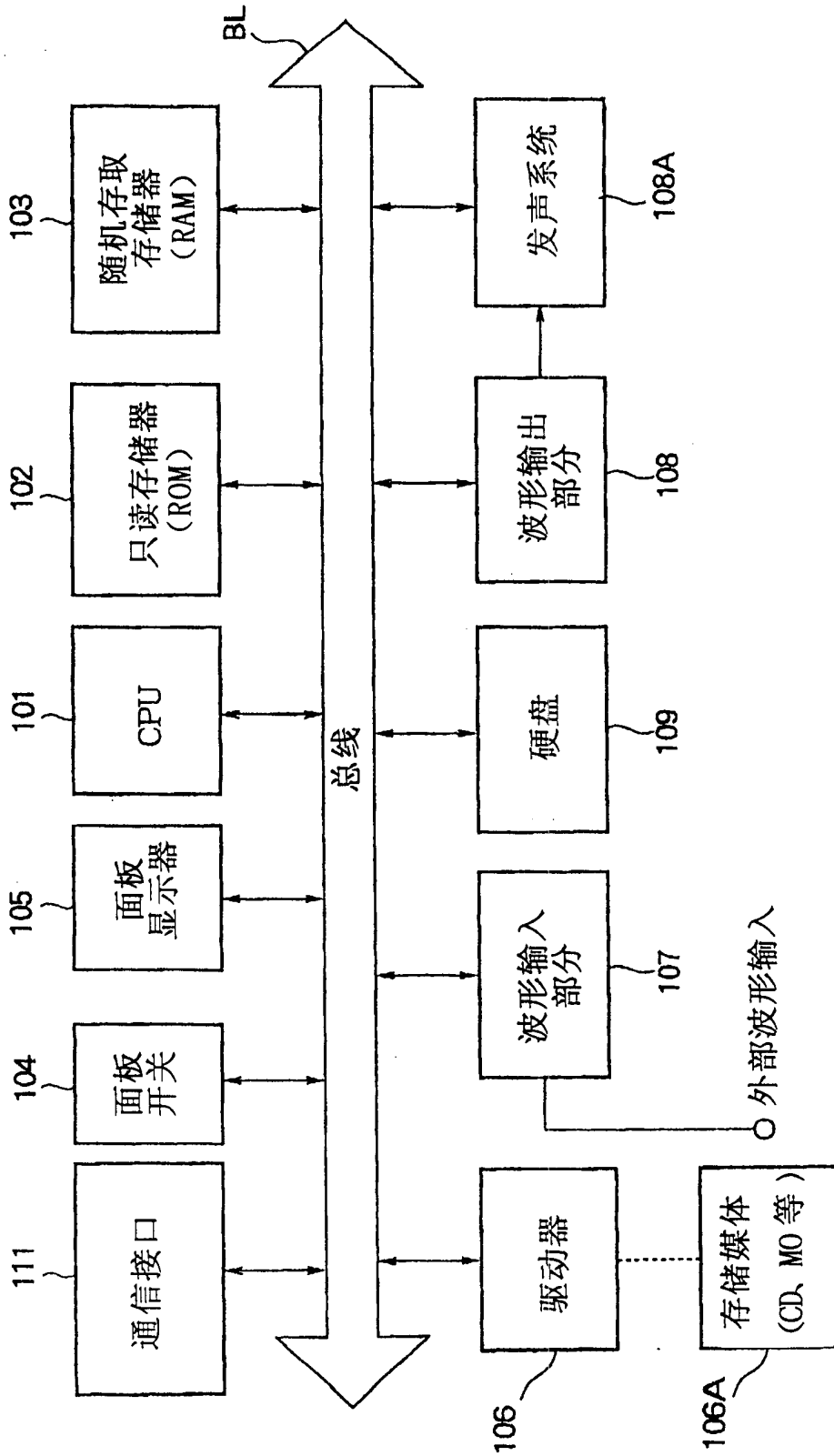


图 1

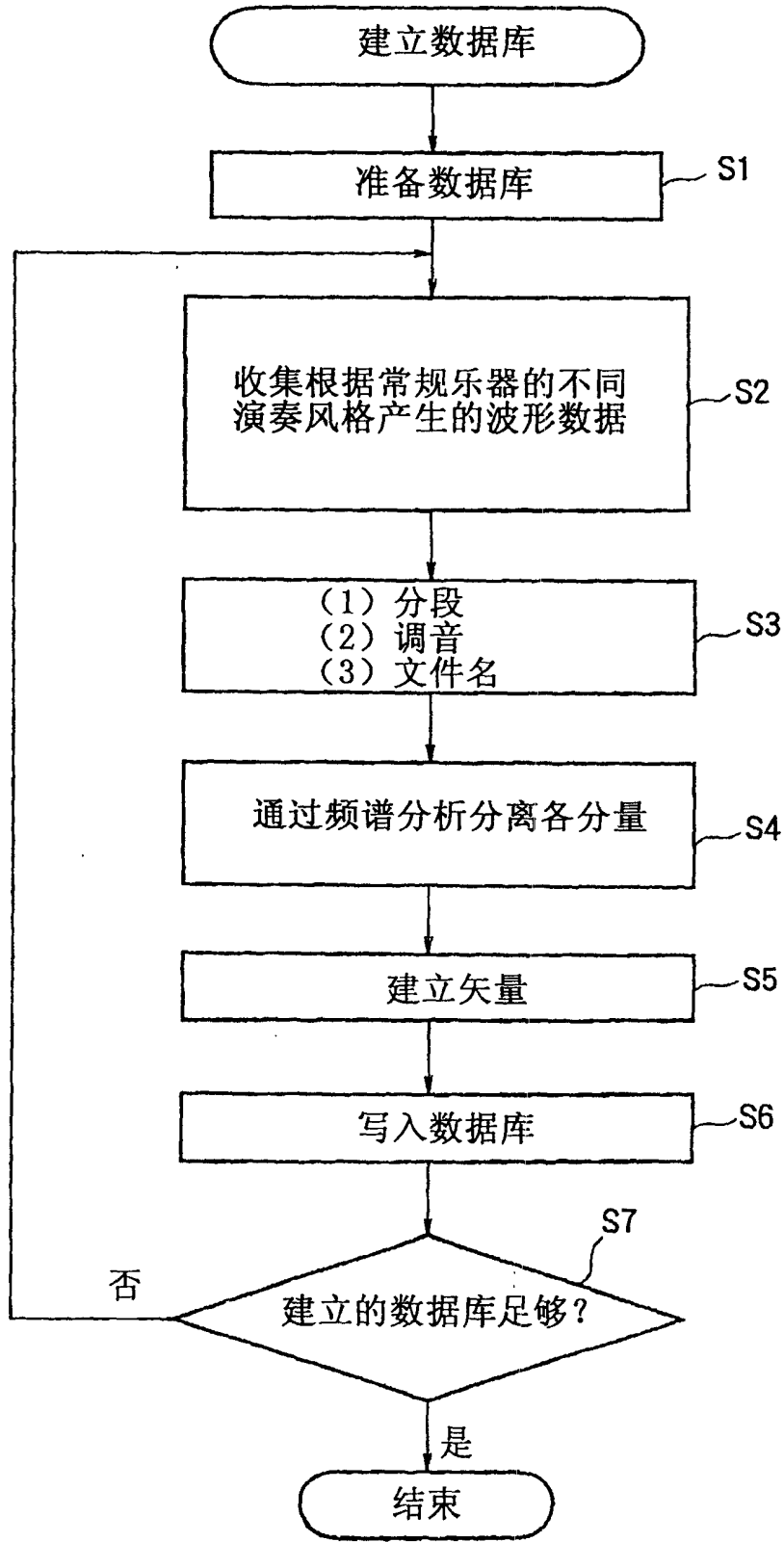


图 2

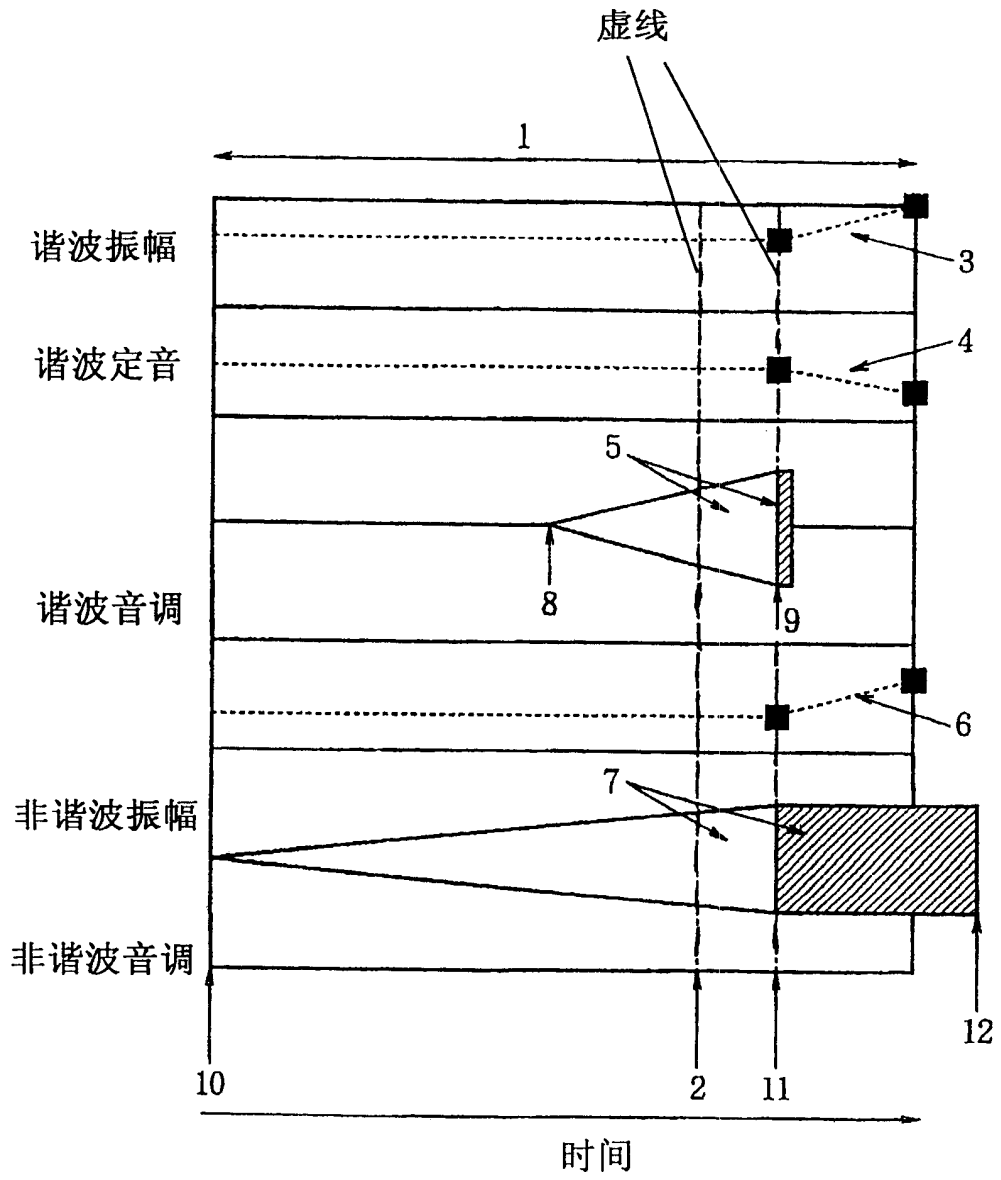


图 3

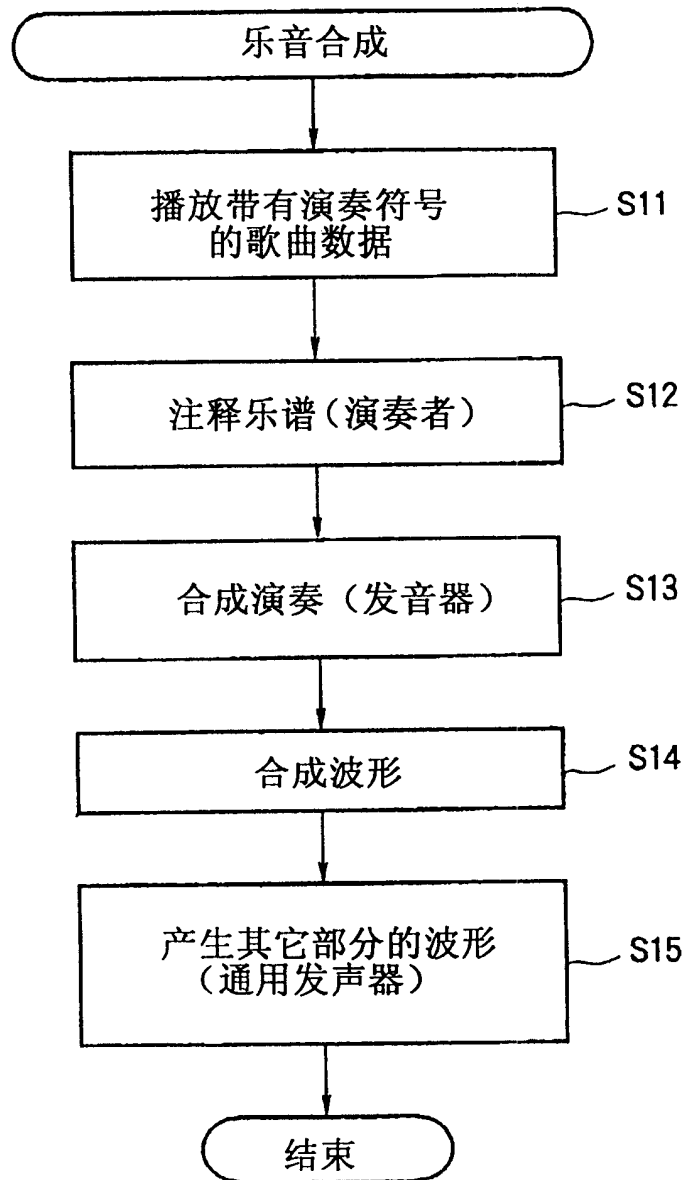


图 4

