(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 122 606 B1

(12) # EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
02.10.2013 Bulletin 2013/40

(21) Application number: **08724644.3**

(22) Date of filing: **18.01.2008**

(51) Int Cl.:
G10H 1/22 *(2006.01)*

(86) International application number:
PCT/US2008/000718

(87) International publication number:
WO 2008/094415 (07.08.2008 Gazette 2008/32)

(54) **REAL TIME DIVISI WITH PATH PRIORITY, DEFINED NOTE RANGES AND FORCED OCTAVE TRANSPOSITION**

ECHTZEIT-DIVISI MIT PFADPRIORITÄT, DEFINIERTEN NOTENBEREICHEN UND ZWANGSOKTAVENTRANSPOSITION

DIVISI EN TEMPS RÉEL AVEC PRIORITÉ DE TRAJET, GAMMES DE NOTES DÉFINIES ET TRANSPOSITION D’OCTAVE IMPOSÉE

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT RO SE SI SK TR

(30) Priority: **18.01.2007 US 885410 P**

(43) Date of publication of application:
25.11.2009 Bulletin 2009/48

(73) Proprietor: **The Stone Family Trust Of 1992**
Hidden Hills, California 91302-2416 (US)

(72) Inventors:
- **STONE, Christopher**
  Hidden Hills, California 91302-2416 (US)
- **DAVIS, Gary**
  West Hills, California 91307-1534 (US)

(74) Representative: **Kirschner, Klaus Dieter et al**
Puschmann Borchert Bardehle
Patentanwälte Partnerschaft
Bajuwarenring 21
82041 Oberhaching (DE)

(56) References cited:
EP-A1- 1 617 406
US-A- 5 369 224
US-A- 5 850 051
US-A- 5 895 877
US-A1- 2006 236 848
US-A1- 2006 236 848
US-B1- 6 582 235
US-B2- 7 109 406

EP 2 122 606 B1

Description

[0001] This system and method generally relates to the field of processing live or sequenced musical notes utilizing a form of divisi as set forth in US- A- 7, 109, 406 and as further developed in PCT/US06/17757, adding multiple semi- automatic functions that may be used to attain real time orchestration of a virtual ensemble of virtual musical instruments during a live or sequenced performance.

[0002] The system and method for manipulation of sampled or synthesized sounds described herein relates to the live or sequenced playback of orchestral sounds, choirs, or any type of music. This affects isolated notes as well as individual or multiple notes which may be part of or entirely comprising a musical chord.

[0003] Sampled musical instruments have absolute ranges which correspond to the physical playable pitch range of the original instruments used to create the recording from which the samples were generated. While these pitch ranges can be artificially extended by various means of pitch shifting, the results of pitch shifting tend to be less than sonically realistic and often sample libraries do not rely upon this technique to produce more than a few semitones of pitch shift. Some pitch shift is acceptable sonically, and use of this technique allows sample libraries containing an notes witnin tne aosoiuie range of a given instrument to be built from recordings of every third or fourth note of that instrument, for example, thus conserving time in library creation as well as reducing the required storage memory and other sample playing resources. It is possible to restrict the playable range of a sampled instrument to be less than the absolute playable range of the acoustic instrument by either not using or by blocking access to notes above some upper limit or below some lower limit, or by allowing only specific pitches to be played. Such restrictions may be done to attain sonic improvements, power balancing, specific orchestrational goals, or for other reasons. In any case, all sampled instruments, also known as virtual instruments, have playable ranges. Within the MIDI specification, there are 128 defined notes, although the largest MIDI keyboard commercially available has a total of 88 black and white keys (corresponding to the 88 notes of the chromatic scale) much like a typical grand piano keyboard. The playable range of any specific virtual instrument may be fewer than 88 notes. Thus it is possible to play notes when using a MIDI keyboard (or MIDI sequencer) that exceeded the playable range of a given virtual instrument in which case, with prior art systems, either no sound will be produced or, if sound is produced, it will be stretched upward or downward as necessary, typically using a pitch bend method that alters the playback sampling rate to artificially extend playability beyond the absolute sampled note range. Such stretching risks the aforementioned sonic defects, which include too-fast or too-slow attack and decay of the sound, audible discontinuities, clicks or similar glitches, and an inappropriate (unrealistic) harmonic overtone structure.

[0004] The use of the orchestration technique known as divisi, such as set forth in US- A- 7, 109, 406, allows played notes to be allocated amongst an available pool of musicians, in this instance an available pool of virtual instruments played by virtual musicians. The original method described in detail in mat patent spec was oase [alpha] iargeiy on lookup tables, although algorithmic methods were also mentioned as a viable alternative. PCT/US06/17757 went on to detail examples of an algorithmic method of accomplishing divisi, and further expanded the method by adding, among other nuances, a method of prioritizing the allocated note paths such that sequentially played notes adding to a chord could be caused to invoke different instruments or groups of instruments depending upon a set priority. Given these methods, what then is divisi and why is it used? In the way of review, the Italian term divisi generally refers to the orchestral allocation of notes amongst a given section of string players such as the first violins, second violins, violas, celli or basses. When a single note is to be played by one of these string sections, for example, all musicians in that section play the same note. When a chord of two or more notes is to be played by one of these string sections, the available musicians split up or divisi themselves so that some musicians play one note, some another, and so forth. Without going into the orchestrational rules here, suffice it to say that sometimes the division of notes amongst available players is even, sometimes not, and when the division is not even more musicians will be playing either the higher or the lower pitched note (s) depending on the desired effect, which preference the author refers to as top weighting or bottom weighting. That's the simple explanation of traditional string divisi.

[0005] Divisi, however, can be abstracted up a level to cover more than just string sections; the same principle can be used to cover allocation of notes among various available individual instruments in the orchestra, among ad- hoc groups of instruments, or even among pitched and unpitched sounds of any description. Musically, however, it would not necessarily sound good to simply split up chords to feed various instruments willy- nilly. So in setting up a system whereby notes and chords can be automatically orchestrated by some implementation of divisi allocation methods, the user should be allowed to make decisions as to which instruments will play first and which instruments come in subsequently as a chord is arpeggiated (that is, as notes are played individually or as they are added to an already playing note or chord) . As well the user should be able to make decisions as to which notes are allowed to be allocated to various instruments (or stems of instruments) based on defined playable ranges for each instrument.

[0006] With any given set of available instruments, their playable ranges may or may not overlap, and even if the ranges do overlap, the span of notes wherein they overlap may vary from as few as 1 to as many as all playable notes. A method of allocating notes played to the set instrument ranges could thereby produce very different sounding results depending on the actual ranges set for the various instruments. Too, it is possible that certain notes may fall outside the playable range of any available instruments, either higher than the highest playable note, lower than the lowest playable note, or in a hole between the playable note ranges of non-overlapping instrument playable ranges. To address the potentiality of a non-playable note, the author has devised a method whereby non-playable notes can be automatically transposed by one-octave increments such that they can be allocated to whatever available instrument has the nearest (by pitch) playable note range that can accommodate the transposed note.

[0007] The ultimate divisi of incoming notes and chords to available instruments (which reference here also includes paths or stems of multiple instruments) will thus depend upon how the user sets up the path priorities, the playable ranges for each path, and the options to transpose notes up or down in pitch, in octave or other desired increments, so they remain within playable ranges. The benefits of the described method which applies these orchestrational principles to create a divisi amongst various instruments include enabling the user to play anywhere on an 88-key or smaller MIDI keyboard while preserving a pleasing, well balanced orchestration that will be playable by flive musicians using actual acoustic instruments that correspond to the sampled (or synthesized) virtual instruments being controlled by the described system, and as well the ability to generate discrete streams of MIDI notes which can be transferred to conventional prior-art musical notation systems for immediate conversion to playable parts or scores that are sufficiently well orchestrated that do not cause live musicians to have to play too-wide intervals or exceedingly difficult if not impossible to play jumps between subsequent notes as may occur with MIDI compositions that are created using conventional prior art sampler or synthesizer systems and note handling methods.

[0008] US 2006/236848 discloses a method and system for assigning notes to be played by a musical synthesizer to a predetermined number of instrument voices available to be sounded by said musical synthesizer, so that the musical synthesizer may emulate the sound of a live orchestra or other ensemble. The method includes the steps of building an array based on the number of notes to be played and the number of instrument voices available to play such notes, and allocating notes to the voices pursuant to algorithmic determination. As notes are released or newly played, all notes are dynamically reassigned to instrument voices so that, to the extent practicable, all channels play almost all the time. Additional methodology provides for correct assignment of notes across multiple different sections (or types) of instruments for purposes of real time orchestration.

[0009] The invention provides a note assignment processor and method whereby each available instrument or group of like instruments can be assigned a playable note range which affects how notes are allocated among said instruments or groups of instruments. Additionally, the method can automatically transpose notes that would otherwise be out of the playable range of the virtual instruments into playable ranges for said instruments in a way that preserves the original melodic intent.

[0010] For this purpose, the note assignment processor of the invention comprises the features of claim 1, and the methods of the invention comprise the features of claims 9 and 10. Preferred embodiments of the invention are characterized in the sub-claims.

[0011] Embodiments of the invention are now described with reference to the drawings.

[0012] Fig. 1 illustrates a simple divisi of all five paths in a string section comprised of First Violins (Vln.1), Second Violins (Vln. 2), Violas (Via.), Celli (Vc.) and Basses (Cb.).

[0013] This is a level 2 divisi (DVZII) which means it is within a section of like instruments as contrasted to a level 1 divisi (DVZI) which is more global, affecting different types of instruments. Middle C is referenced by an arrow and, dotted notes are being played.

[0014] Fig. 2 illustrated a DVZII involving three notes and four of the 5 paths of the string section.

[0015] Fig. 3 illustrates a DVZII involving three notes and three of the five String Section paths.

[0016] Fig. 4 illustrates a DVZII whereby the number of notes (3) exceeds the number of paths which are able to play them (2) because the notes are above the playable ranges of three of the five paths.

[0017] Fig. 5 illustrates a more complex DVZII whereby some notes are within range of multiple paths, but not of all paths, and choices must be made as to how to allocate notes where they might go to various paths

[0018] Fig. 6 illustrates the set of notes shown in Fig 5, abstracted to a two-dimensional matrix, which is part of the actual method by which we solve for note allocation in this divisi process, showing the multiple possibilities where various notes might be assigned to various paths based on playable ranges.

[0019] Fig. 7 illustrates the matrix of possible playable notes by the different paths, per Fig. 6, as depicted with a multi-keyboard representation.

[0020] Fig. 8 illustrates a solution of the matrix per Fig 6, using a method which assures proper distribution and top weighting per the divisi principles in the referenced patent and PCT, with actual notes allocated per path shown in black, possible but not allocated notes shown in gray.

[0021] Fig. 9 illustrates a solved matrix of Fig. 8, now depicted as an orchestral outcome on a multi-keyboard representation, with actual notes played by each path in black, notes that might have been played (e.g., they were within playable range) but were assigned elsewhere shown in gray.

[0022] Fig. 10 illustrates a solution to the matrix similar to Fig 8 but done with bottom instead of top weighting.

[0023] Fig. 11 illustrates a solved matrix of Fig. 9, now depicted as an orchestral outcome on a multi-keyboard representation.

[0024] Fig. 12 illustrates a two-note divisi among a section of eight single-player first violin paths, showing how equal sound power is maintained for each note through even allocation of 1 note to each of four players. The parenthetic numbers (1) adjacent to each desk number indicate there is one player (one musician) per each desk.

[0025] Fig. 13 illustrates a two-note divisi among a section of eight first violin paths, four of which have single players (1) and four of which have two players(2) per desk. Here more of the higher note allocation is different from that in Fig 12 in order to maintain correct power balance; 6 musicians are playing the upper note and 6 musicians are playing the lower note even though 5 paths play the upper note and 3 paths play the lower note.

[0026] Fig. 14 illustrates an example of what may happen when no notes fall within range of a given path, this five-note divisi is applied to five paths. However no notes are within the playable range of the violas (Vla.) and so they are excluded with no notes assigned to them.

[0027] Fig. 15 illustrates the set of notes shown in Fig 14, abstracted to a two-dimensional matrix (on the left) and the solution to that matrix (on the right) with actual notes allocated per path shown in black, possible but not allocated notes shown in gray.

[0028] Fig. 16 illustrates the matrix of possible placements and the solution thereof for the same notes shown in Figs 14 and 15, but in this instance a Force Octave Shift Down function is set for the violas. Notes that previously would not have been playable by the violas are now generated through downward transposition, as indicated by the dark cross hatch and light cross hatch boxes in the viola (Vla.) rows.

[0029] Fig. 17 illustrates the solved matrix of Fig 16 (right), now depicted as an orchestral outcome on a multi-keyboard representation, with actual notes played by each path in black. Unlike Fig 14 where the violas had no notes to play, they now have a note due to the force octave shift down function being set for this path.