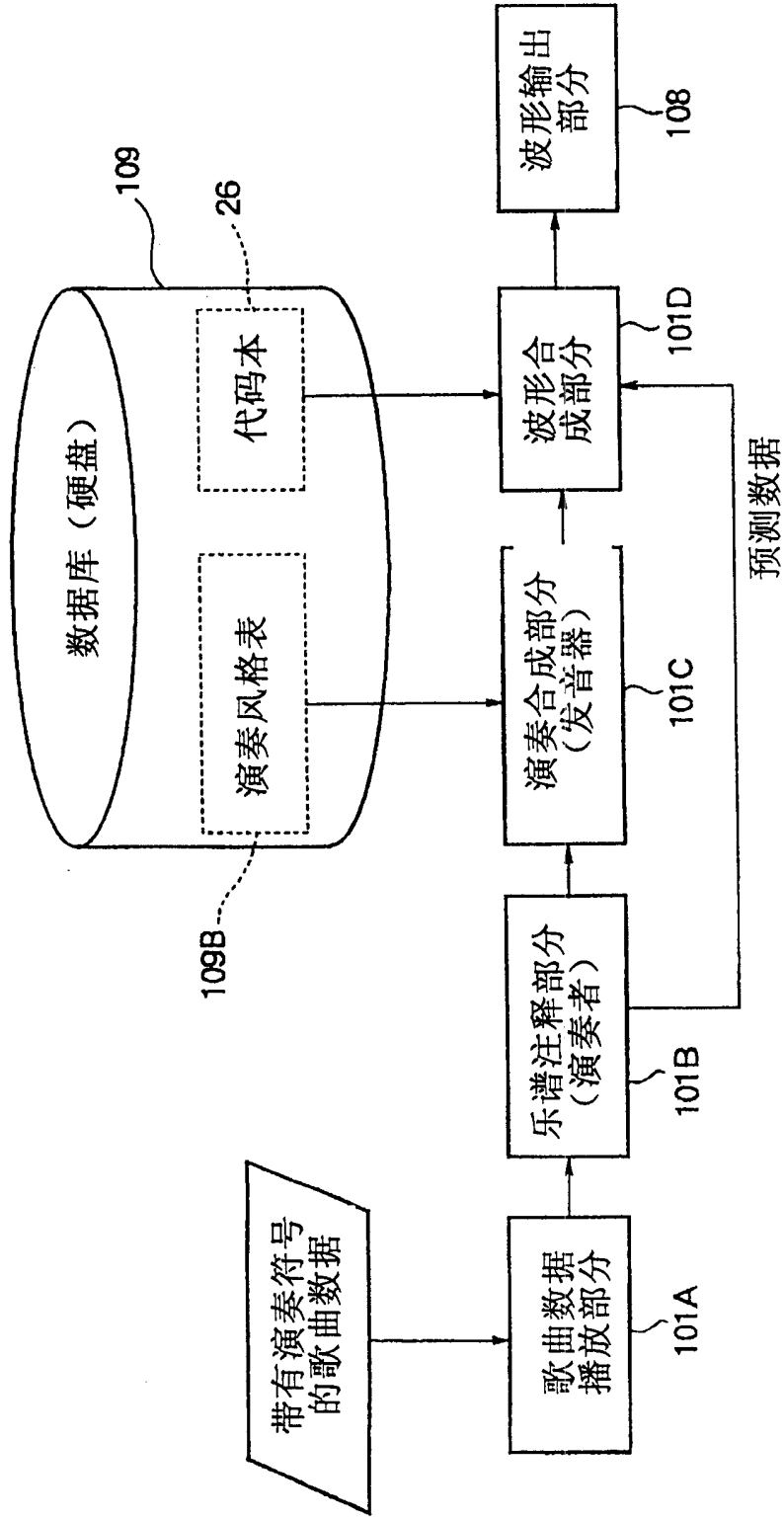


图 5

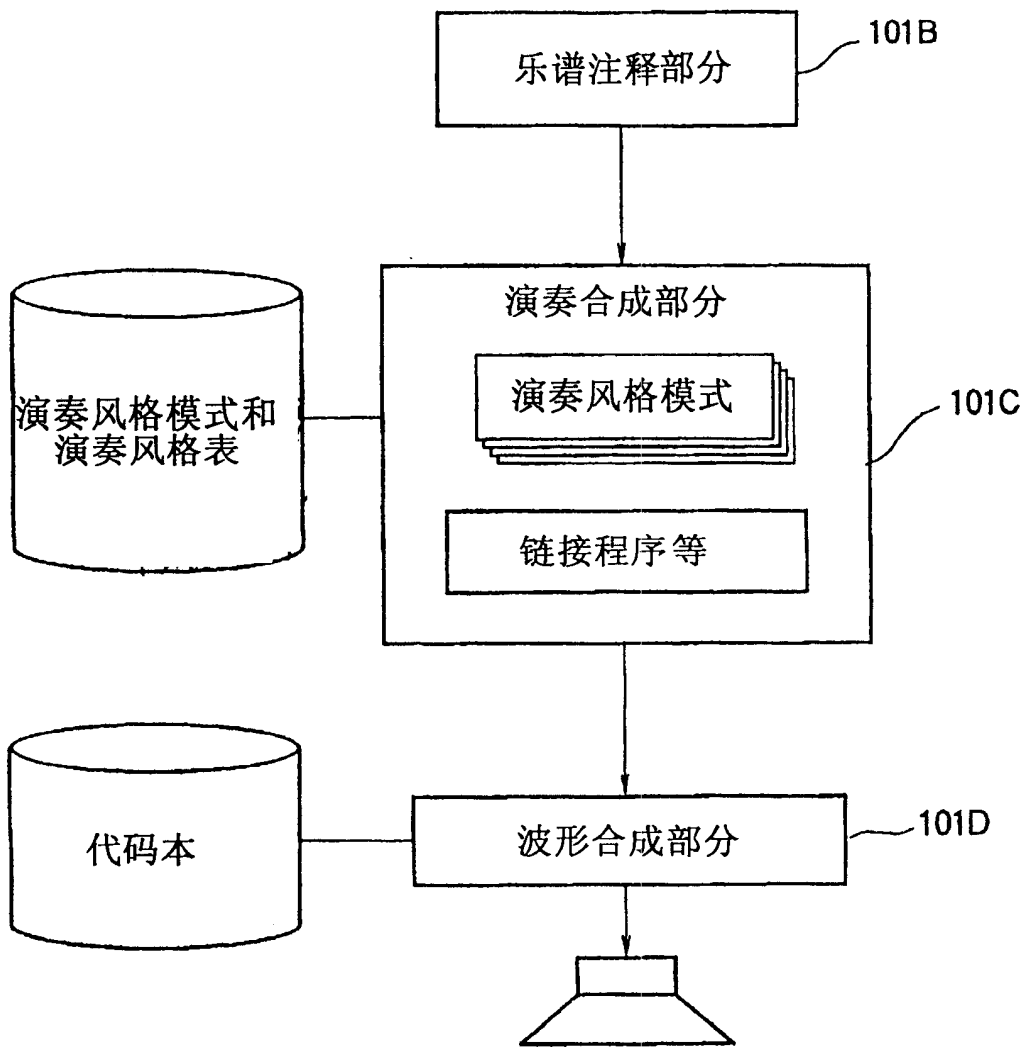


图 6

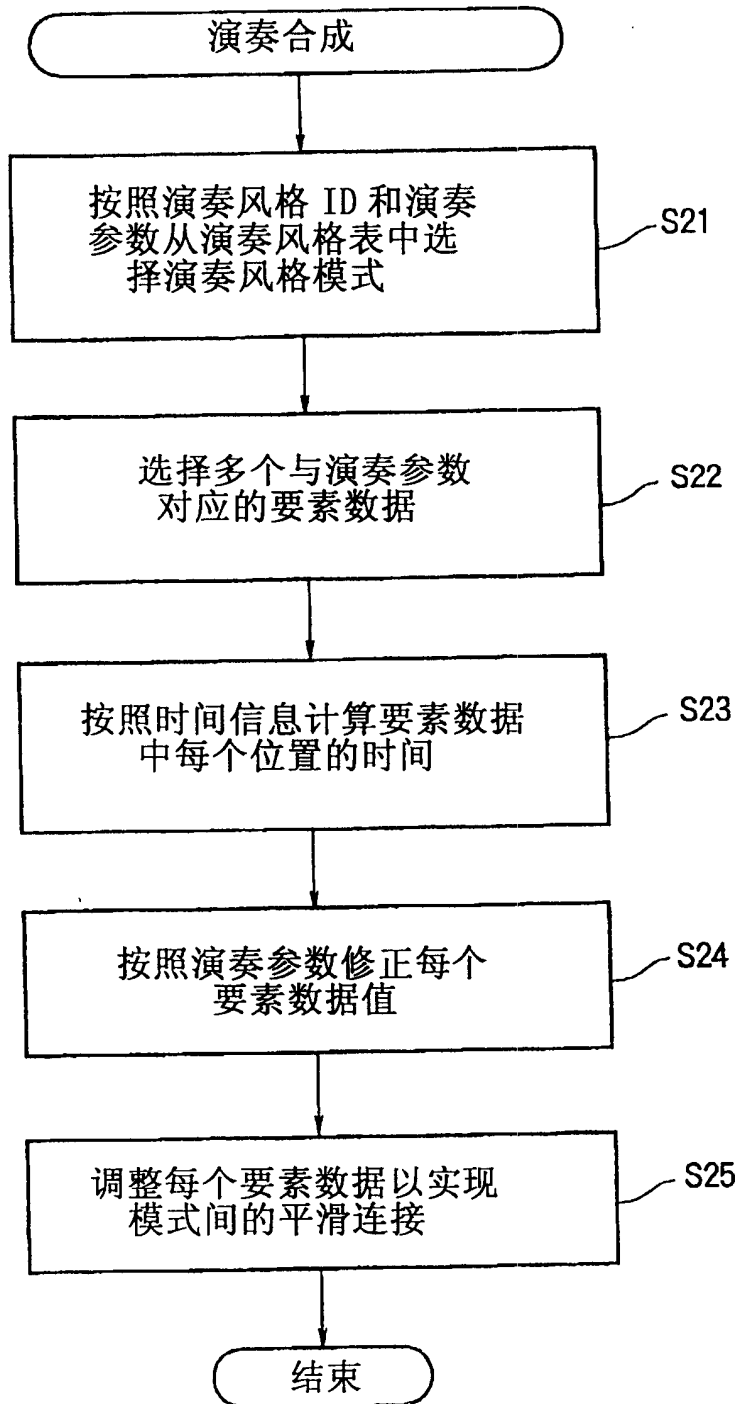


图 7

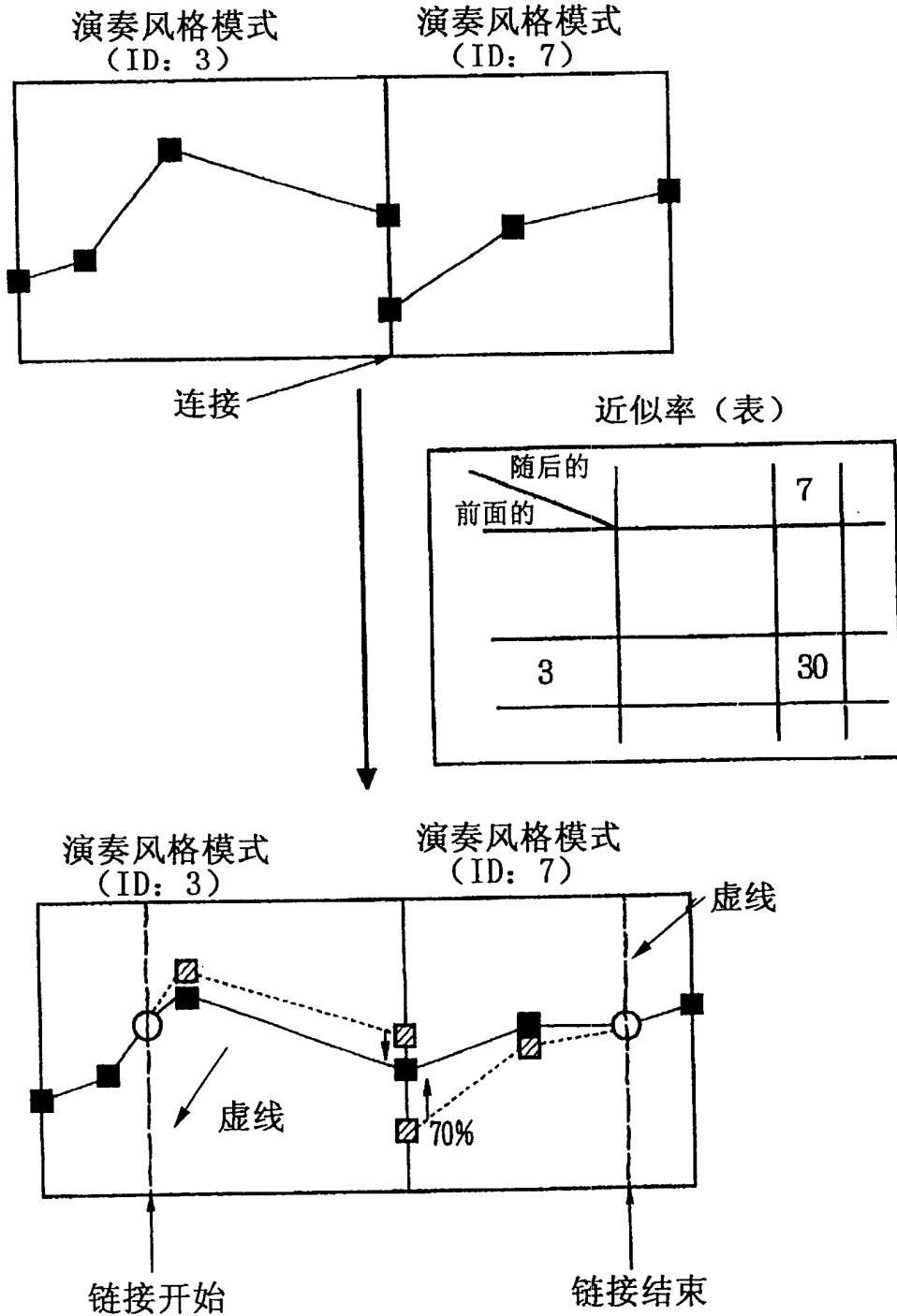


图 8

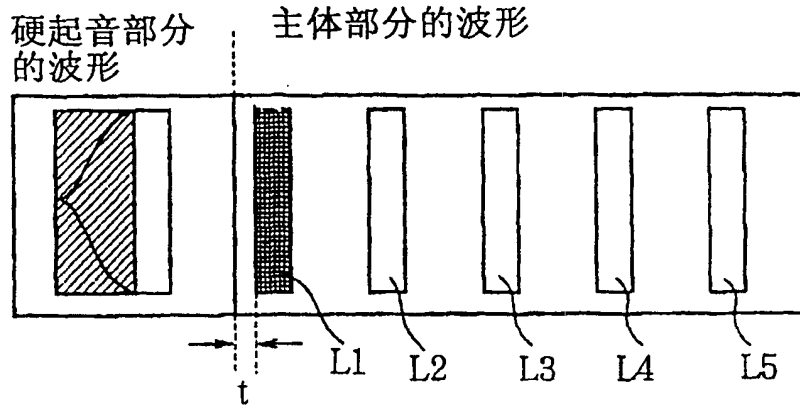


图 9

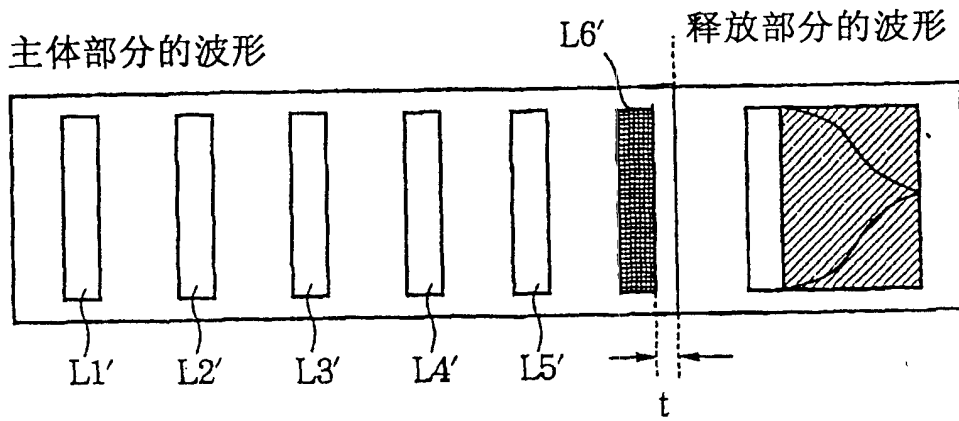


图 10

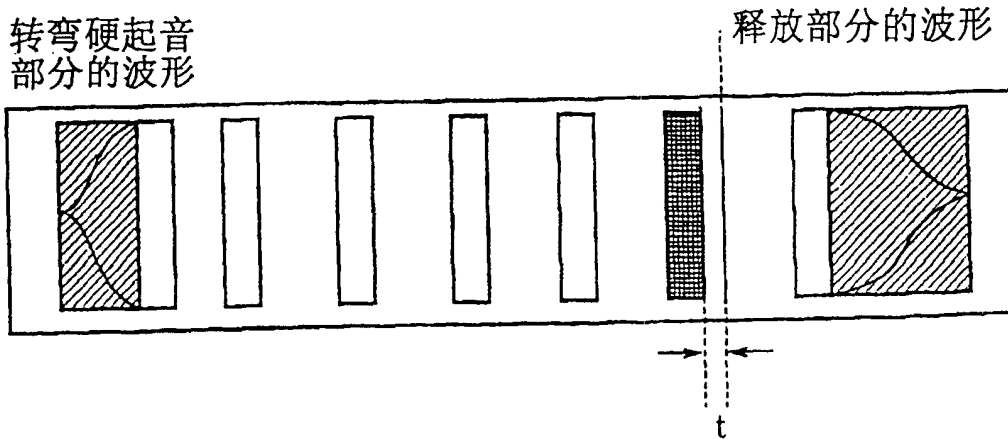


图 11

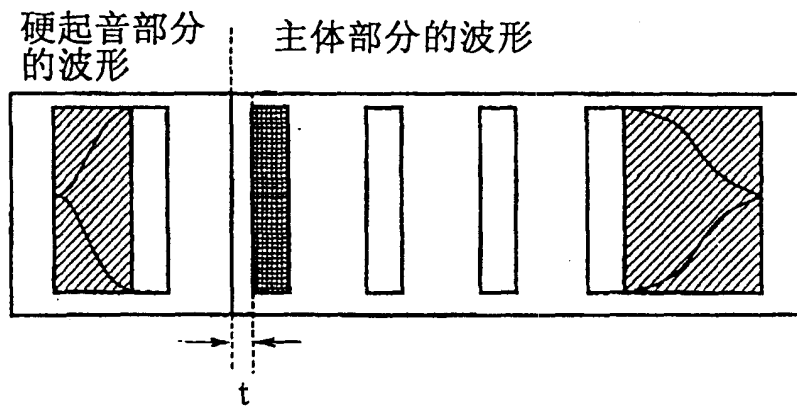


图 12