[0030] Fig. 18 illustrates the matrix of possible placements and the solution thereof for the same notes shown in Figs 14 and 15, but in this instance a Force Octave Shift Up function is set for the violas.

[0031] Fig. 19 illustrates the solved matrix of Fig 18 (right), now depicted as an orchestral outcome on a multi-keyboard representation.

[0032] Fig. 20 illustrates that it is possible to set both the Force Octave Shift Up and Force Octave Shift Down functions for any given path, and this illustration shows such a situation for the violas, given the same notes played and ranges as in the previous several illustrations. More possibilities exist for viola note allocation (left side) and while the solution (right) still gives them a single note to play as occurred in Fig 18, the overall DVZII note allocation is different here.

[0033] Fig. 21 illustrates the solved matrix of Fig 20 (right) now depicted as an orchestral outcome on a multi-keyboard representation.

[0034] Fig. 22 illustrates Level 1 divisi (DVZI), using the concept of priorities, per our referenced patent/PCT filings. Each path's set priority is shown in a box to the left of its keyboard. This and Figs 23 through 25 all show how priorities work when all the notes are within the playable range of the available paths. Here there are five paths, four different priorities set, and a single note played.

[0035] Fig. 23 illustrates DVZI with five paths, four different priorities set, and two notes played.

[0036] Fig. 24 illustrates DVZI with five paths, four different priorities set, and three notes played.

[0037] Fig. 25 illustrates DVZI with five paths, four different priorities set, and four notes played.

[0038] Fig. 26 illustrates DVZI when priorities conflict with playable ranges. The Cello path (Cb) is set to priority 1 but the note is out of that path's playable range, so instead it is allocated to both of the paths which are set to priority 2, in this case the first and second violins (Vln. 1 and Vln. 2)

[0039] Fig. 27 illustrates two notes that are played and both are within range of the Celli, while neither is in range of the first or second violins; given the Cello path is Priority 1 they get the lowest note, but priority 2 is skipped due to the out-of-range condition so the second note goes to the violas whose path is priority 3.

[0040] Fig. 28 illustrates an examination of the interaction between priority and force octave shift functions. Here the Cello (Vc.) path is priority 1 and it has the force octave shift up set. The first (and only) note played is below the cello range and would otherwise be played by the basses (Cb.) but is instead transposed up an octave and given to the Cello path because it is now within range of this priority 1 path due to the force octave shift up function being set.

[0041] Fig. 29 is an example where all paths have both the Force Octave Shift Up and Down features set, and the system is set to Top Weighting. Two notes played below the range of the priority 1, 2 and 3 paths are thereby transposed and allocated to priority 1 and 2 paths in this example. That is, notes in the bass range instead go to the violins and cello.

[0042] Fig. 30 is similar to Fig 29 except now bottom weighting is set instead of top weighting. For this reason the lower note (D) is played by two stems instead of the higher note (E) as was done in Fig 29.

[0043] Fig. 31 illustrates an example of what happens with a mix of Force Octave Transpose settings and out-of-range notes. Here Force Octave Shift Up is set for the first violins (Vln. 1) and the Celli (Vc.) paths, but not for the second violins (Vln. 2). While both the first and second violins have the same Priority 2 value, the second violins cannot play either of the two notes since they remain out of range when their path's Force Octave Transpose Up feature is not set.

[0044] Fig. 32 is another example of the interaction between priority and Force Octave Shift functions; here the

Force Octave Shift Up is set only for the Cello (Vc.) and Bass (Cb.) paths but for no others. With one of the two notes out of range for the Bass, and Priority 1 favored for the Cello path, a single transposition occurs to bring the upper note higher and into the Cello range.

[0045] Fig. 33 is an example where the priorities and Force Octave settings are the same as in Fig 32, but the one note (E) that is within range of the bass is the higher of the two original notes. Since transposing only the out-of-range (for the cello) D into the cello path would violate the rule of keeping the notes in the order played, both notes are transposed up an octave.

[0046] Fig. 34 illustrates an example of a processor for executing the method according to an embodiment of the invention.

DETAILED DESCRIPTION

DVZII With Crossovers (Playing Ranges)

Overview

[0047] For simplification of reference, we use the initials DVZ to represent the divisi note allocation process in general. The term DVZI refers to a Level 1 divisi (the highest or most global note allocation), and the term DVZ-II refers to a Level 2 divisi (an allocation of notes to multiple players or desks within a single Level I divisi path). The following terms are either defined in the text as we proceed or evident by context: path, stem, player, voice, playable range, and crossover.

[0048] Each instrument or each path which addresses instruments can have a playable range, a span of notes which the instrument (or group of instruments) is capable of playing. The term crossover refers to what happens when notes fall outside the range of a given instrument or path, and are instead allocated to another instrument or path where the note are within the playable note range. For simplicity (fewer words), we sometimes use the term crossovers more-or-less interchangeably with playable note ranges even though, technically, they are not precisely the same thing. It should be understood by context what is meant here.

[0049] In order two understand DVZI with crossovers, first we should look at the simpler of the two methods or algorithms - simpler because it does not involve priority values - the DVZII with crossovers. The idea of DVZII is to keep the number of voices playing constant no matter how many notes are playing. Imagine, we have eight instruments. If we play one note, each instrument plays that note for a total of eight sounding voices. If we play two notes, the first note is played by the first four instruments, and the second note is played by the second four instruments for a total of eight sounding voices. If we play four notes, each note is played by two instruments for again a total of eight sounding voices.

[0050] DVZII with crossovers uses the same principle, but the playing ranges of the instruments involved are taken into account. In most cases, not all instruments involved in a DVZ will be able to participate. Some instruments may not have any notes in their playing ranges. When we perform a DVZII with crossovers, the goal is to distribute notes evenly across all the instruments participating. Fig. 2 shows the distribution of a three note chord that falls outside of the playing range of the bass. Fig. 3 shows the distribution of another three note chord that falls in the playing range of the top three path: violins I and II and violas.

[0051] When the number of notes exceeds the number of paths involved in the DVZ, multiple notes must be assigned to a single path. In Fig.4 Violins I is allocated two notes; if this path represents an eight chair violin section, then the first four violins would play the first note and the second four would play the second note. If Violins I were a single player, the two notes would be played as a double stop. Fig. 4 illustrates this distribution.

Distribution based on limitations of playing range

[0052] Fig. 5 illustrates a slightly more complicated five note DVZII. If we didn't take playing ranges into account, we would expect each note to be played by one path - this would be the optimal distribution of notes for a DVZ. Unfortunately, the playing ranges of the instruments prohibit such a uniform distribution. Instead, because the third note from the highest is above the range of the third path (Violas), the Violins I have to take two of the notes, and the fourth note has to be doubled and given to both the violas and the cellos.

[0053] When playing ranges prohibit use of an ideal distribution, we determine how notes will be orchestrated by abstracting the problem to a two dimensional matrix. Fig. 6 shows the all possible placements of the five notes with the instruments we are using, and their ranges, per Fig. 5. Horizontal rows represent instruments, and vertical columns represent notes as indicated by the labels next to them. If a note falls in the playing range of an instrument, we set the box (here shown in black) in the matrix at the row and column representing that note and instrument. For clarification, Fig. 7 shows this matrix of possible note placements on keyboards.

[0054] To obtain our actual DVZ note distribution we solve the matrix. This usually looks like finding the straightest path from the top left corner of the matrix to the bottom right corner, although the method is more complex than such a convenient conceptualization, involving a number of iterative processes. Then we translate this matrix-based solution back onto paths that correspond to the instruments, as shown in Fig. 9.

Weighting

[0055] There are often cases, as we have illustrated above, where an odd distributions of notes occurs, such as three notes split among the Violins I and Violins II or the two bottom notes split between the violas, cellos, and basses. How do we decided where to distribute notes in these cases? The DVZ process includes a method called weighting. DVZs can either be top-weighted or bottom-weighted. With top weighting, odd distributions lean towards the higher pitch range instruments (or the higher notes for a single type of instrument), giving more of those instruments the notes which cannot be evenly distributed. That is how we decided to give two notes to the first violins instead of the second violins and how we decided to let the violas and cellos share the fourth note rather than letting the cellos and basses share the fifth note. Bottom weighting is the inverse, giving the extra notes to the lower pitch range instruments (or lower notes within the range of a given set of instruments). Fig. 10 shows the solution to our matrix of possible placements with bottom weighting instead of top weighting. Fig. 11 shows this bottom-weighted matrix solution translated back into a keyboard representation of our orchestra.

Desks and voices

[0056] The inventive method of DVZ described in this application takes into account desks and voices. At the instrument level, typically a Level II divisi (DVZII) within the string section, a desk (or player) is another term for a path. A voice is the number of musicians sounding at a desk. Some of the violin samples are single musician players, some are two musician players - for example two people with violins recorded in one channel (or converted into one sample) while playing simultaneously in unison. Since a two-voice player will have more sound power than a single-voice player, DVZ takes into account the number of voices (musicians) per desk when distributing notes.

[0057] We'll consider a violin section to explore how desks and so forth are handled in the present DVZ process. In the example of Fig. 12, we have eight single-voice violin desks in a violin section - eight one-musician players. The number of voices at a given desk is indicated in parenthesis on the figure to the right of the desk's name. If each desk has only one voice, then we have eight voices. If we send this violin section two notes, each note will be played by four voices.

[0058] Fig. 13 shows a violin section with eight desks again, but in this case, the first four desks are single voice players, and the second four desks are two-voice players. This gives us a total of twelve voices if each desk were to play one note. To maintain an equal power (equal number of voices) per note, when we play two notes in this setup, each note should be played by six voices. So here the first note gets sent to the first five desks and the second note gets sent to the last three desks.

Forcing paths to play

[0059] We may want to ensure that all paths play regardless of what the input notes happen to be, even if apparently out of range of some paths. This perhaps would be useful in sections we want richly orchestrated, or if we want the entire string section to play a unison line. For these and other situations there are two new divisi parameters that can be set for each path: force octave shift up and force octave shift down. These parameters work in the following way. If there is a given path with no notes in range and that path has either of the force octave flags set, points in the matrix of possible placements will be filled in *before* performing the DVZ. The modified matrix is solved, and when converting the matrix back to MIDI, those notes assigned to paths that fall outside the playing range of those paths are transposed by whatever minimum number of octaves is necessary so they now fall within the range of the specified path.

[0060] The same notes played (input) as used in Fig. 14 will be the input for all examples in this section through Fig. 21. Fig. 14 shows a five-note DVZ. None of the notes lies within playing range of the violas, and neither force octave shift up nor force octave shift down is enabled. The resulting note distribution is shown by the dotted keys on the 5 string instrument paths. Fig. 15 shows the matrix of possible placements and its solution for this situation.

[0061] What if force octave shift down is set for the violas? Fig. 16 shows the matrix of possible placements and its solution for this situation. As illustrated, possible placements are borrowed from the immediately previous path, in this case Violins II, to fill in the viola's row. Then the matrix is solved. Fig. 16 shows the note distribution matrix with the possible note allocations on the left, and the solved transpositions and allocations on the right, using the setup described above. Compare this to the possibilities and solution without Force Octave Shift down, as illustrated in Fig. 15.

[0062] Now suppose force octave shift up were set for the violas. Possible placements would be borrowed instead from the cellos as seen in the matrix possibilities and solution of Fig. 18. Fig. 19 shows the note distribution with transpositions for the solution from Fig. 18.

[0063] If both force octave shift up *and* force octave shift down are set, we borrow both from the row above and the row below; the matrix and its solution for this situation are shown in Fig. 20. The note distribution with transpositions for the solution from Fig. 20 is shown in Fig. 21.

DVZI Priorities Overview

[0064] Paths can be given priority numbers such as 1, 2, 3 and so forth, and the same number can be given to more than one path; priority number values cannot be skipped, they must be contiguous. When priorities are set and fewer notes are played than the number of paths to which notes may be assigned, those paths with lowest numbered priorities will be first in line to play notes, those with higher numbered priorities then play additional notes. If one arpeggiates a chord, then successively higher numbered paths play successively played notes, increasing the number of instruments involved with each added note; this is not like standard DVZ where there are always a constant number of instruments playing. However, when there are as many or more notes being played as there are path priority values, then the priorities cease to have a function and the standard constant player count DVZ process ensues. The addition of the concept of playable note ranges in this spec somewhat complicates the way priorities function and requires additional logical steps to perform correct note allocations.

[0065] Figures 22 through 25 illustrate the sequence of note assignments to priorities if all notes fall in playing range of all paths. This is how DVZI without playing ranges works. Priorities for each path are listed in the boxes next to the path labels.

[0066] With DVZ priority and no playing ranges involved, if we were to play just one note, that note would be assigned to all paths whose priorities are set to one. If we play two notes, notes will be allocated to those paths with priorities set to one or two. The top note will go to the higher path(s), and the bottom note will go to the lower path(s) regardless of whether they're set to priority one or two. What happens now that we have playing ranges active for the paths, we play one note, and it is out of range of all paths with priority one? It goes to whatever path(s) has the lowest priority number and where it is not out of the path's range.