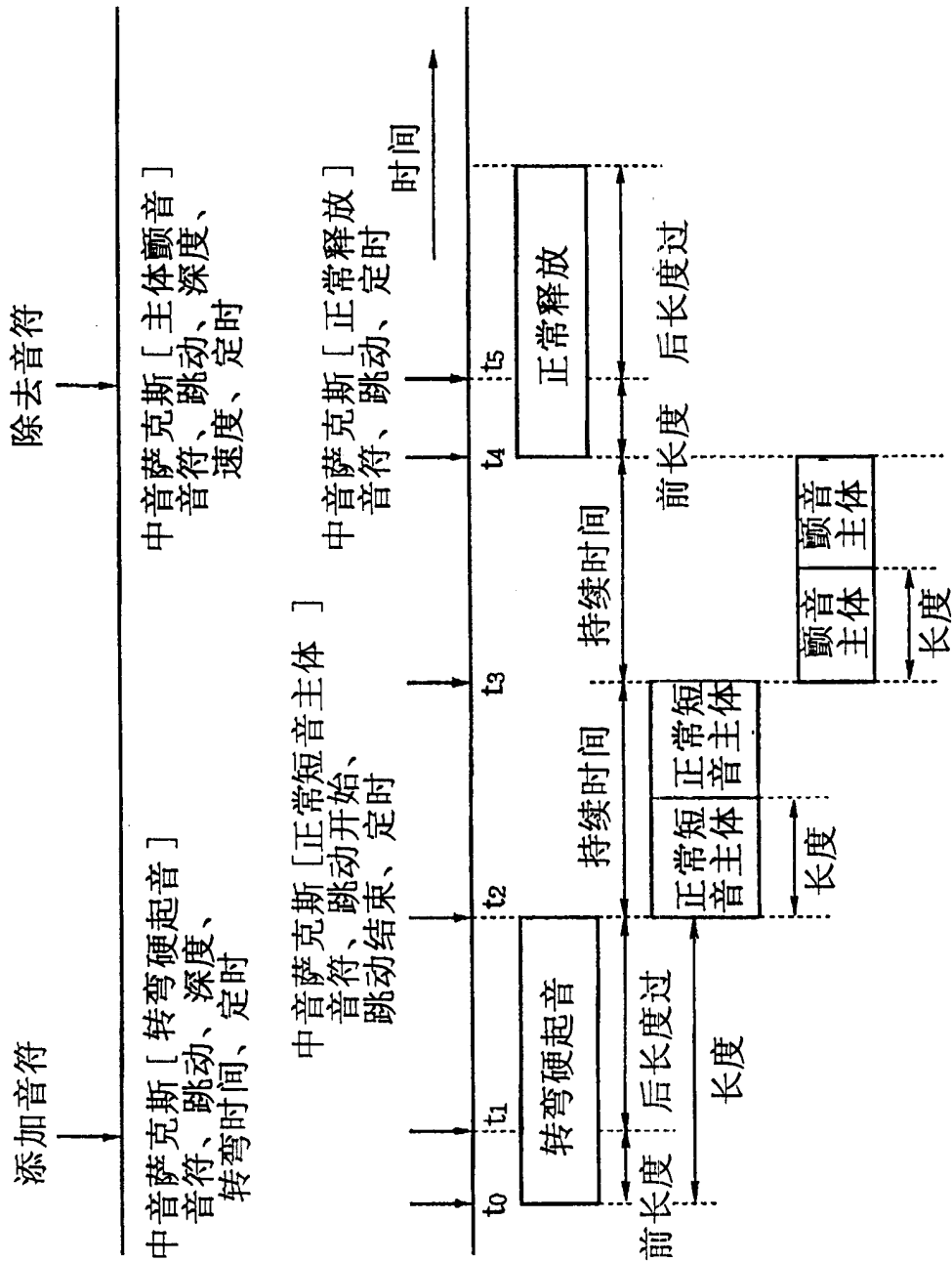


图 13

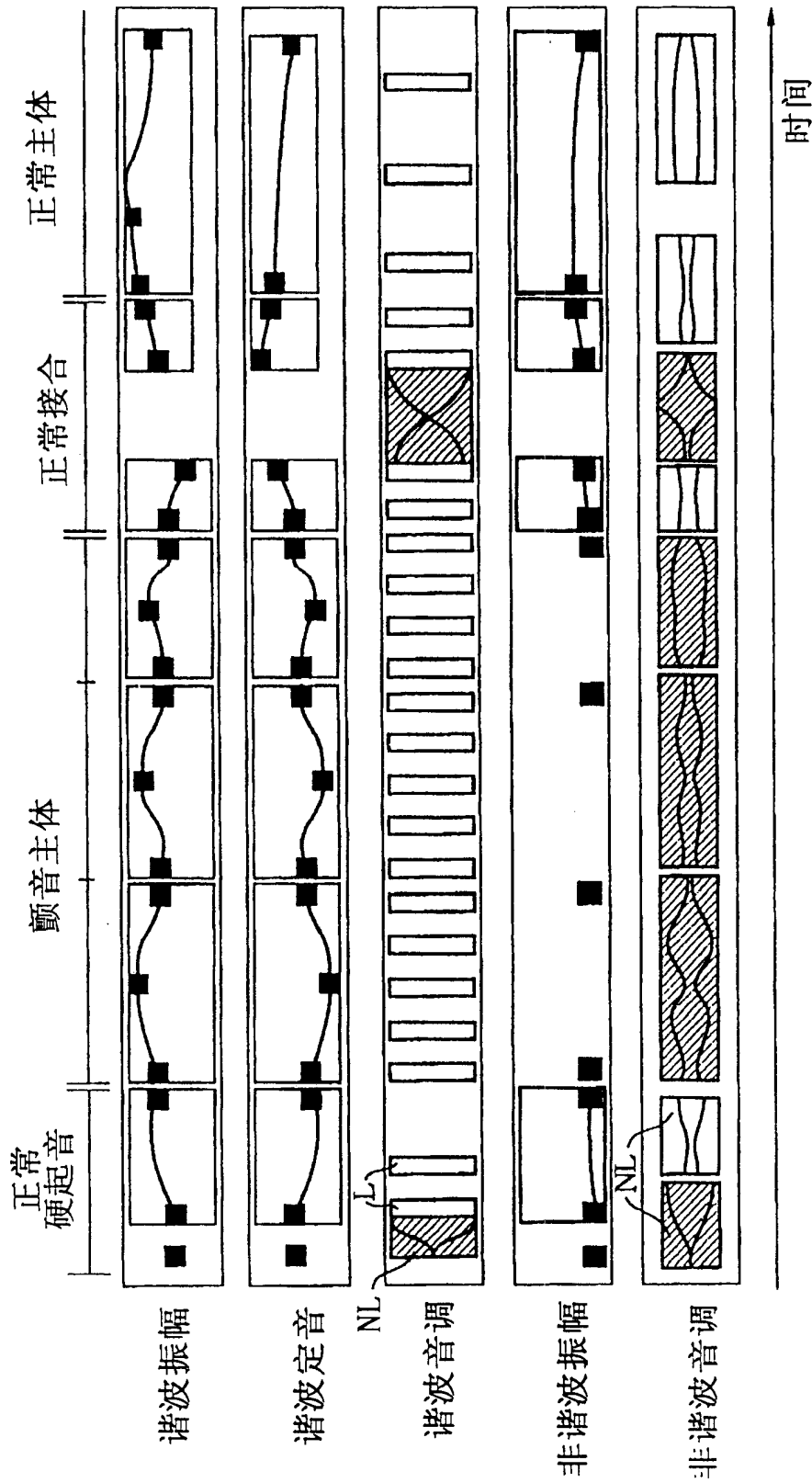


图 14

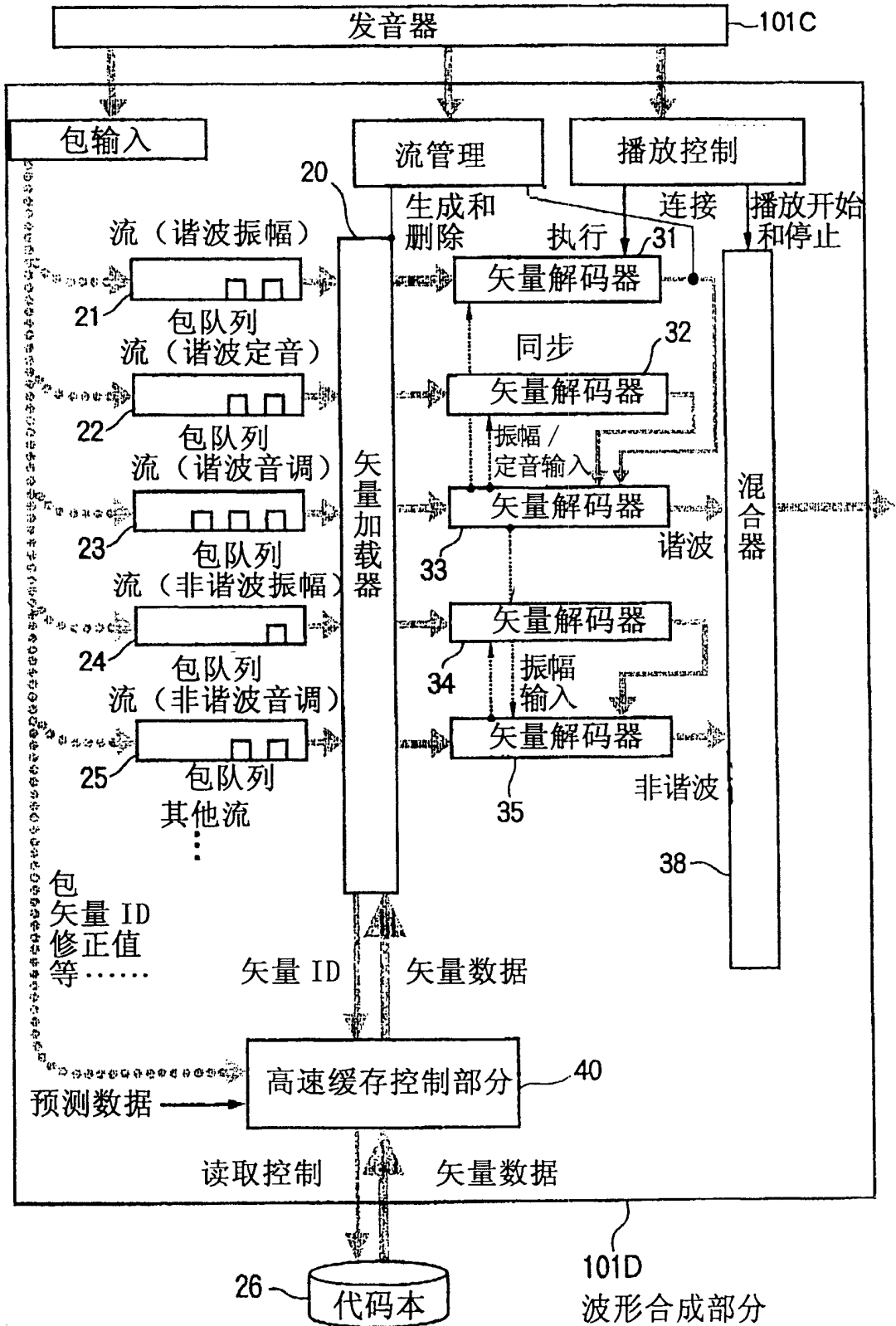


图 15

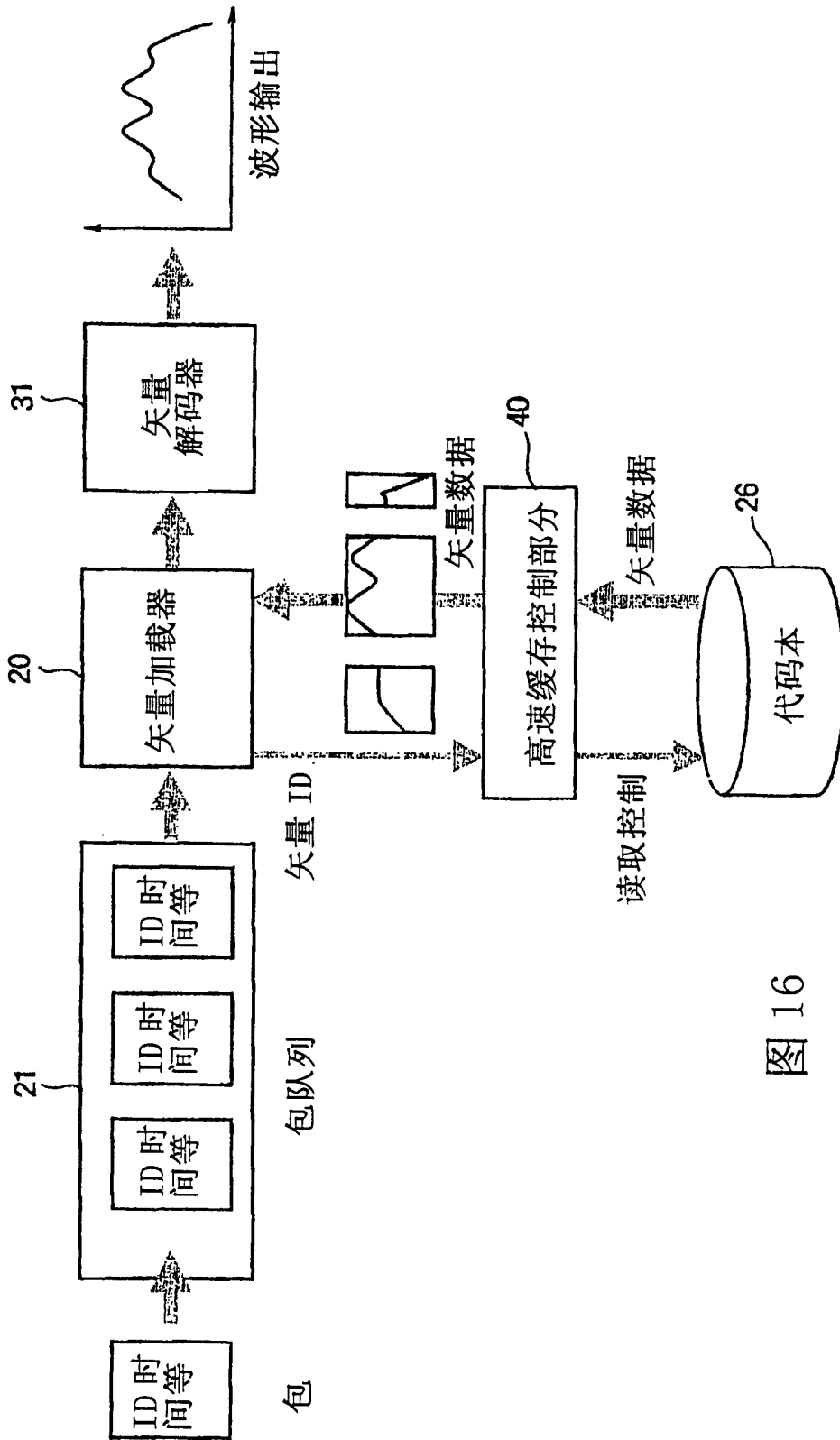


图 16

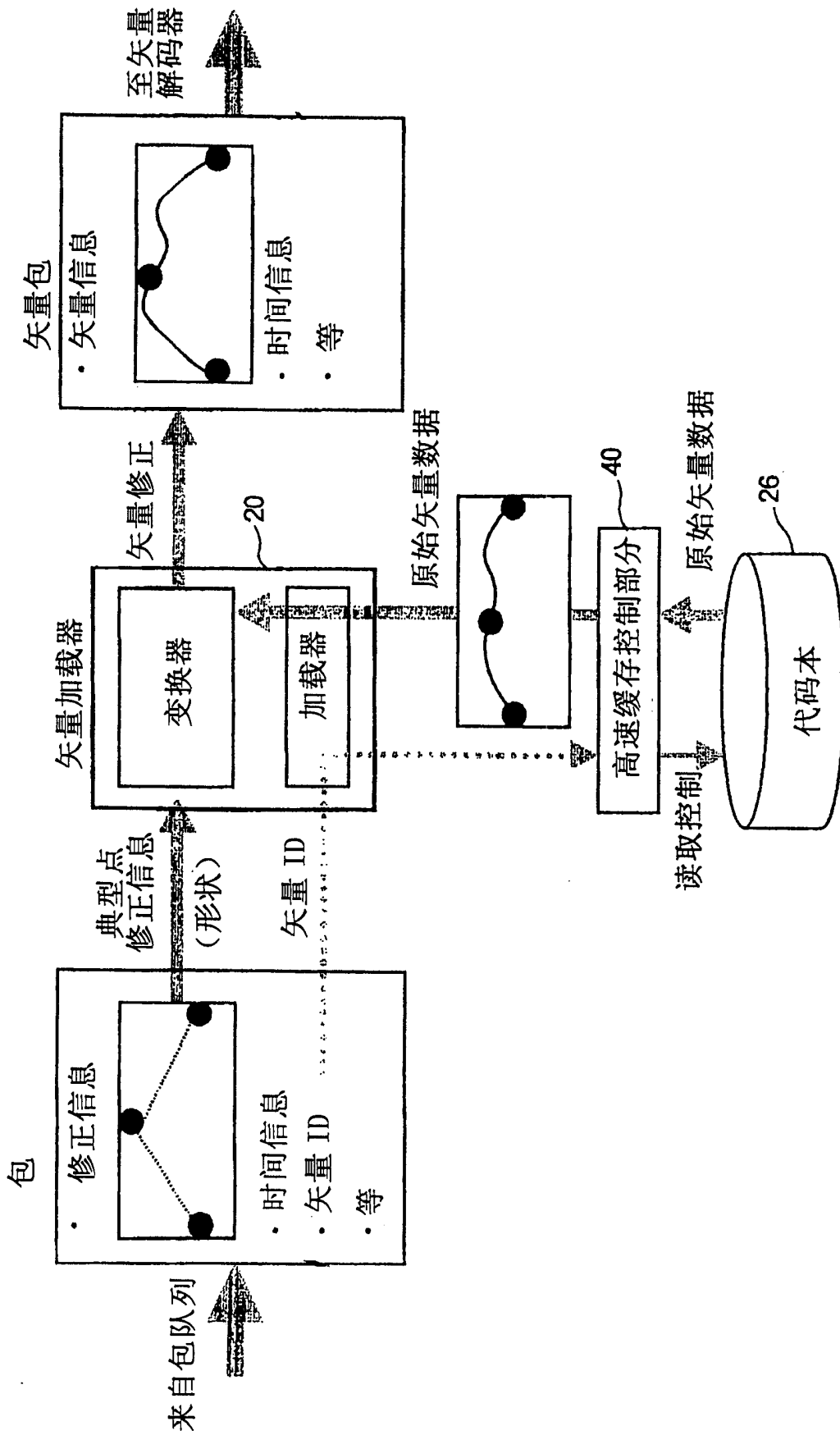


图 17

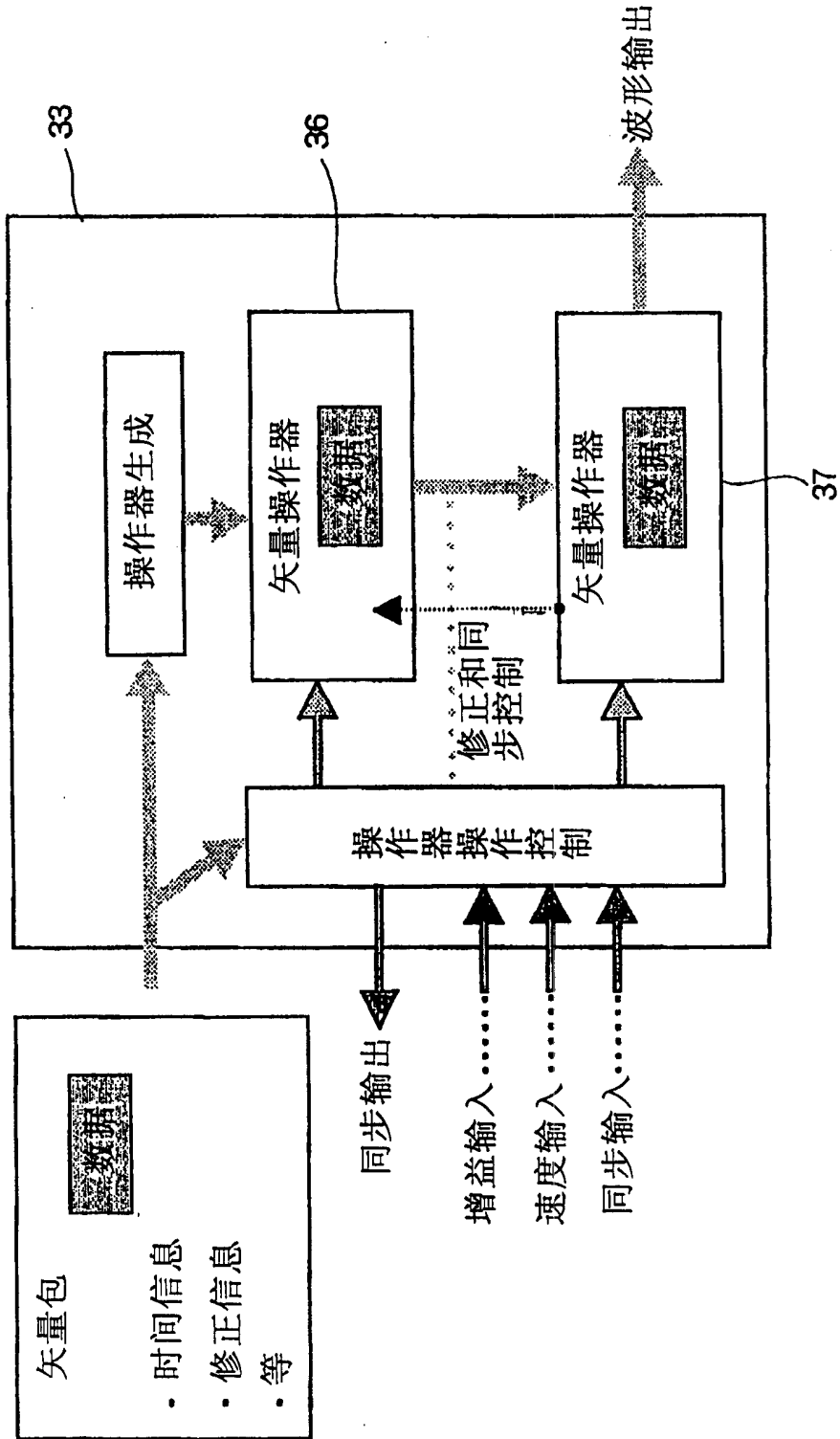


图 18

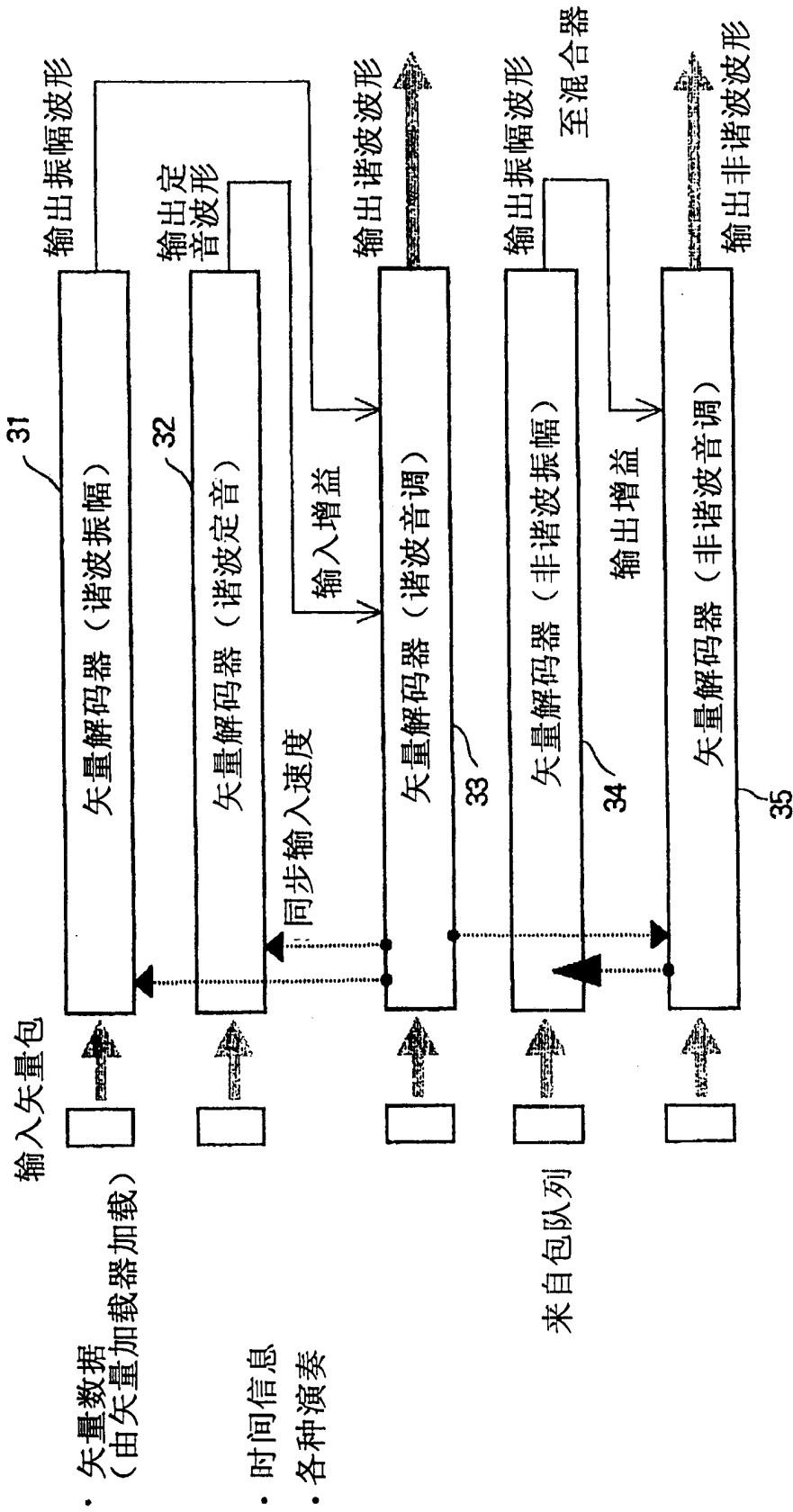
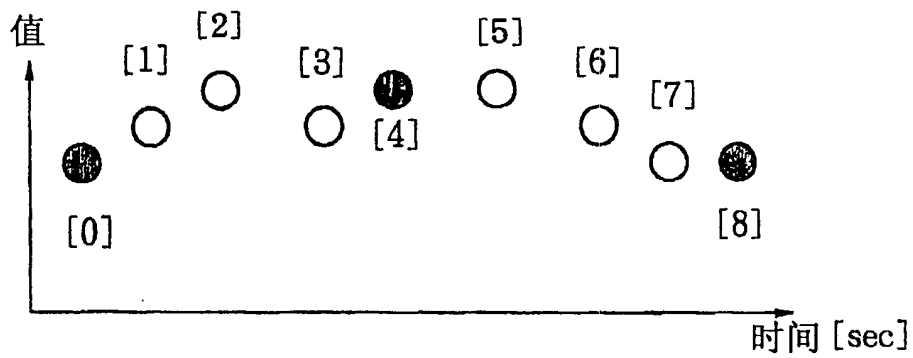


图 19

- 矢量数据 (由矢量加载器加载)