[0067] Fig. 26 shows the same priority setup as the previous several figures: five paths with four priorities. Here one note is sounding. Since there is only one note, it should go to the cello path, whose priority is set to one. But since the note is out of the playing range of the cello, it has to go somewhere else. The only paths that can play it are violins I, violins II, and violas. Violins I and II are both set to priority two and violas is set to priority three, so the note is assigned to violins I and II.

[0068] Fig. 27 illustrates a similar situation ; two notes are sounding. The higher of the two notes can be played by priorities one, three, and four. The lower of the two notes can be played by priorities one and four. If playing ranges were not taken into consideration, these notes would be assigned to priorities one and two. Since priority one is still in play, we know we *must* use priority one. There are two combinations of distributions that involve priority one: assigning the notes to priorities one and three or assigning the notes to priorities one and four. Since three is a lower priority than four, the method of solving for note allocation assigns the top note to priority three and the bottom note to priority four.

Priorities and octave transposition

[0069] Paths that have force octave transposition enabled will *always* play notes provided that notes can be transposed in the correct direction (we point out here that transpose up and transpose down may be enabled separately, and that either, neither or both may be enabled per path). This means that if notes can be played on paths with lower priorities with transposition, then notes will be transposed and may not sound where played on any path. For the next example we assume that force octave transposition up is set for the cello path, which is set to priority one. Even though the input note falls only within the playable range of the basses, it gets transposed into the cello range. This is illustrated in Fig. 28.

[0070] Here's another example, per Fig. 29; all paths have transpose up and transpose down set (activated). Two notes are played, both in the bass range, which is priority 4. Since both *can be transposed* such that they become within the playable range of priority one and priority two paths, this is done. No note is now sounding in its originally played octave.

[0071] Remember, if the DVZ is set to bottom weighting, the D and not the E will be doubled because the method solves the DVZ without taking any transpositions into account, and the D is the lower note at the input. Refer to Fig. 30.

[0072] What if violins I is set to force octave shift up but *not* violins II. Refer to Fig. 31. We have two notes. Neither input note is within range of a priority one or two path. Because Force Octave Shift UP is active for the violins I and the Celli, both notes are transposed up, but into the celli and the first violins, not into the second violins. Even though second violins and first violins have the same priority and would otherwise play a given note, second violins cannot because the note is out of their range.

[0073] What if we play two notes that both are below the Priority One and two playable ranges, and Force Octave Shift Up (transposition) is on for the celli, but not anything else, per Fig. 32. Since a note can be transposed to be played by the priority one celli, the method must do this. The next highest priority that can play a note (the basses) are set to priority four; since the method must play the second note at another priority and no others have transposition enabled, that note goes to the basses.

[0074] In the next example, per Fig. 33, let's assume force octave shift up is enabled for both the celli and the basses, but not anything else. We play two notes: one note is out of range of all instruments, and one note is in range of the basses. Since there are two notes playing, the method must try to use priority one, which it can. This leaves only priority four left as a within playable range path so the two notes will be assigned to priorities one and four. One of the notes can be played without transposition by priority four, the E. If the system were to do this, however, then the lower note (the D) would have to be transposed above the playable-as-input E. This is not allowed or the melodic intent would be violated; higher input notes have to remain higher, at least with respect to absolute (note letter) value if not octave value, so both notes are transposed.

[0075] Fig. 34 illustrates an example of an apparatus 3400 for executing the method according to an embodiment of the invention. The note assignment processor 3402 comprises a note input for receiving a notes- to- be- played signal from a note input source 3410. A central processing unit (CPU) 3404 performs steps according to embodiments of the invention, including detecting the signal, determining the number of notes to be played simultaneously, and performing iterative process to assign each note to a selected channel. A note register 3405 is provided for storing the total number of notes to be played simultaneously, while a note list register 3415 is provided for storing the notes to be played simultaneously in a pitch order. A current note register 3420 is used for storing the identity of the current note processed by the central processing unit 3404. A channel register 3425 is used for storing the total number of channels available for note assignment, while a channel list register 3430 is used for storing the channels in a specified order and a channel pitch range register 3435 is used for storing lowest and highest playable pitches per each channel.

[0076] When implementing transposition, a transpose up register 3440 is used for each channel to indicate upward note transposition for that channel and a transpose down register 3445 s used for each channel to indicate downward note transposition for that channel. A weighting preference register 3450 is used to store top or bottom weighted assignment of notes to channels when such is utilized. A channel priority register 3455 is used to store a value corresponding to the order in which notes played will be assigned to channels in which they are playable, when this feature is utilized.

[0077] While the particular embodiment discussed herein involves MIDI (Musical Instrument Digital Interface) note definitions, implementation of the methods using any other system that defines and controls musical note generation would equally fall within the envisioned scope of this system and method.

Claims

1. A note assignment processor (3402) for assigning notes to selected channels to be played by said channels, comprising:

   an input for receiving a notes- to- be- played signal;
   a central processing unit (3404) for detecting said signal, determining the number of notes to be played simultaneously, and performing iterative process to assign each note to a selected channel;
   a note register (3405) for storing the total number of notes to be played simultaneously;
   a note list register (3415) for storing the notes to be played simultaneously in a pitch order;
   a current note register (3420) for storing the identity of the current note processed by said central processing unit (3404);
   a channel register (3425) for storing the total number of channels available for note assignment;
   a channel list register (3430) for storing the channels in a specified order; and **characterized by**
   a channel pitch range register (3435) for storing lowest and highest playable pitches per each channel,
   a transpose up register (3440) for each channel to indicate upward transposition of a note for that channel, so as to bring the transposed note within the range on the channel; and
   a transpose down register (3445) for each channel to indicate downward transposition of a note for that channel, so as to bring the transposed note within the range on the channel.

2. The note assignment processor (3402) of claim 1, further comprising a weighting preference register (3450) to store either top weighted assignment of notes to channels for giving, in case an odd distributions of notes occurs, extra notes to the higher pitch instruments thereby giving preference to the higher sounding musical instrument, or bottom weighted assignment of notes to channels for giving said extra notes to the lower pitch instruments thereby giving preference to the lower sounding musical instrument.

3. The note assignment processor (3402) of claim 1 or 2, further comprising a channel priority register (3455) to store a value corresponding to the order in which notes played will be assigned to channels in which they are payable.

4. The note assignment processor (3402) of claim 3, further comprising operating the processor (3402) to designate actual notes to be played by one or more particular channels, which are determined to be capable of playing the notes, by filling out an array in accordance with a playable range specified in the channel pitch range register (3435), the transpose up register (3440), the transpose down register (3435) and the weighting preference register (3450) for each channel.

5. The note assignment processor (3402) of claim 1, wherein the total number of channels remains constant for the entire duration of a music piece played.

6. The note assignment processor (3402) of claim 1, wherein said processor (3402) performs an iterative process to determine to which available channels played notes may possibly be assigned according to the note pitch and each channel's allowable ranges, as designated by the channel's stored pitch range limits and as may be extended by virtue of any values

stored in the transpose up register (3440) and in the transpose down register (3445).

7. The note assignment processor (3402) of claim 1, wherein the total number of channels may vary from zero up to the number of available channels as more notes are played after which number of notes the number of channels does not increase with additional notes played and instead an overflow iterative process assigns remaining unassigned notes to an already assigned channel, to thereby assign at least one channel to play at least two notes.

8. The note assignment processor (3402) of claim 1, wherein the order in which notes are assigned to channels accords with the specified channel priority values.

9. A method for assigning notes to selected channels to be played by said channels, comprising:

designating a plurality of channels, each channel emulating an audio instrument; defining a note range for each channel; receiving instructions to play a plurality of defined notes simultaneously; and
allocating each of the plurality of notes to at least one of the channels according to note range assigned for each channel;
wherein when one channel has no note within its range, performing transposition of one of the note so as to bring the transposed note within the range on the channel.

10. A method of assigning a sequence of sounds, each sound comprising a plurality of notes, the method comprising:

assigning a plurality of channels;
assigning a number of voices to each channel to play the sequence of sounds;
forcing the number of voices in each channel to remain constant throughout the sequence,
regardless of the number of notes assigned to each channel throughout the sequence;
defining a notes range for each of the channels;
allocating each of the plurality of notes to at least one of the channels according to note range assigned for each channel..

11. The method of claim9 or 10, wherein voices within a given channel emulate a musical instrument of the same kind.

12. The method of claim 11, further comprising performing one of top- weighting or bottom weighting note allocation, wherein top-weighting comprises giving preference to higher sounding musical instrument and bottom-weighting comprises giving preference to lower sounding musical instruments.

13. The method of claim 11, further comprising assigning priority value to each of the channels, and assigning notes to channels according to the priority.

Patentansprüche

1. Notenzuordnungsprozessor (3402) zur Zuordnung von Noten zu ausgewählten Kanälen, die von den Kanälen gespielt werden sollen, umfassend:

eine Eingabe zum Empfang eines die zu spielenden Noten enthaltendes Signal;
eine zentrale Verarbeitungseinheit (3404), um das Signal zu detektieren, eine Anzahl von Noten, die gleichzeitig gespielt werden sollen, zu bestimmen und einen iterativen Prozess durchzuführen, um jede Note einem ausgewählten Kanal zuzuordnen;
ein Notenregister (3405) zum Speichern der Gesamtzahl der Noten, die gleichzeitig gespielt werden sollen;
ein Notenlistenregister (3415) zum Speichern der Noten, die gleichzeitig gespielt werden sollen, in einer Höhenlagen-Ordnung;
ein Aktuell-Notenregister (3420) zum Speichern der Identität der gegenwärtigen Note, die von der zentralen Verarbeitungseinheit (3404) verarbeitet wird;
ein Kanalregister (3425) zum Speichern der Gesamtzahl der Kanäle, die zur Notenzuordnung zur Verfügung stehen;
ein Kanallistenregister (3430) zum Speichern der Kanäle in einer vorgegebenen Reihenfolge; und **gekennzeichnet durch**
ein Kanalhöhenlagen-Bereichsregister (3435) zum Speichern der niedrigsten und höchsten spielbaren Höhenlage für jeden Kanal;
ein Aufwärts-Transpositionsregister (3440) für jeden Kanal, um eine Aufwärtstransposition einer Note für diesen Kanal anzuzeigen, um die transponierte Note in den Bereich auf dem Kanal zu bringen; und
ein Abwärts-Transpositionsregister (3445) für jeden Kanal, um eine Abwärtstransposition einer Note für diesen Kanal anzuzeigen, um die transponierte Note in den Bereich auf dem Kanal zu bringen.

2. Notenzuordnungsprozessor (3402) nach Anspruch 1, ferner umfassend ein Gewichtungs- Präferenzregister (3450), um eine am höchsten gewichtete Zuordnung von Noten zu Kanälen, um, wenn eine ungeradzahlige Verteilung von Noten auftritt, zusätzliche Noten den Instrumenten mit hoher Höhenlage

zu übergeben, wodurch eine Präferenz für das Musikinstrument mit höherer Höhenlage erteilt wird, oder eine am tiefsten gewichtete Zuordnung von Noten zu den Kanälen zu speichern, um zusätzliche Noten den Instrumenten mit niedrigerer Klanglage zu übergeben, wodurch eine Präferenz für das Musikinstrument mit niedrigerer Höhenlage erteilt wird.

3. Notenzuordnungsprozessor (3402) nach Anspruch 1 oder 2, ferner umfassend ein Kanalprioritätsregister (3455), um einen Wert zu speichern, der der Reihenfolge entspricht, in der die gespielten Noten den Kanälen zugeordnet werden, in denen sie gespielt werden können.

4. Notenzuordnungsprozessor (3402) nach Anspruch 3, ferner umfassend das Betreiben des Prozessors (3402), um aktuelle Noten zu bezeichnen, die von einem oder mehreren speziellen Kanälen gespielt werden sollen, von denen festgestellt wird, dass sie in der Lage sind, die Noten zu spielen, indem ein Feld entsprechend einem spielbaren Bereich, der in dem Kanal Höhenlage-Bereichsregister (3435), dem Aufwärts-Transpositionsregister (3440), dem Abwärts-Transpositionsregister (3435) und dem Gewichtungspräferenzregister (3450) für jeden Kanal spezifiziert ist.

5. Notenzuordnungsprozessor (3402) nach Anspruch 1, worin die Gesamtzahl der Kanäle während der gesamten Dauer des gespielten Musikstücks konstant bleibt.

6. Notenzuordnungsprozessor (3402) nach Anspruch 1, worin der Prozessor (3402) einen iterativen Prozess durchführt, um festzustellen, welchem der zur Verfügung stehenden Kanäle Noten zugeordnet werden können entsprechend der Notenhöhenlage und den zulässigen Bereichen von jedem der Kanäle, wie durch die dem Kanal zugeordneten, gespeicherten Höhenlagen-Bereichsgrenzen festgelegt ist und wie sie mit Hilfe von beliebigen Werten ausgedehnt werden können, die in dem Aufwärts-Transpositionsregister (3440) und dem Abwärts-Transpositionsregister (3445) gespeichert sind.