- 时间信息
- 各种演奏



矢量点结构阵列

典型的点阵列

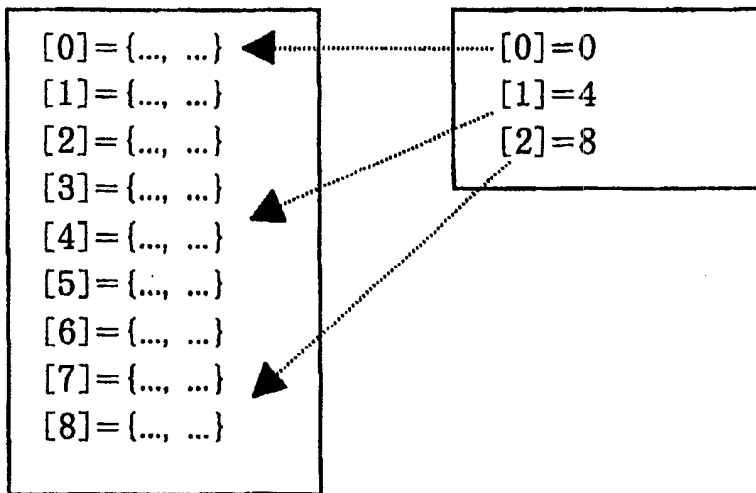


图 20

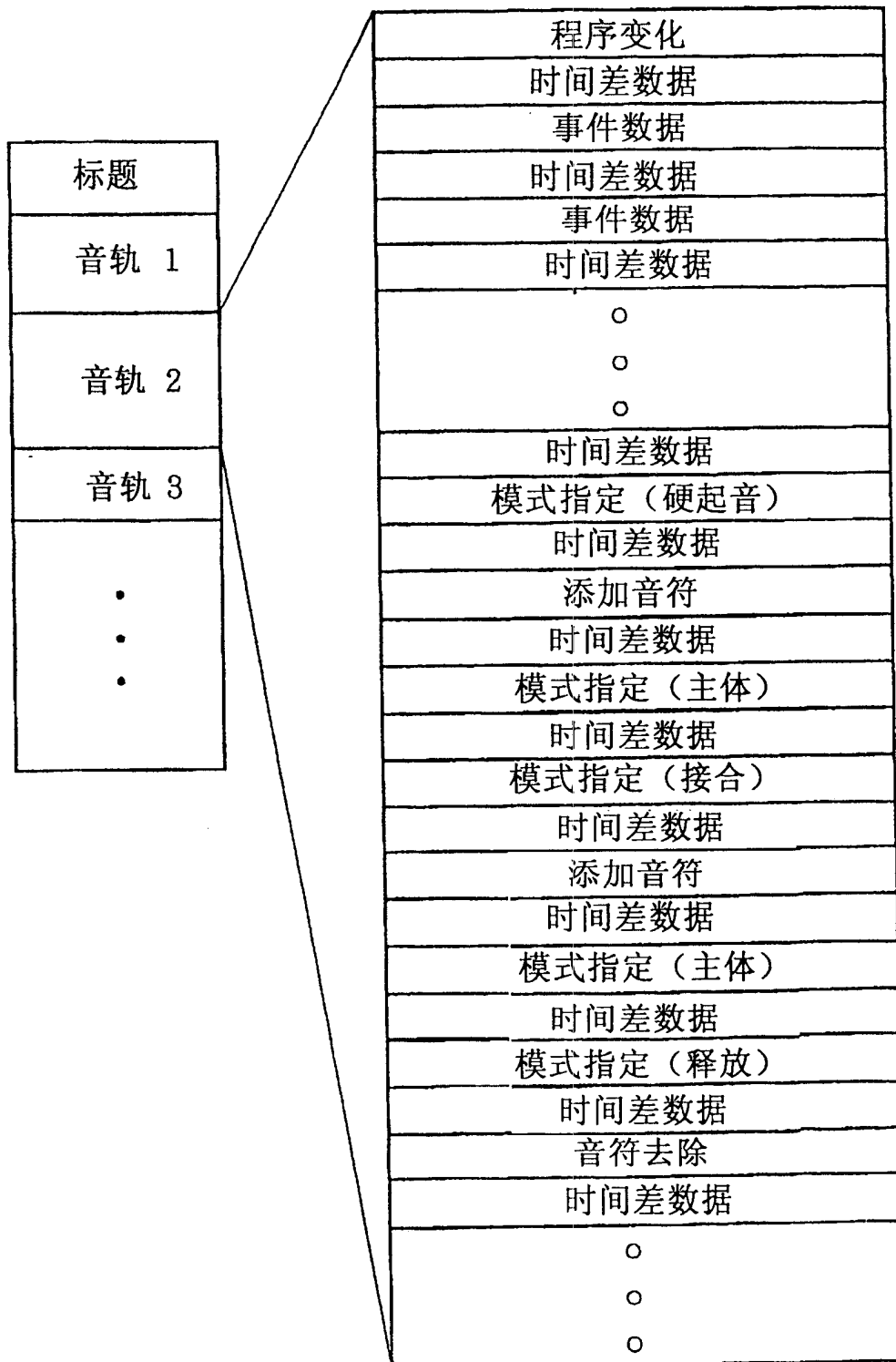


图 21

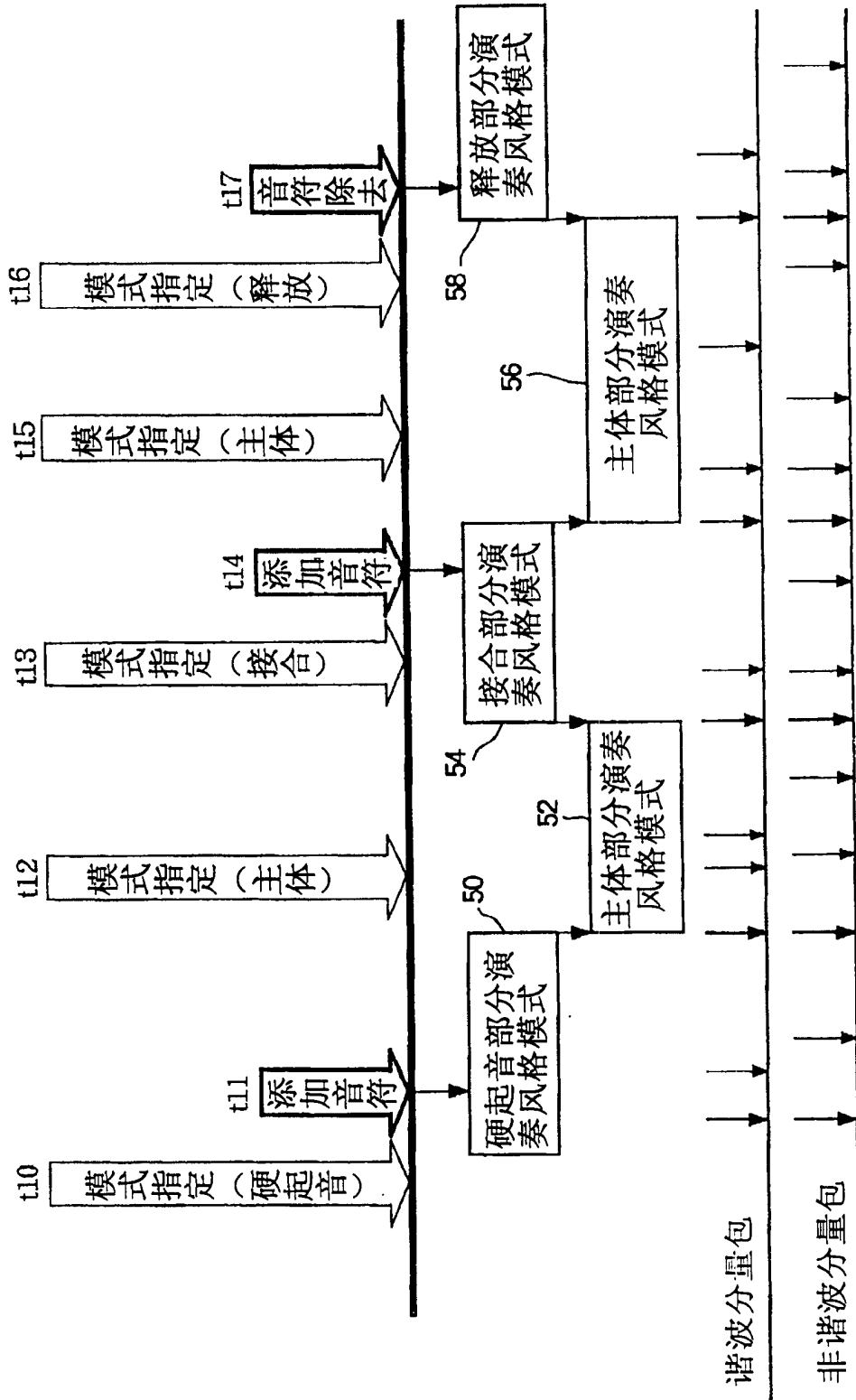


图 22

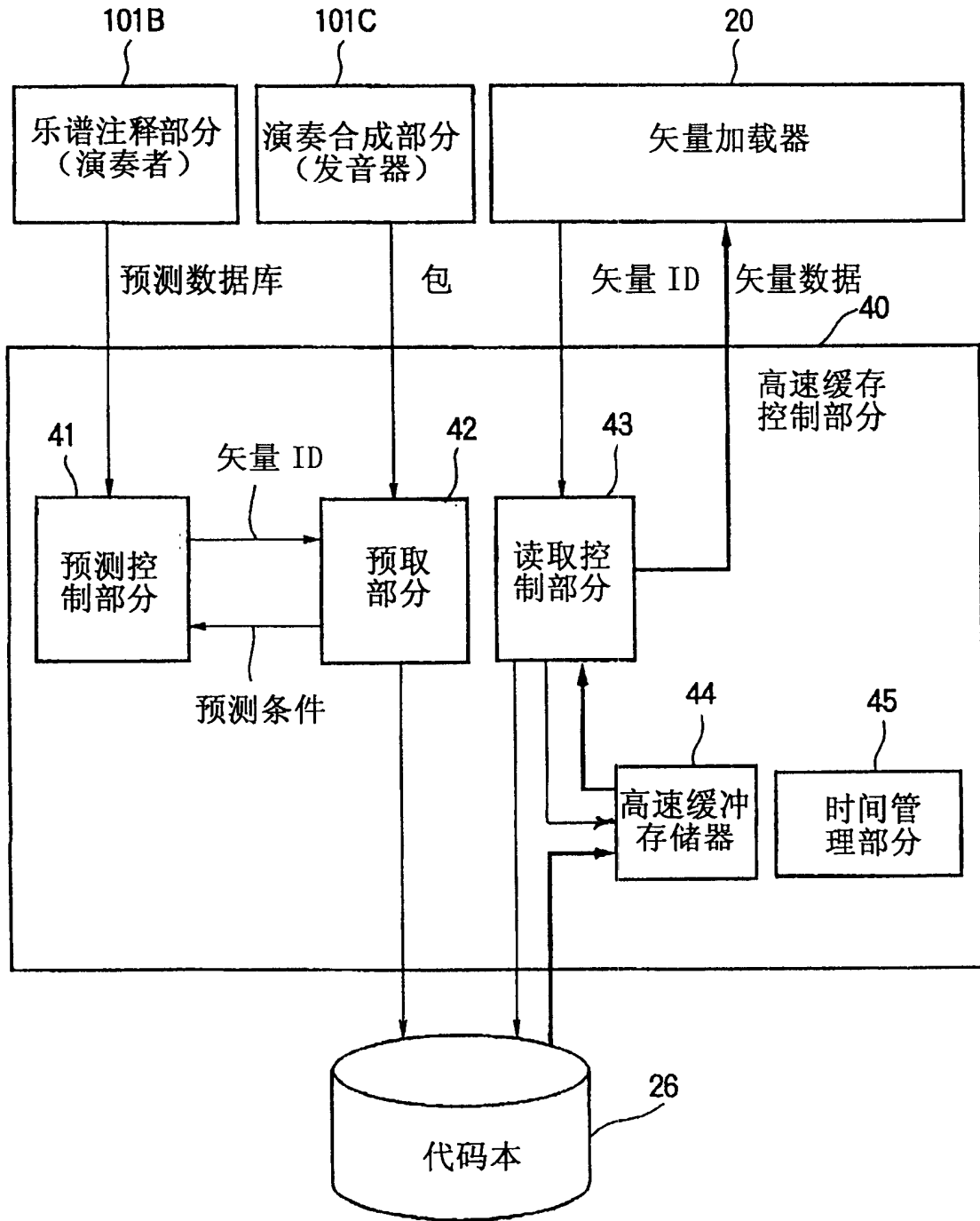