7. Notenzuordnungsprozessor (3402) nach Anspruch 1, worin die Gesamtzahl der Kanäle von Null bis zu einer Anzahl von zur Verfügung stehenden Kanälen variiert werden kann, wenn mehr Noten gespielt werden, wobei nach der Anzahl der Noten die Anzahl der Kanäle nicht erhöht wird, wenn zusätzliche Noten gespielt werden, und wobei stattdessen ein iterativer Überlauf-Prozess die restlichen nicht zugeordneten Noten einem bereits ausgewählten Kanal zuordnet werden, um dadurch wenigstens einen Kanal zum Spielen von wenigstens zwei Noten auszuwählen.

8. Notenzuordnungsprozessor (3402) nach Anspruch 1, worin die Reihenfolge, mit der die Noten den Kanälen zugeordnet werden, den spezifizierten Kanalprioritätswerten entspricht.

9. Verfahren zur Zuordnung von Noten zu ausgewählten Kanälen, die von den Kanälen gespielt werden sollen, umfassend:

Bezeichnen einer Vielzahl von Kanälen, wobei jeder Kanal ein hörbares Instrument emuliert;
Definieren eines Notenbereichs für jeden Kanal;
Empfangen von Befehlen, um eine Vielzahl von definierten Noten gleichzeitig zu spielen; und
Zuordnung von jeder der Vielzahl der Noten zu wenigstens einem der Kanäle entsprechend dem Notenbereich, der jedem Kanal zugeordnet ist;
wobei, wenn ein Kanal keine Note innerhalb seines Bereichs hat, eine Transposition von einer der Noten durchgeführt wird, um die transponierte Note in den Bereich auf dem Kanal zu bringen.

10. Verfahren zur Zuordnung von einer Sequenz von Schallereignissen, wobei jedes Schallereignis eine Vielzahl von Noten umfasst, wobei das Verfahren umfasst:

Auswählen einer Vielzahl von Kanälen;
Zuordnen einer Anzahl von Stimmen zu jedem Kanal, um die Sequenz der Schallereignisse zu spielen;
Zwingen der Anzahl der Stimmen in jedem Kanal, während der Sequenz konstant zu bleiben;
unabhängig von der Anzahl der Noten, die zu jedem Kanal während der gesamten Sequenz zugeordnet wurden, Definieren eines Notenbereichs für jeden der Kanäle;
Zuordnen von jeder der Vielzahl der Noten zu einem der Kanäle entsprechend einem Notenbereich, der jedem Kanal zugeordnet ist.

11. Verfahren nach Anspruch 9 oder 10, worin die Stimmen in einem vorgegebenen Kanal ein Musikinstrument derselben Art emuliert.

12. Verfahren nach Anspruch 11, ferner umfassend Durchführung einer am höchsten gewichteten oder am tiefsten gewichteten Notenzuordnung, worin die höchste Gewichtung darin besteht, dass die Präferenz einem höher höhenlagigen Musikinstrument gegeben wird, und dass die tiefste Gewichtung darin besteht, die Präferenz dem Musikinstrument mit der tieferen Höhenlage erteilt wird.

13. Verfahren nach Anspruch 11, ferner umfassend Zuordnung eines Prioritätswertes zu jedem der Kanäle und Zuordnung von Noten zu den Kanälen entsprechend der Priorität.

Revendications

1. Processeur d'attribution de notes (3402) pour attribuer des notes aux canaux sélectés pour être jouées par lesdits canaux, comprenant:

   - une entrée pour recevoir un signal de notes à jouer;
   - une unité de traitement centrale (3404) pour détecter ledit signal, déterminer le nombre de notes à jouer simultanément, et réaliser le procédé itératif pour attribuer chaque note à un canal sélecté;
   - un registre de notes (3405) pour stocker le nombre total de notes à jouer simultanément;
   - un registre avec la liste de notes (3415) pour stocker les notes à jouer simultanément dans un ordre de hauteur du son;
   - un registre de notes courantes (3420) pour stocker l'identité de la note courante traitée par ladite unité de traitement centrale (3404);
   - un registre de canaux (3425) pour stocker le nombre total de canaux disponibles pour l'attribution de notes;
   - un registre de listes de canaux (3430) pour stocker les canaux dans un ordre spécifié; et **caractérisé par**
   - un registre de gammes de hauteurs de son par canaux (3435) pour stocker les hauteurs de son à jouer les plus basses et les plus hautes par chaque canal,
   - un registre de transposition vers le haut (3440) pour chaque canal pour indiquer la transposition vers le haut d'une note pour ce canal, pour amener la note transposée dans la gamme sur le canal; et

   un registre de transposition vers le bas (3445) pour chaque canal pour indiquer la transposition vers le bas d'une note pour ce canal, pour amener la note transposée dans la gamme sur le canal.

2. Processeur d'attribution de notes (3402) selon la revendication 1, comprenant de plus un registre de préférences de pondération (3450) pour stocker soit l'attribution pondérée supérieure de notes aux canaux pour donner, en cas qu'une distribution impaire de notes survient, des notes extra aux instruments à hauteurs de son plus hautes, en donnant ainsi préférence à l'instrument musical de son plus haut, soit l'attribution pondérée inférieure de notes aux canaux pour donner lesdites notes extra aux instruments à hauteurs de son plus basses, donnant ainsi préférence à l'instrument musical de son plus bas.

3. Processeur d'attribution de notes (3402) selon la revendication 1 ou 2, comprenant de plus un registre de priorités de canaux (3455) pour stocker une valeur correspondant à l'ordre dans lequel les notes jouées seront attribuées aux canaux dans lesquels elles sont jouables.

4. Processeur d'attribution de notes (3402) selon la revendication 3, comprenant de plus faire fonctionner le processeur (3402) pour designer les notes réelles à jouer par un ou plusieurs canaux particuliers, qui sont déterminés pour être capables à jouer les notes, en remplissant un réseau en concordance avec une gamme jouable spécifiée dans le registre de gammes de hauteurs de son de canaux (3435), le registre de transposition vers le haut (3440), le registre de transposition vers le bas (3435) et le registre de préférences de pondération (3450) pour chaque canal.

5. Processeur d'attribution de notes (3402) selon la revendication 1, où le nombre total de canaux reste constant pour la durée totale d'une pièce musicale jouée.

6. Processeur d'attribution de notes (3402) selon la revendication 1, où ledit processeur (3402) réalise un procédé itératif pour déterminer auxquels canaux disponibles les notes jouées peuvent possiblement être attribuées selon la hauteur de la note et les gammes disponibles de chaque canal, telles que désignées par les limites des gammes de hauteurs de son stockées dans les canaux et comme elles peuvent être étendues en vertu de toutes les valeurs stockées dans le registre de transposition vers le haut (3440) et dans le registre de transposition vers le bas (3445).

7. Processeur d'attribution de notes (3402) selon la revendication 1, où le nombre total de canaux peuvent varier de zéro au nombre de canaux disponibles quand plusieurs notes sont jouées après quoi le nombre de notes le nombre de canaux ne s'augmente pas avec les notes additionnelles jouées et au lieu un procédé itératif excédentaire fait attribuer les notes non attribuées restantes à un canal déjà attribué, pour attribuer de cette manière au moins un canal pour jouer au moins deux notes.

8. Processeur d'attribution de notes (3402) selon la revendication 1, où l'ordre dans lequel les notes sont attribuées aux canaux est en accord avec les valeurs de priorité des canaux spécifiés.

9. Procédé pour attribuer des notes aux canaux sélectés pour être jouées par lesdits canaux, comprenant:

   - designer une pluralité de canaux, chaque canal émulant un instrument audio; définir une gamme

de notes pour chaque canal; recevoir les instructions à jouer une pluralité de notes définies simultanément; et
- attribuer chacune de la pluralité de notes à au moins l'un de ces canaux selon la gamme de notes attribuée pour chaque canal;

où quand un canal ne présente pas de note dans sa gamme, réaliser la transposition d'une de la note pour amener la note transposée dans la gamme sur le canal.

10. Procédé pour attribuer une séquence de son, chaque son comprenant une pluralité de notes, le procédé comprenant:

- fixer une pluralité de canaux;
- attribuer un nombre de voix à chaque canal pour jouer la séquence de son;
- imposer le nombre de voix en chaque canal pour rester constant tout au long de la séquence, quelle que soit le nombre de notes attribuées à chaque canal tout au long de la séquence;
- définir une gamme de notes pour chacun des canaux;
- attribuer chacune de la pluralité de notes à au moins un d'entre les canaux selon la gamme de notes attribuée pour chaque canal.

11. Procédé selon la revendication 9 ou 10, où les voix dans un canal donné émulent un instrument musical du même type.

12. Procédé selon la revendication 11, comprenant de plus réaliser une allocation pondérale de notes hautes ou basses, ou l'allocation pondérale haute comprend donner préférence à l'instrument musical de son plus haut et l'allocation pondérale basse comprend donner préférence à l'instrument musical de son plus bas.

13. Procédé selon la revendication 11, comprenant de plus attribuer la valeur de priorité à chacun des canaux, et attribuer les notes aux canaux selon la priorité.

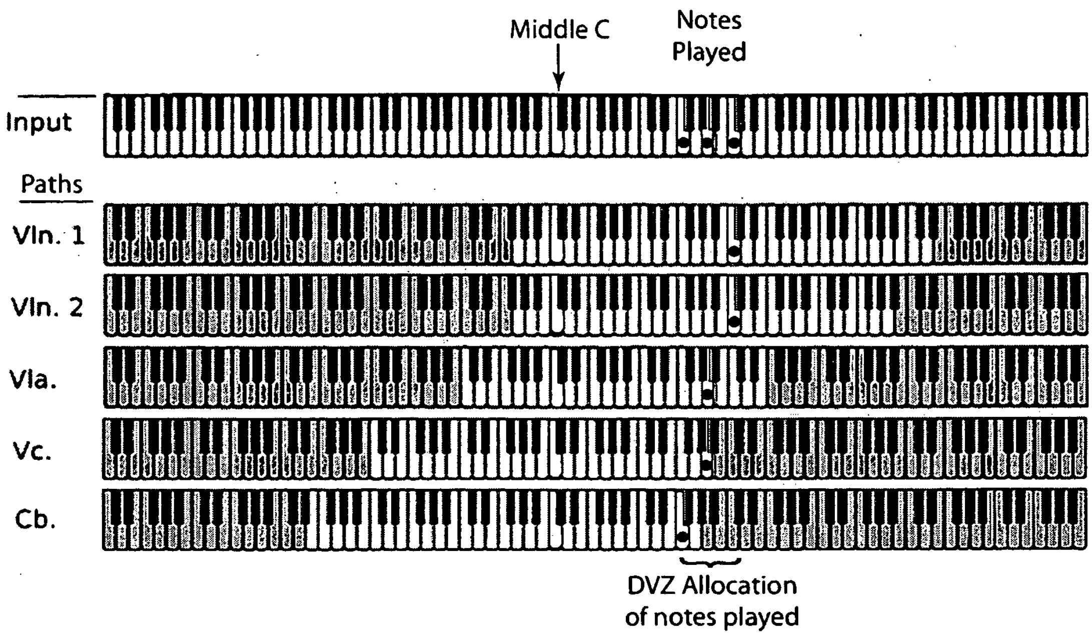

Fig. 1 of 33

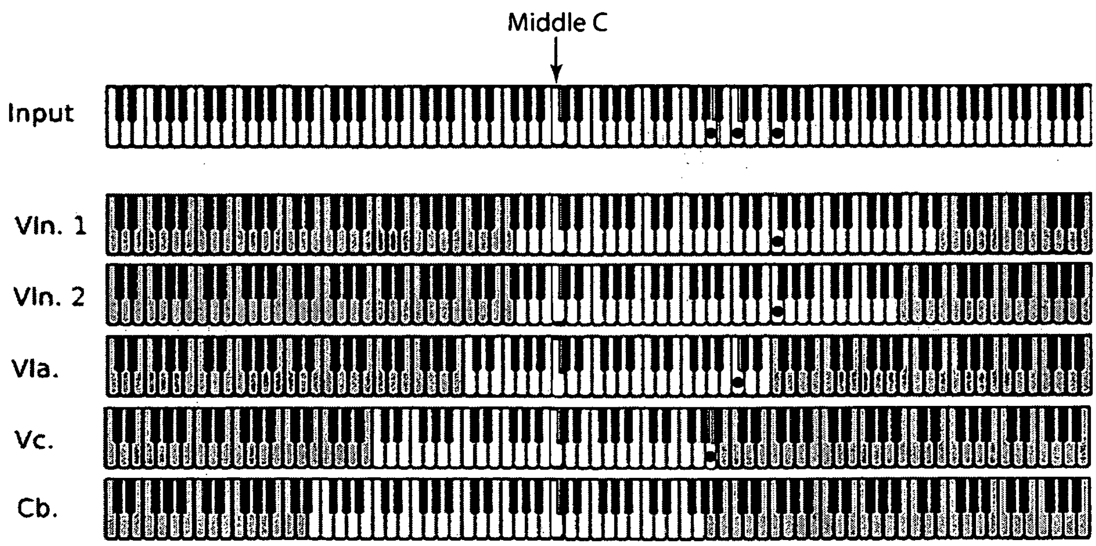

Fig. 2 of 33

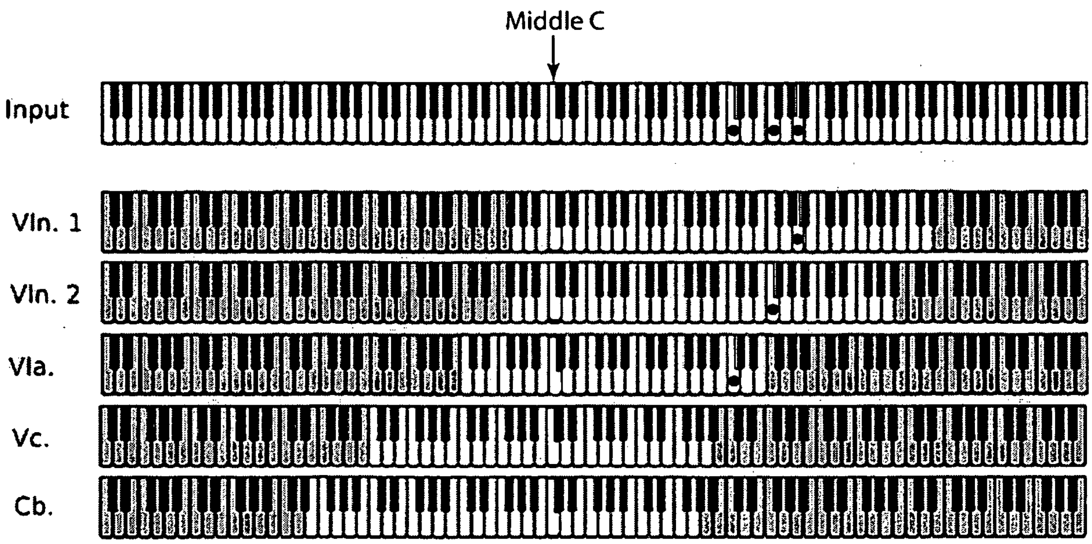

Fig. 3 of 33

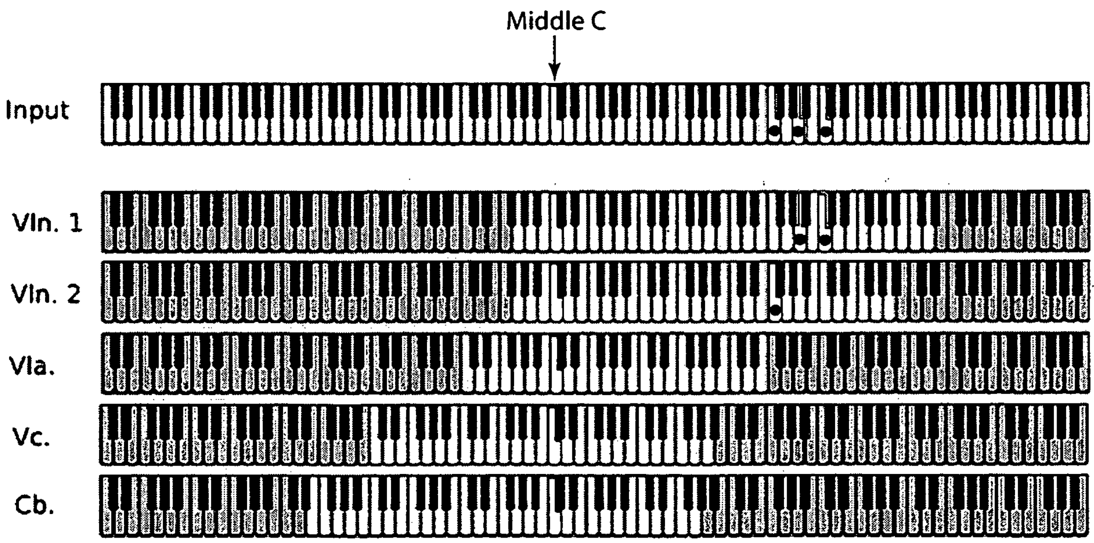

Fig. 4 of 33

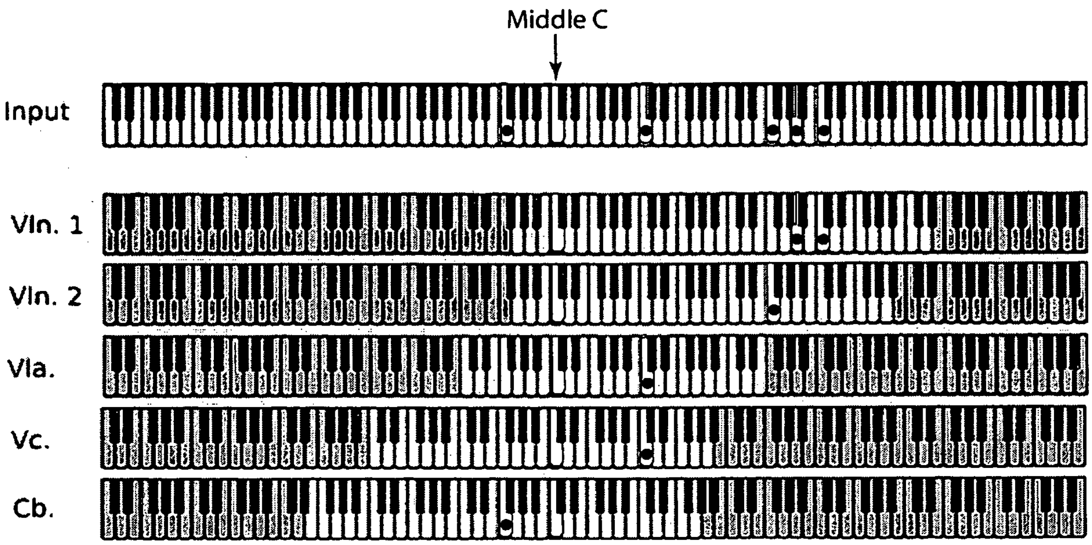

Fig. 5 of 33

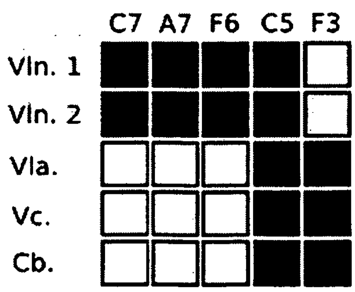

Fig. 6 of 33

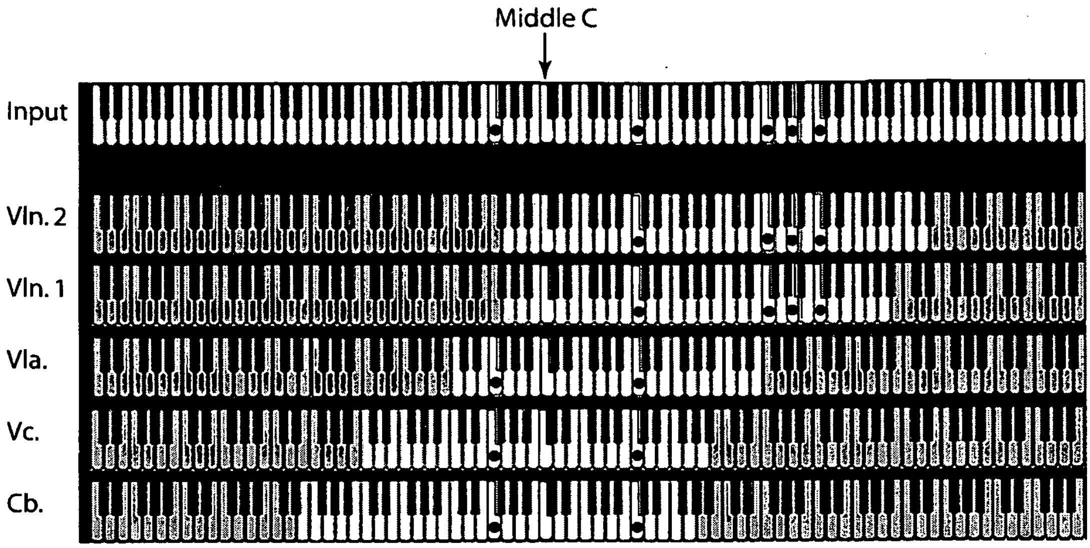

Fig. 7 of 33

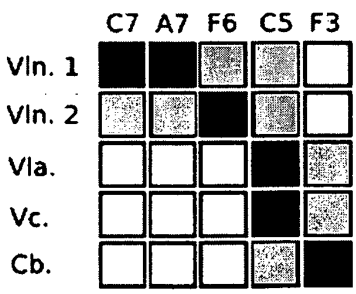

Fig. 8 of 33