图 23

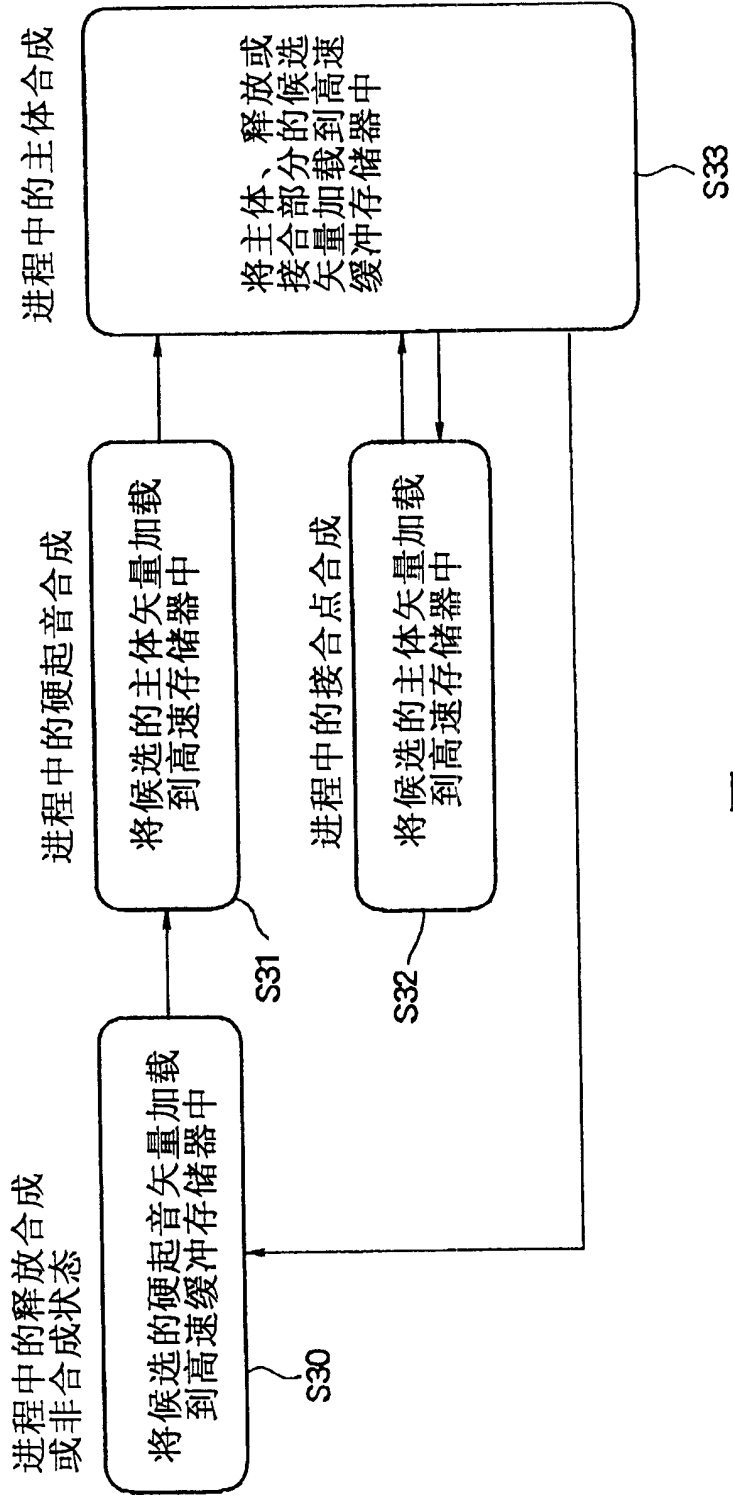


图 24

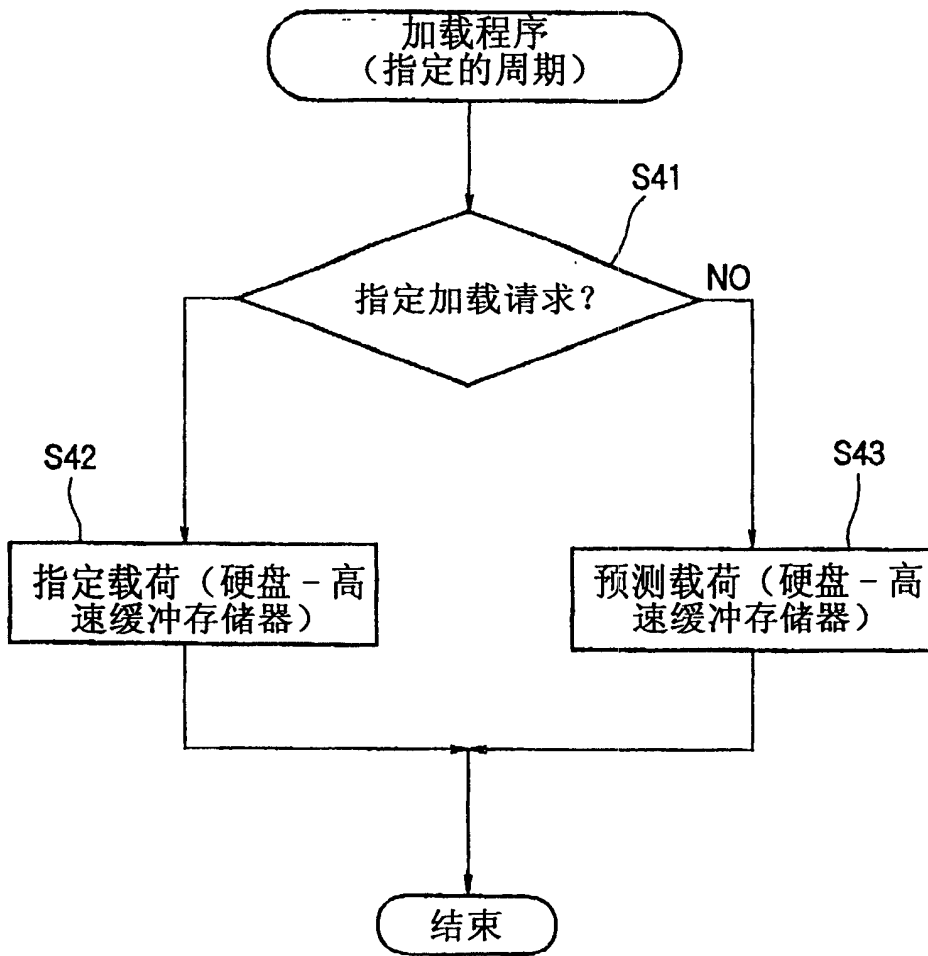


图 25

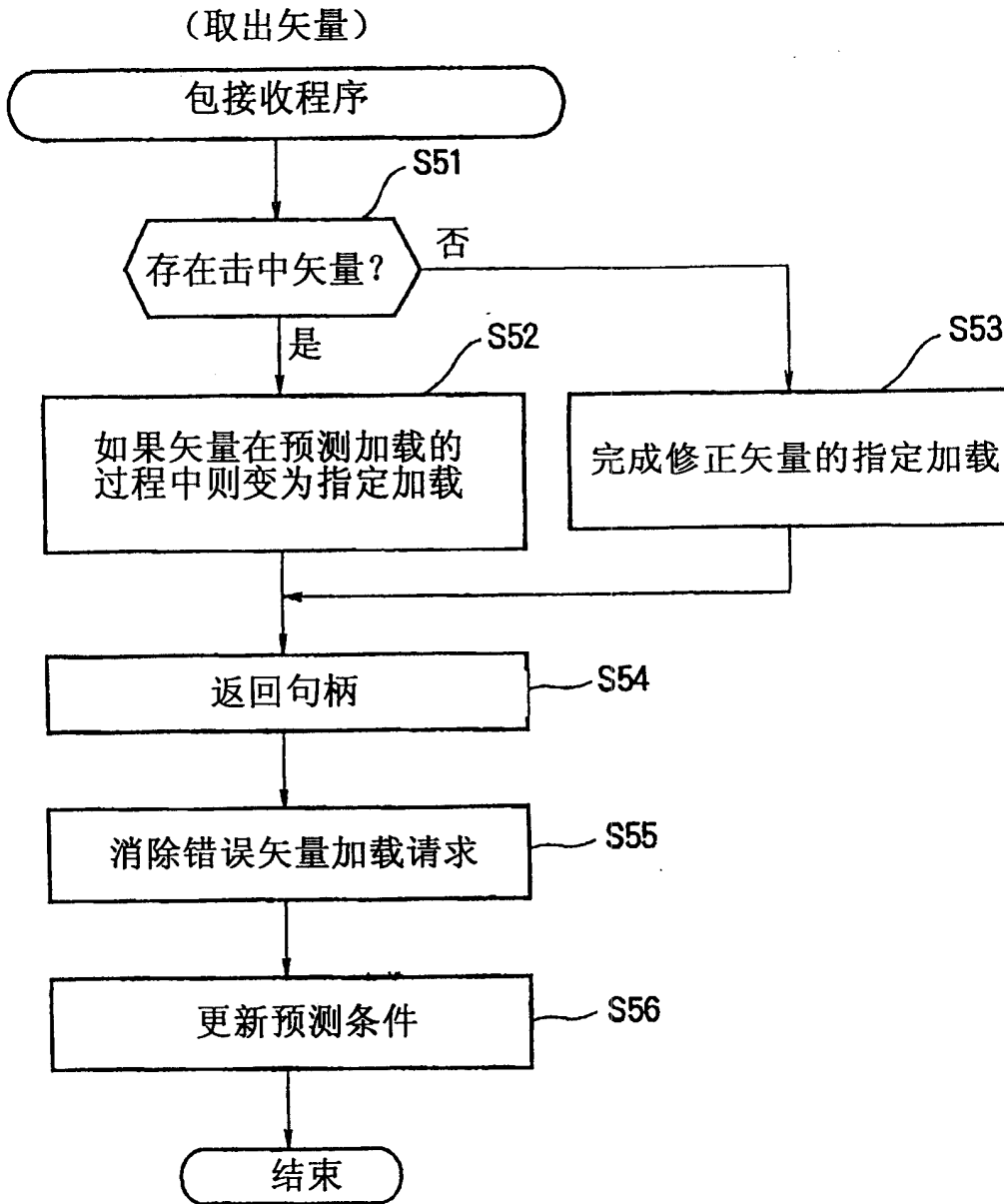


图 26

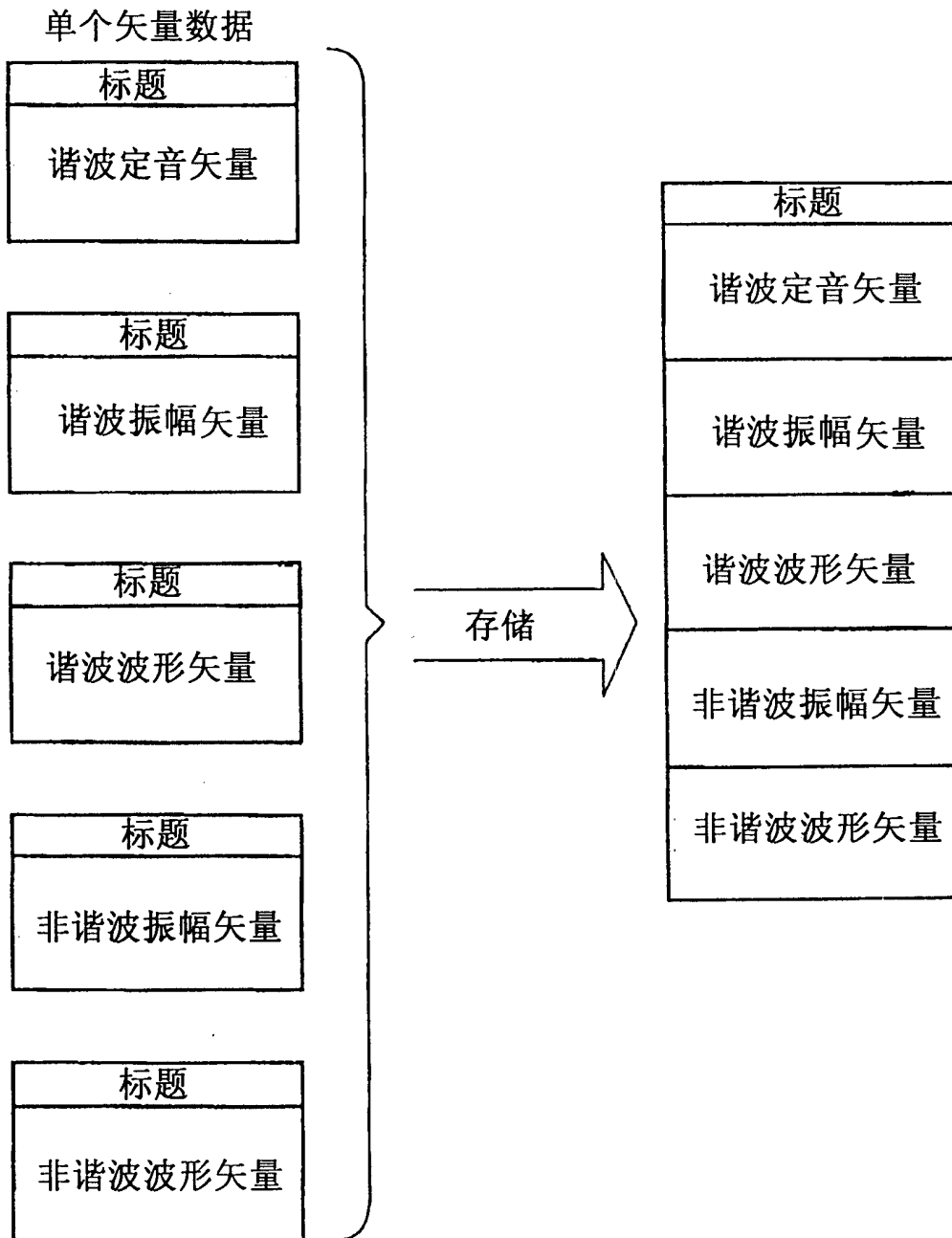


图 27

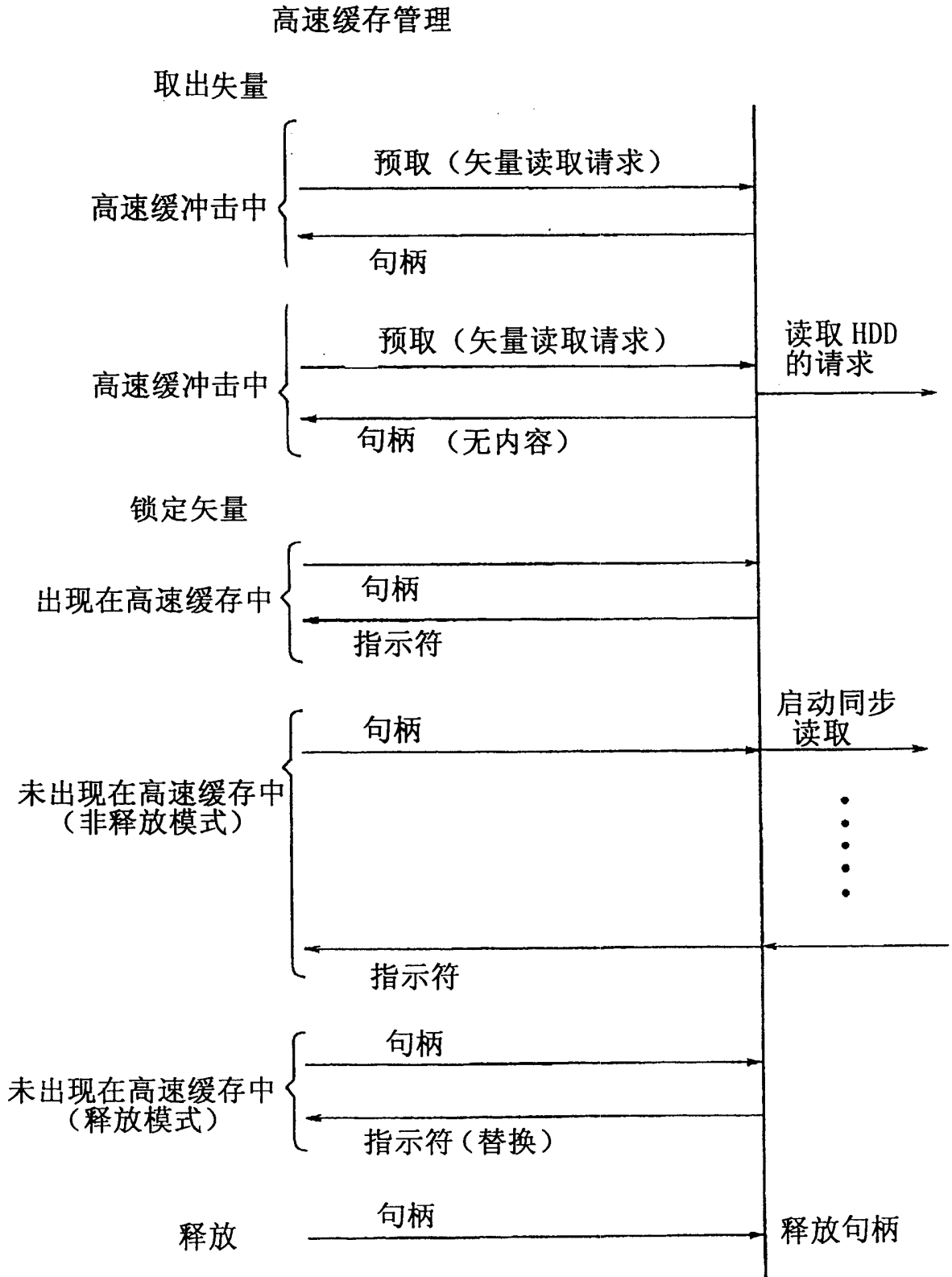


图 28

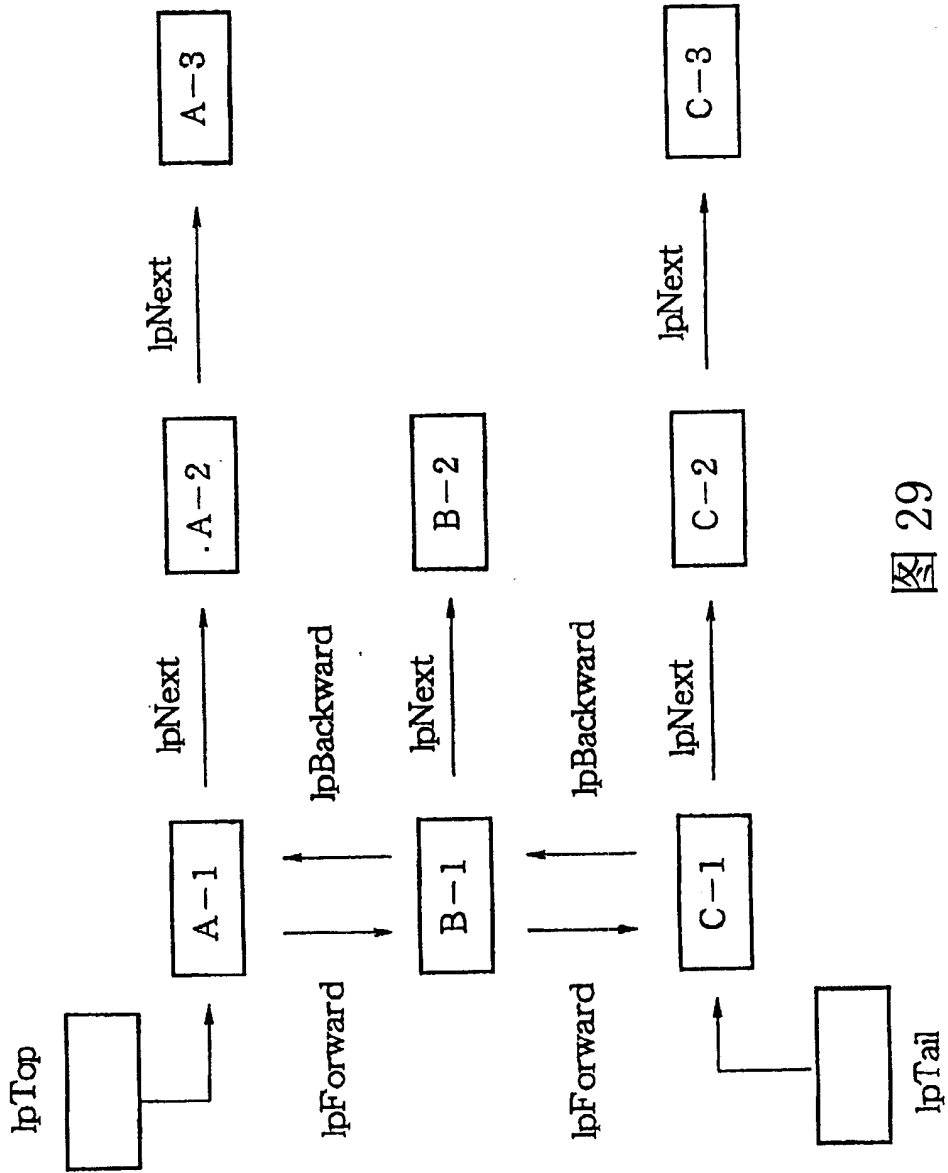


图 29

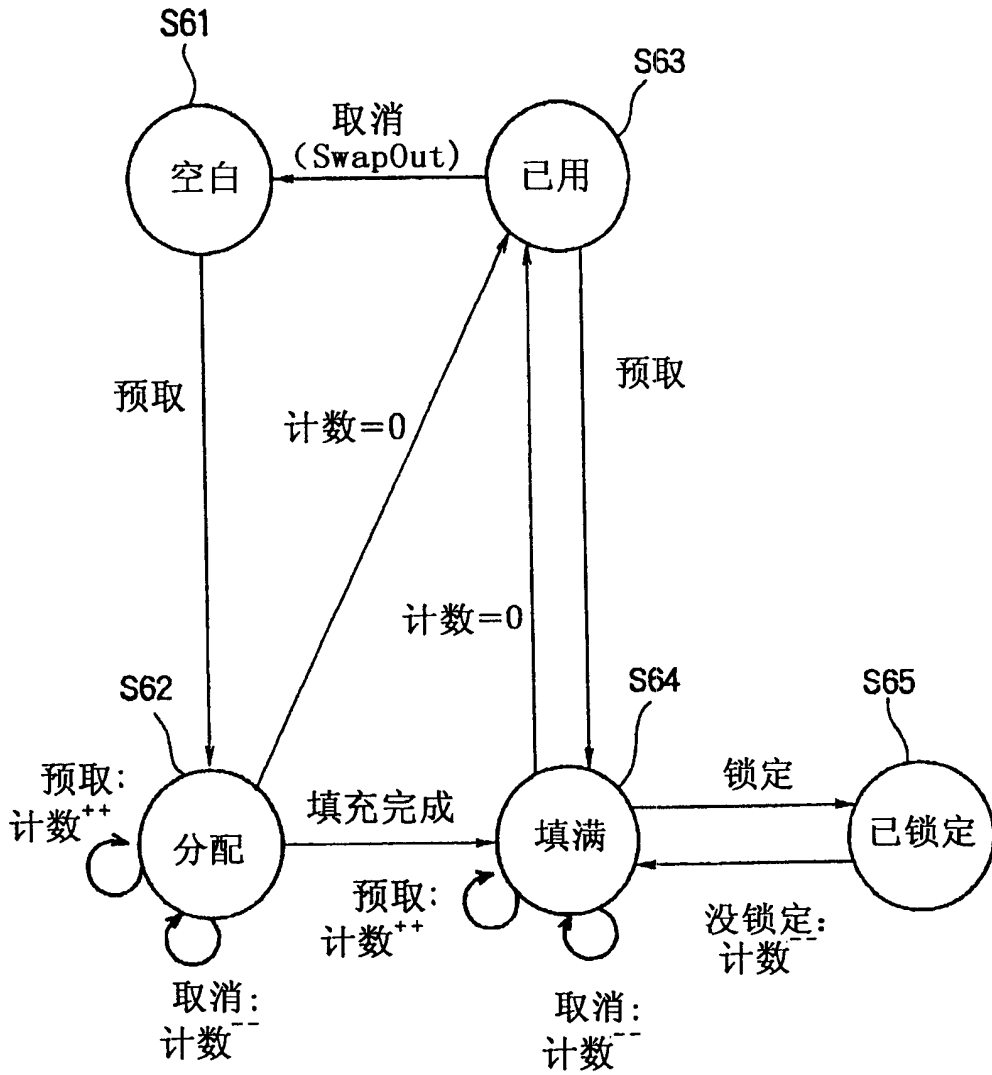


图 30

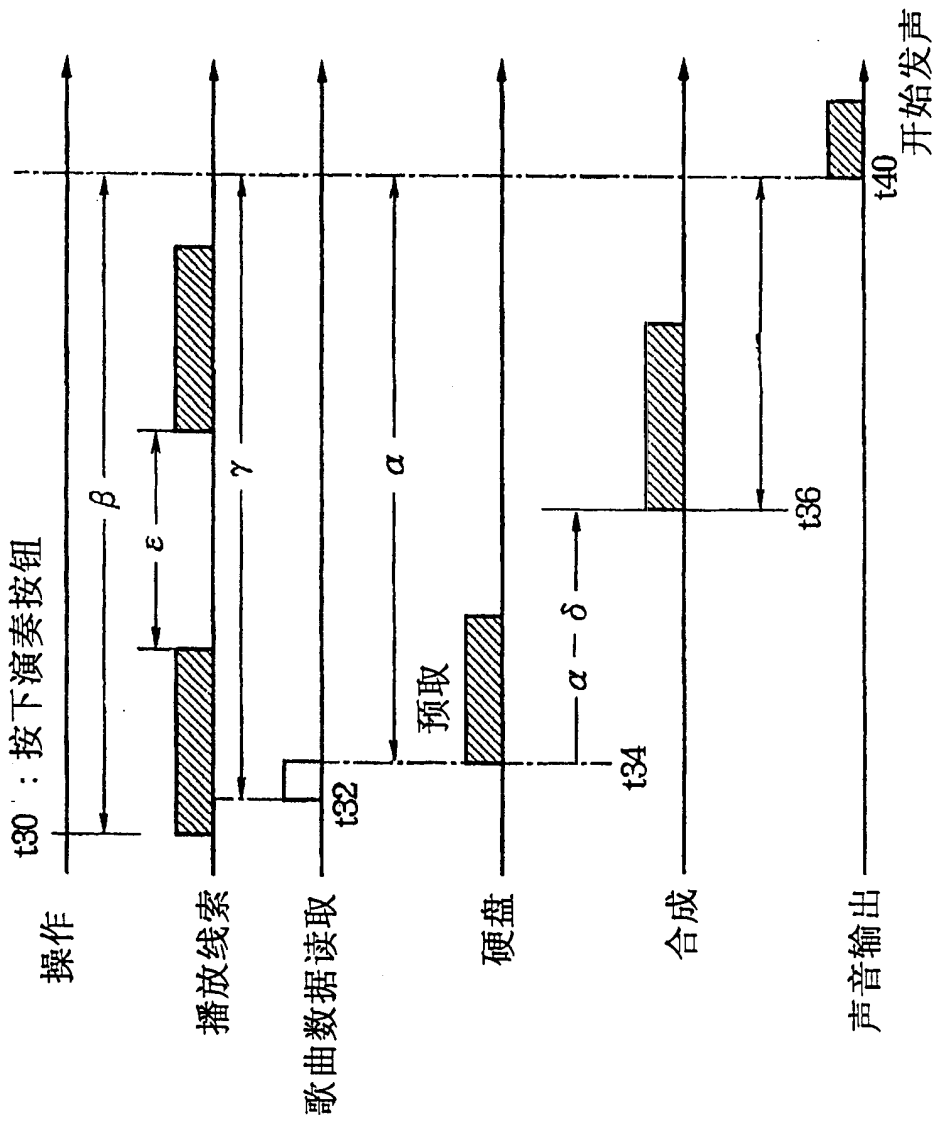


图 31