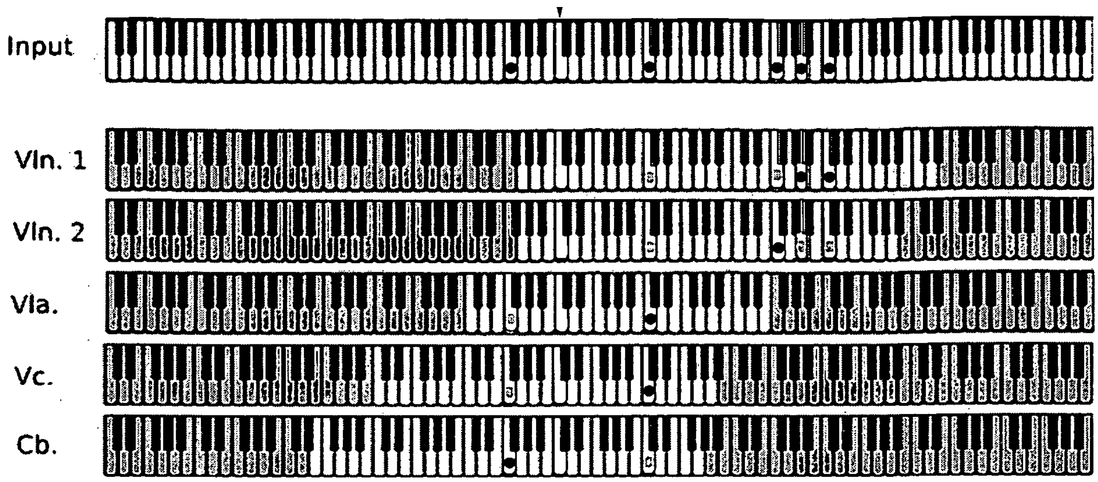

Fig. 9 of 33

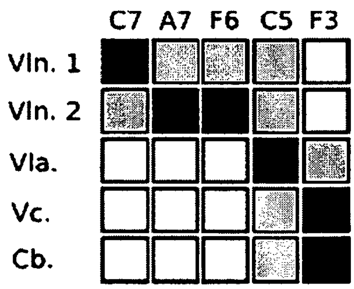

Fig. 10 of 33

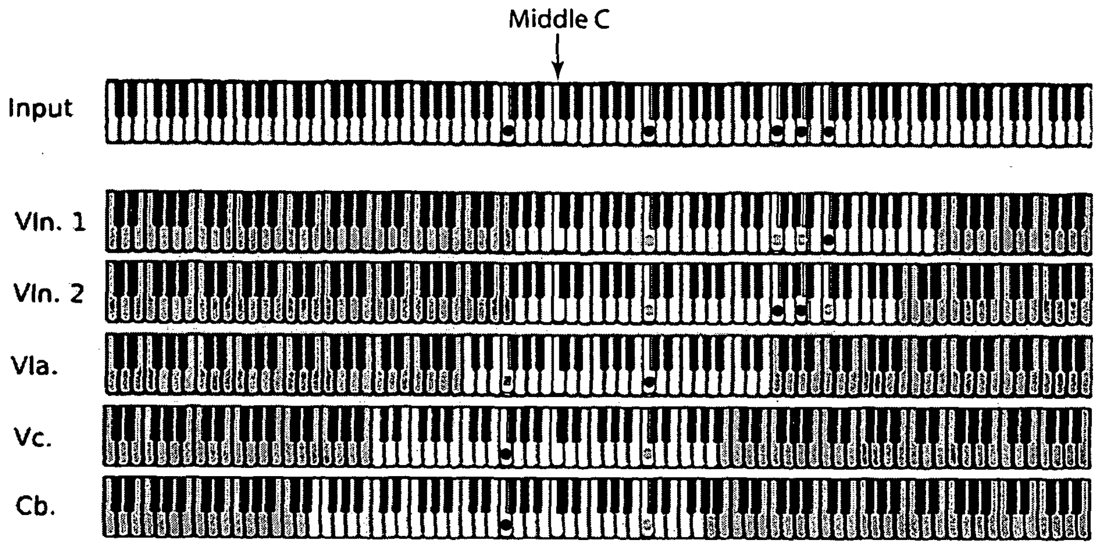

Fig. 11 of 33

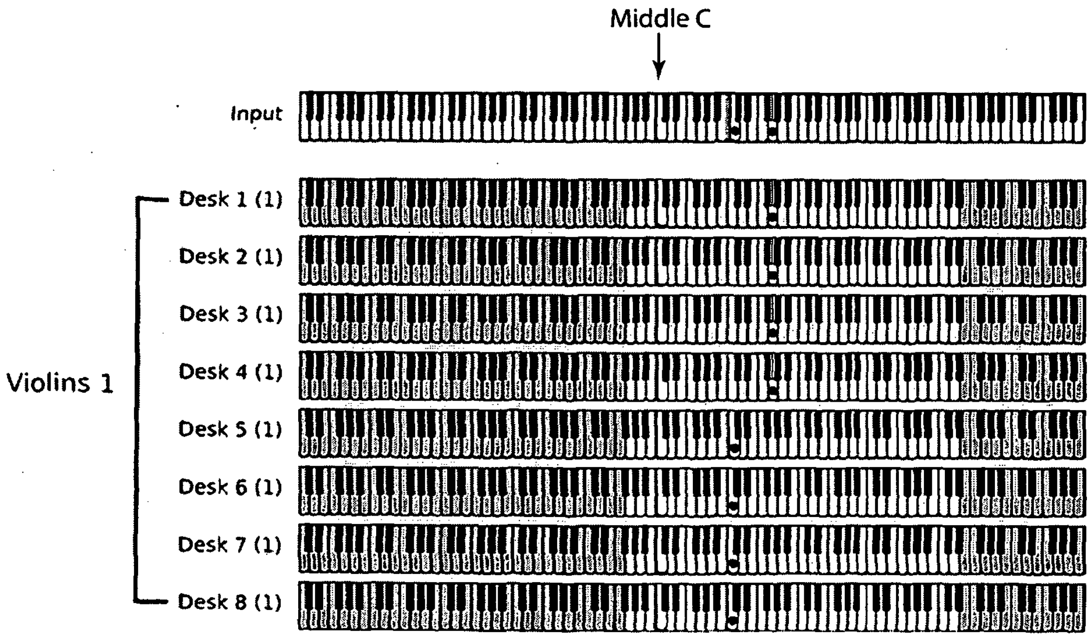

Fig. 12 of 33

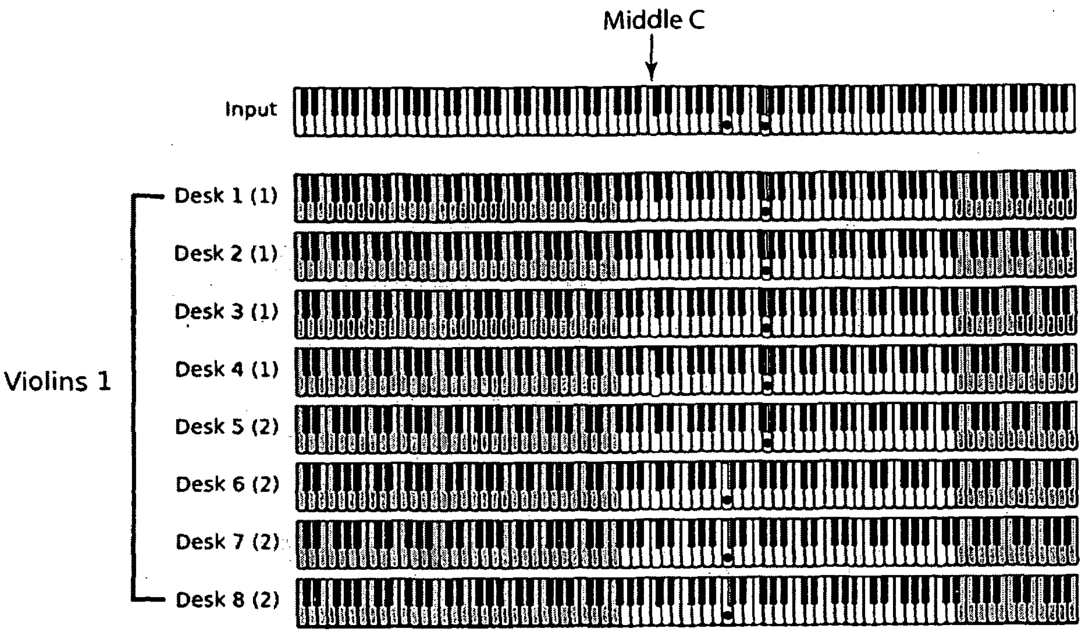

Fig. 13 of 33

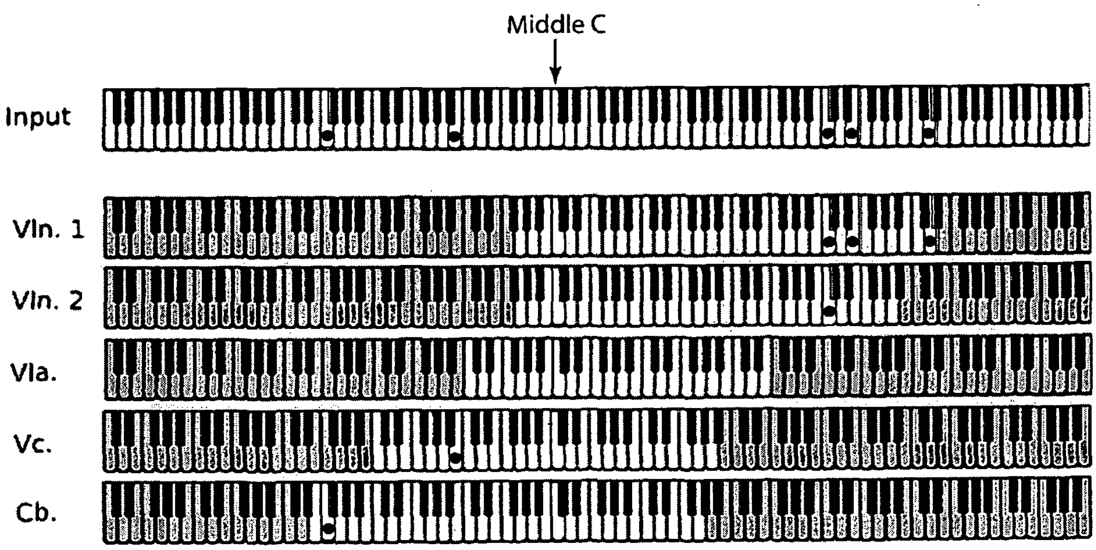

Fig. 14 of 33

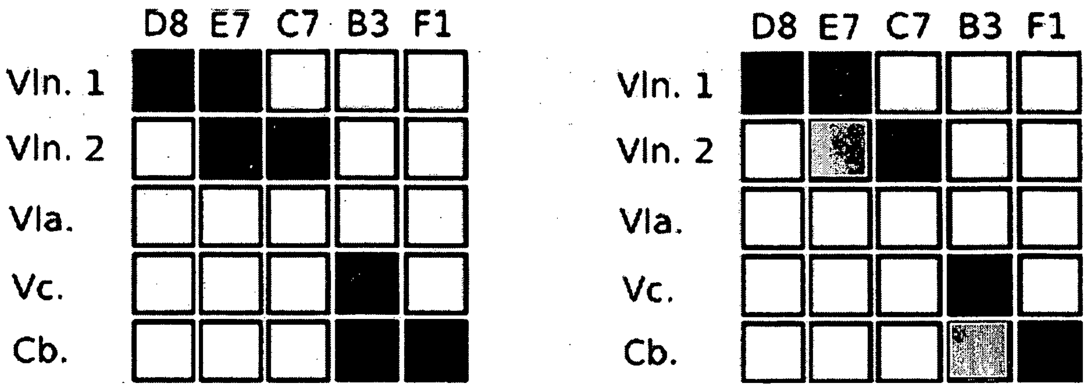

Fig. 15 of 33

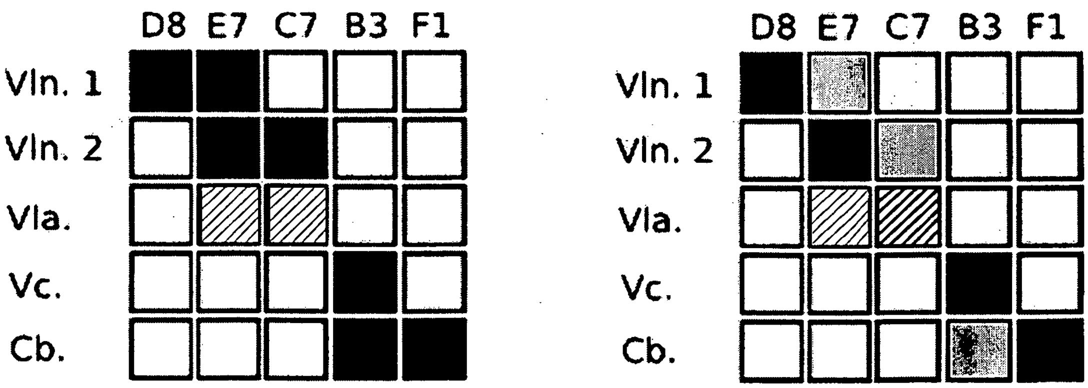

Fig. 16 of 33

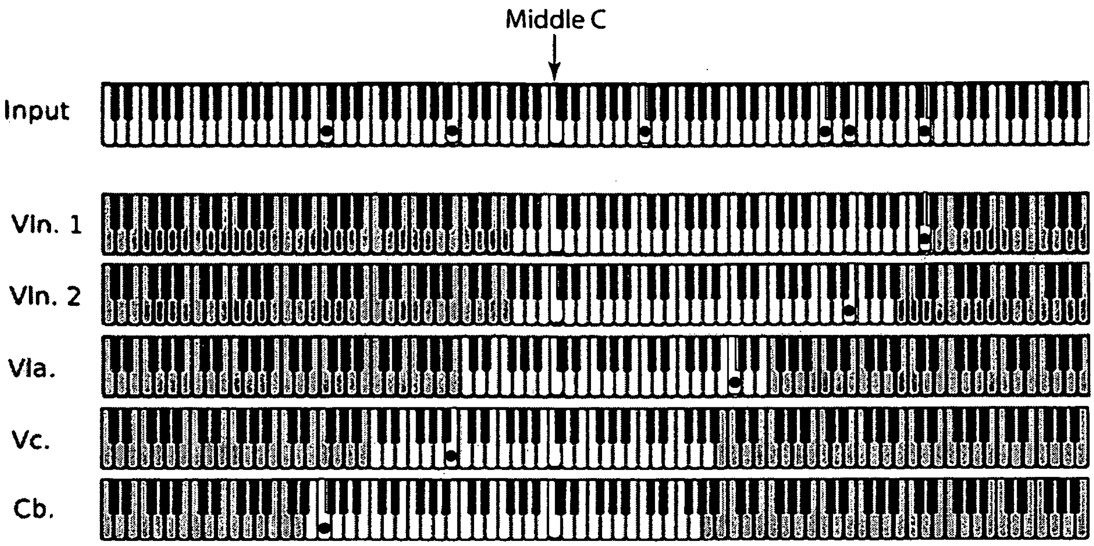

Fig. 17 of 33

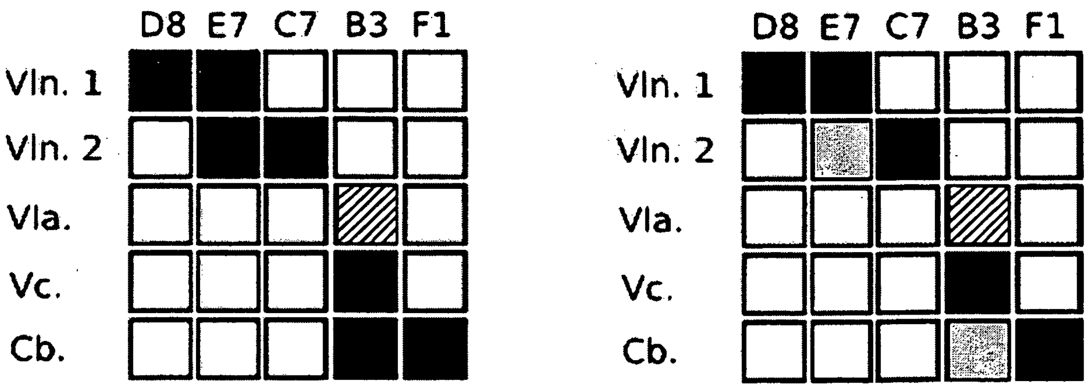

Fig. 18 of 33

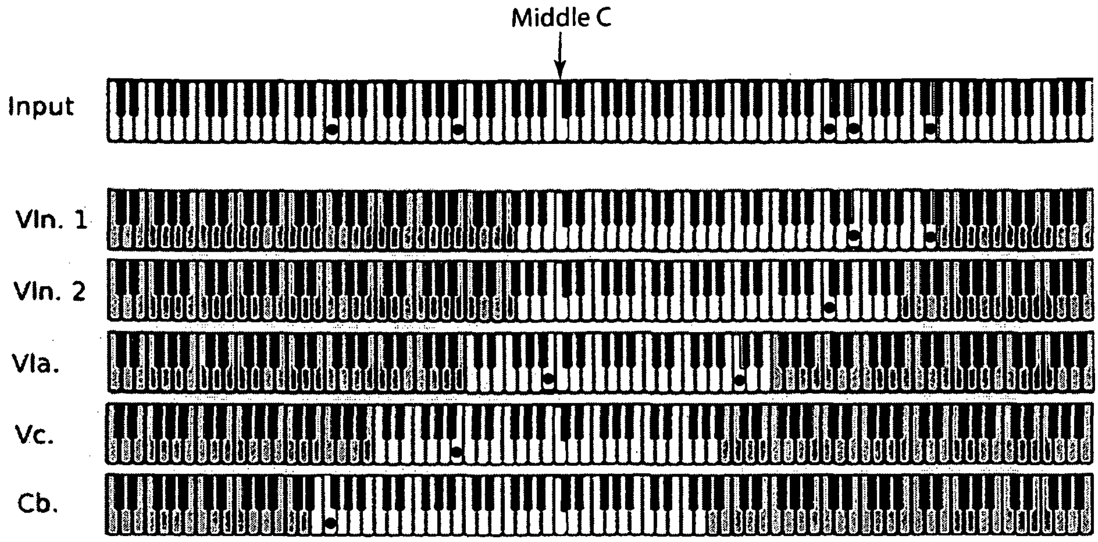

Fig. 19 of 33

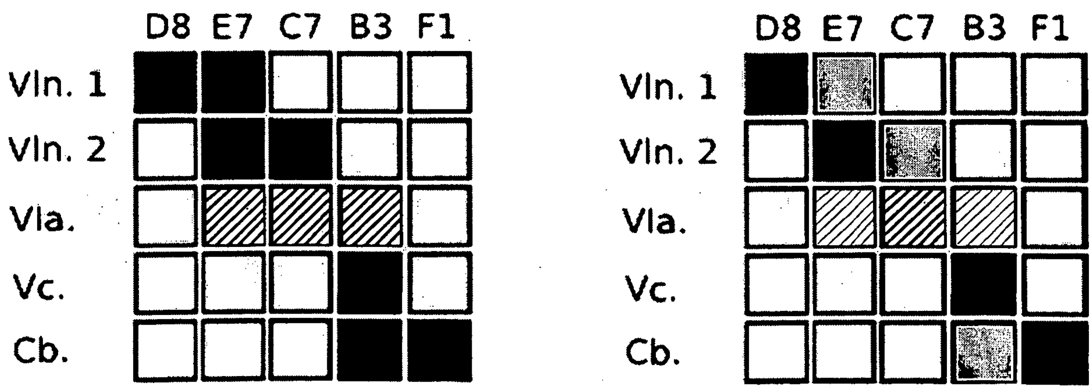

Fig. 20 of 33

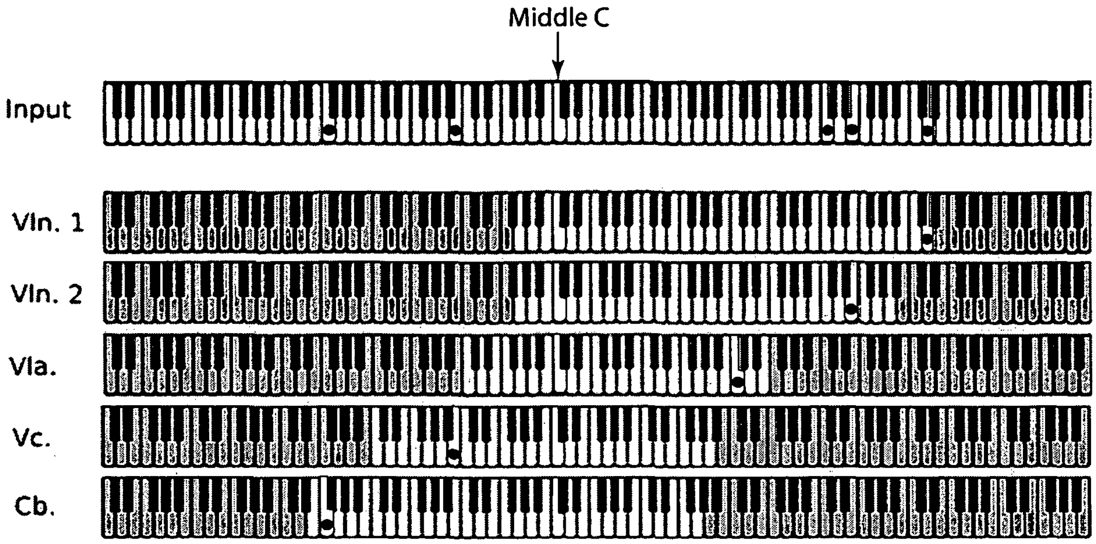

Fig. 21 of 33

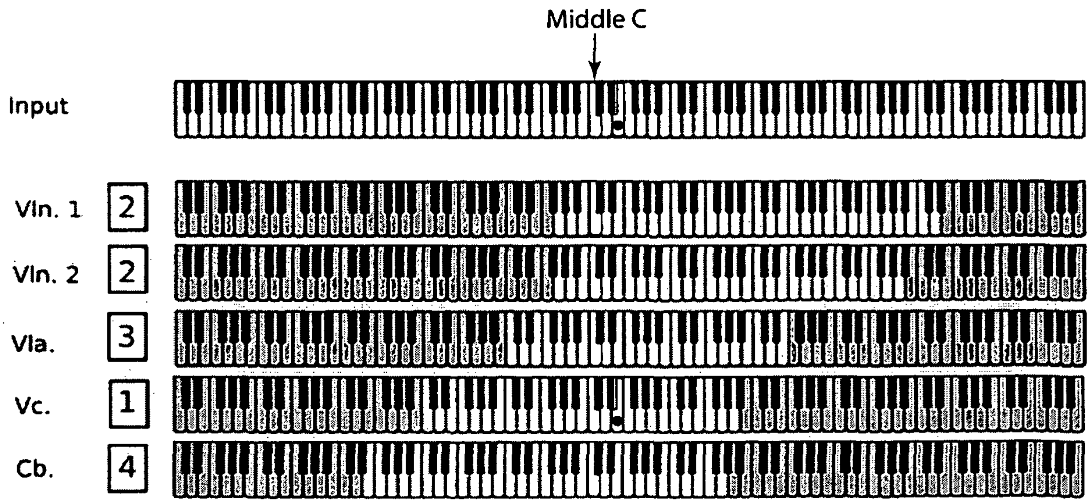

Fig. 22 of 33

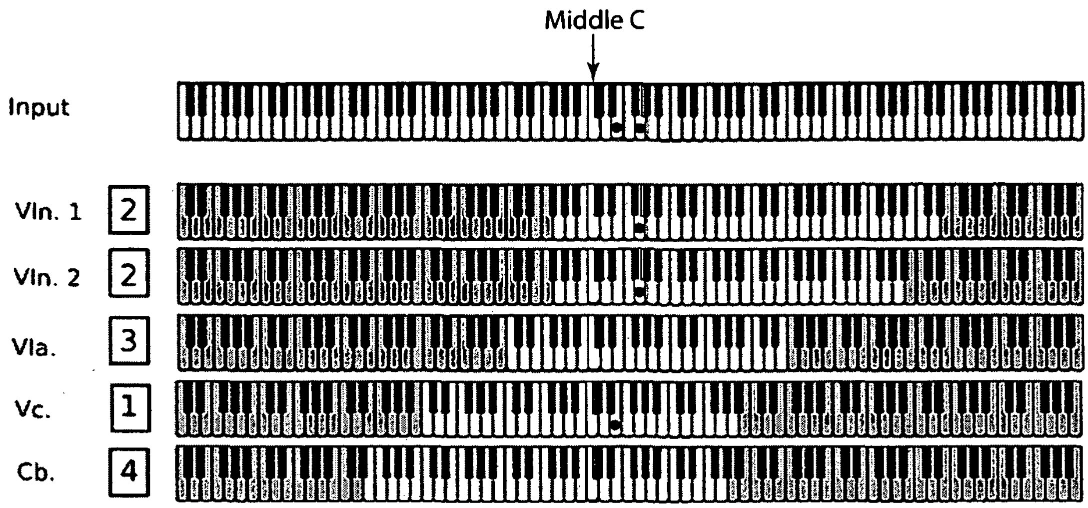

Fig. 23 of 33

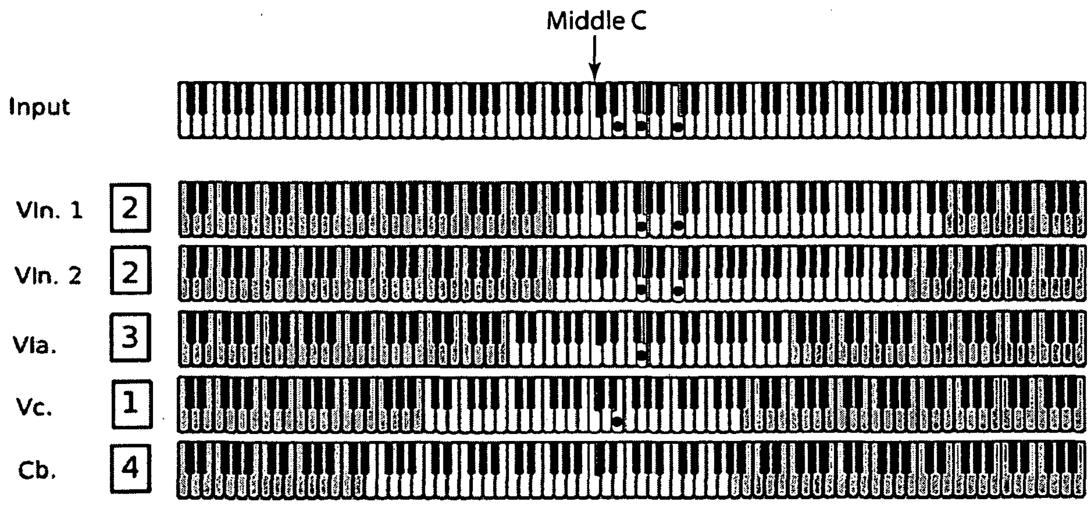

Fig. 24 of 33

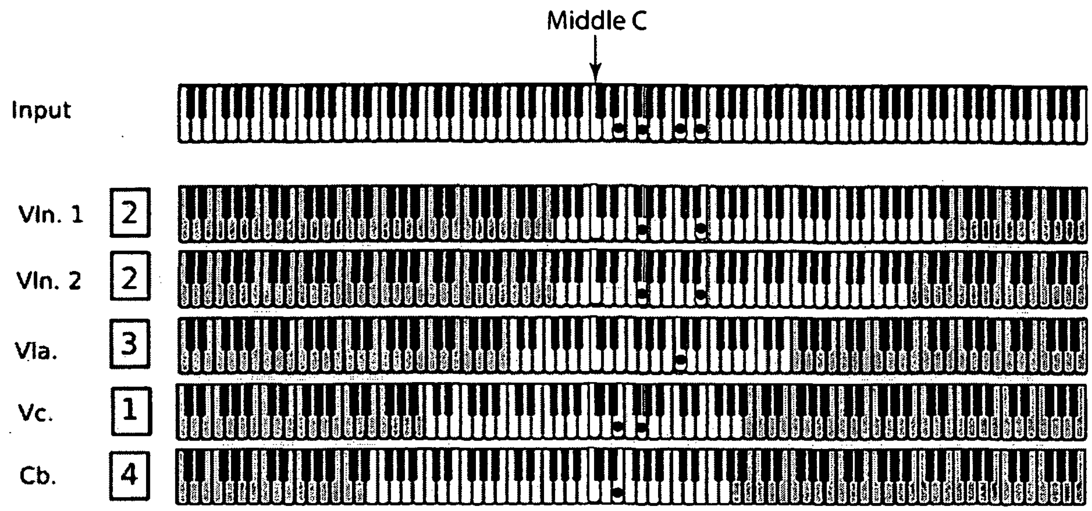

Fig. 25 of 33

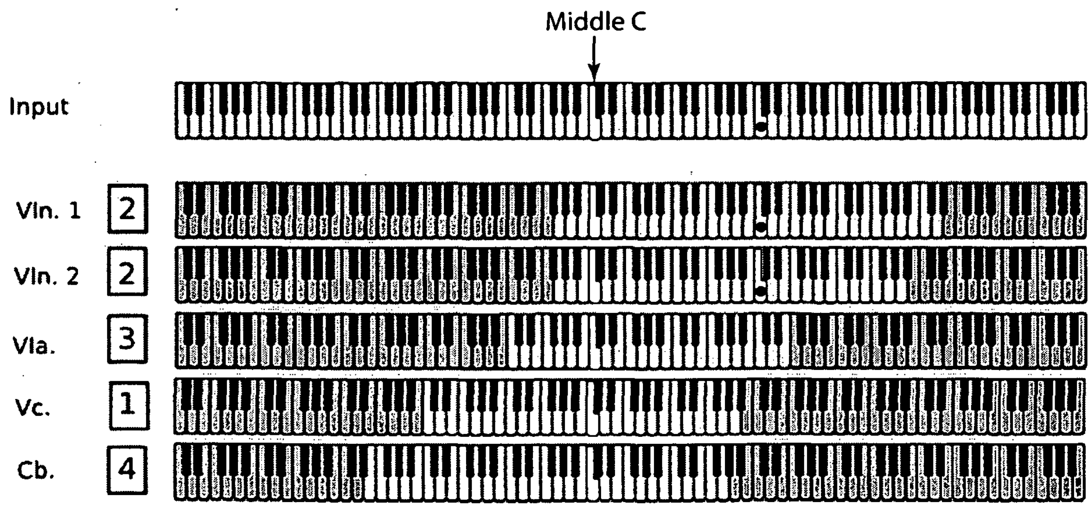

Fig. 26 of 33

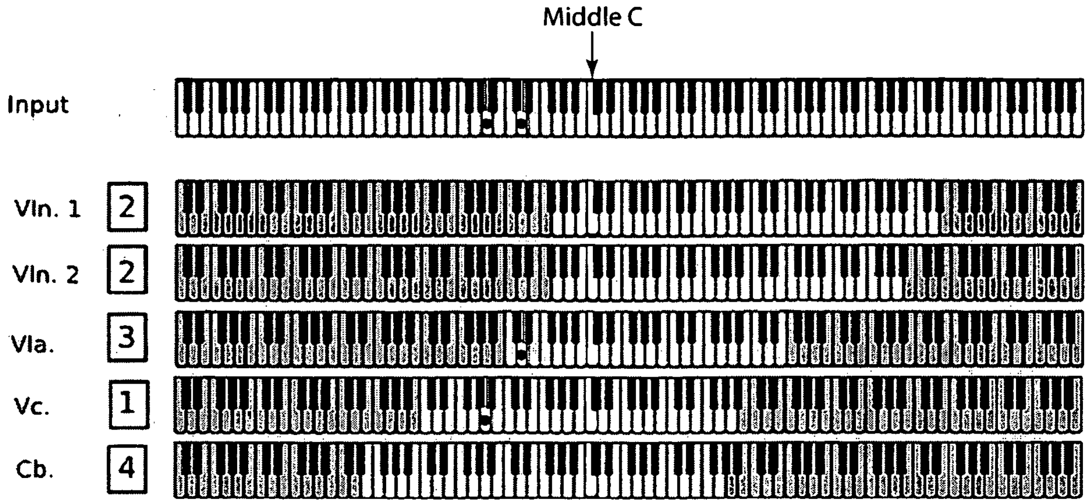

Fig. 27 of 33

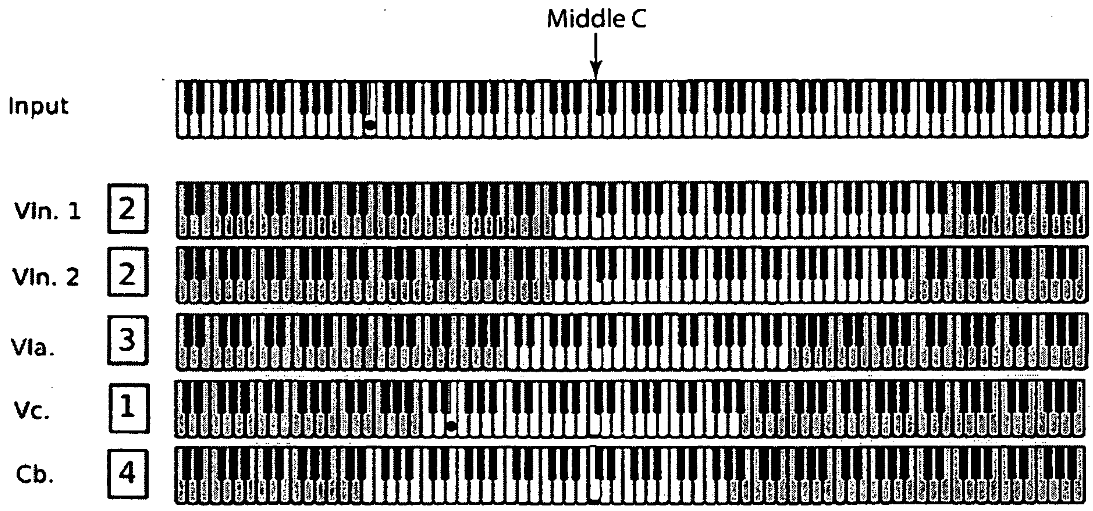

Fig. 28 of 33

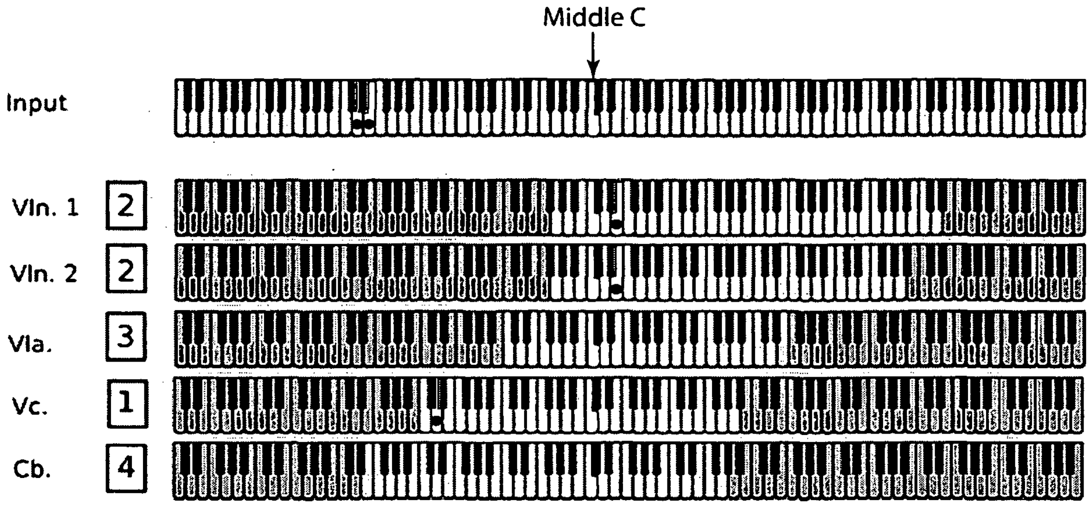

Fig. 29 of 33

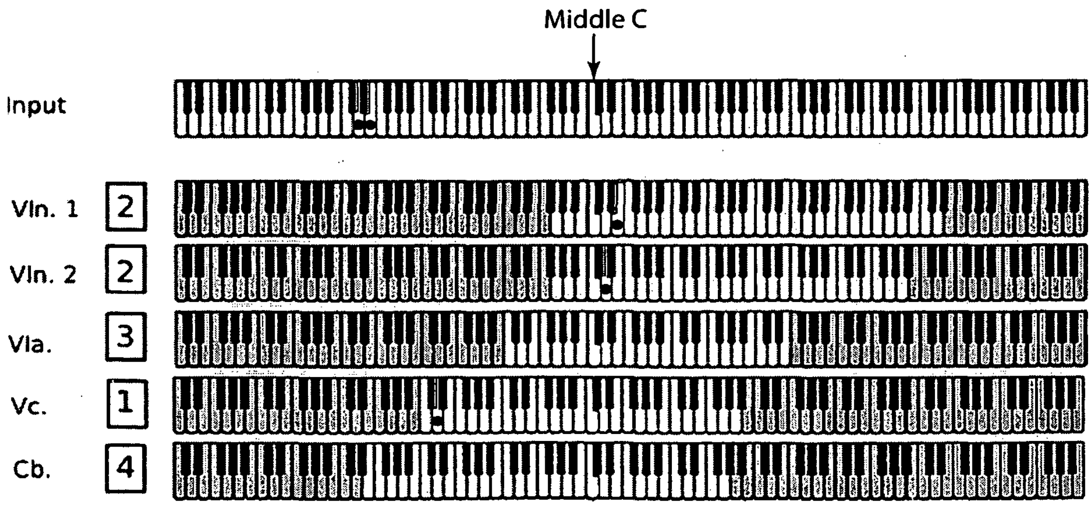

Fig. 30 of 33

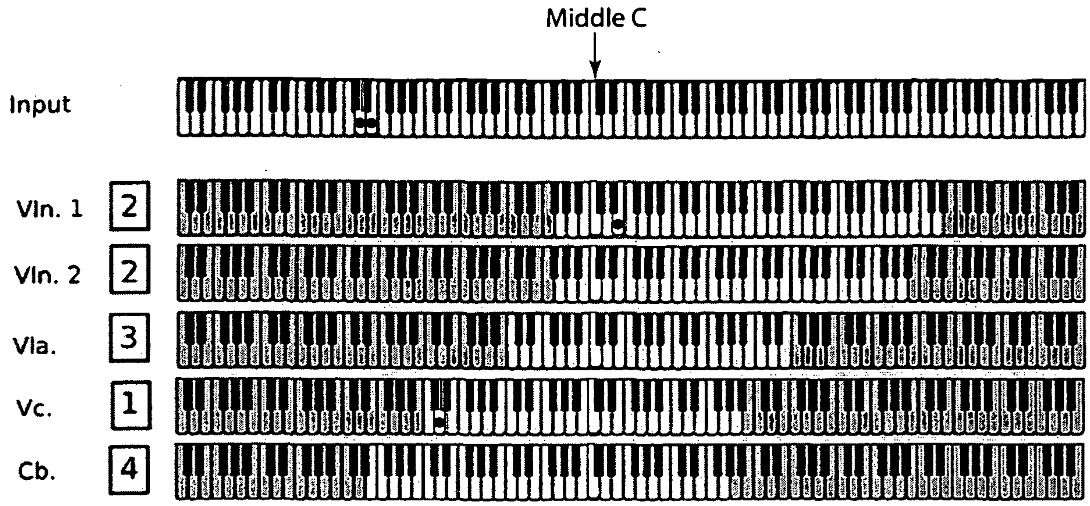

Fig. 31 of 33

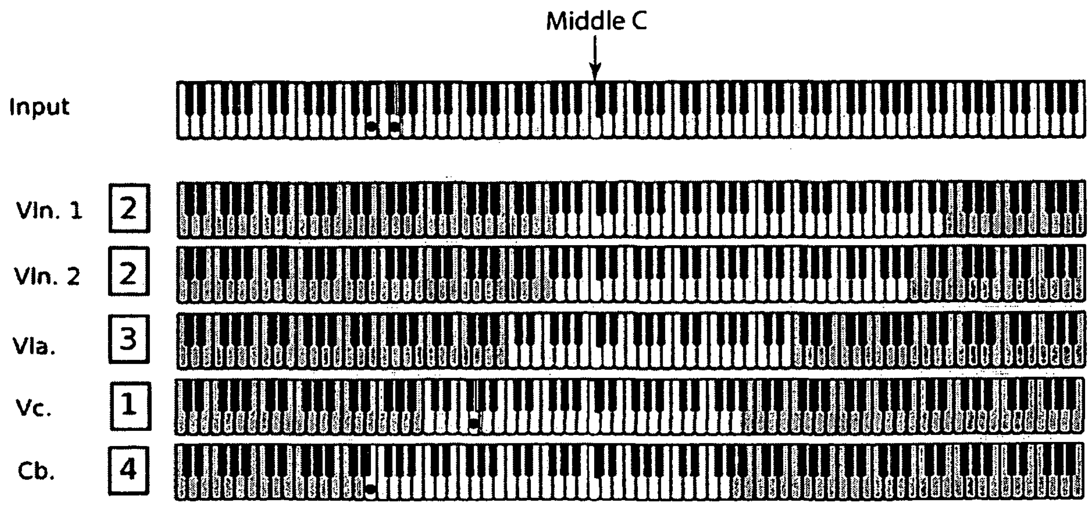

Fig. 32 of 33

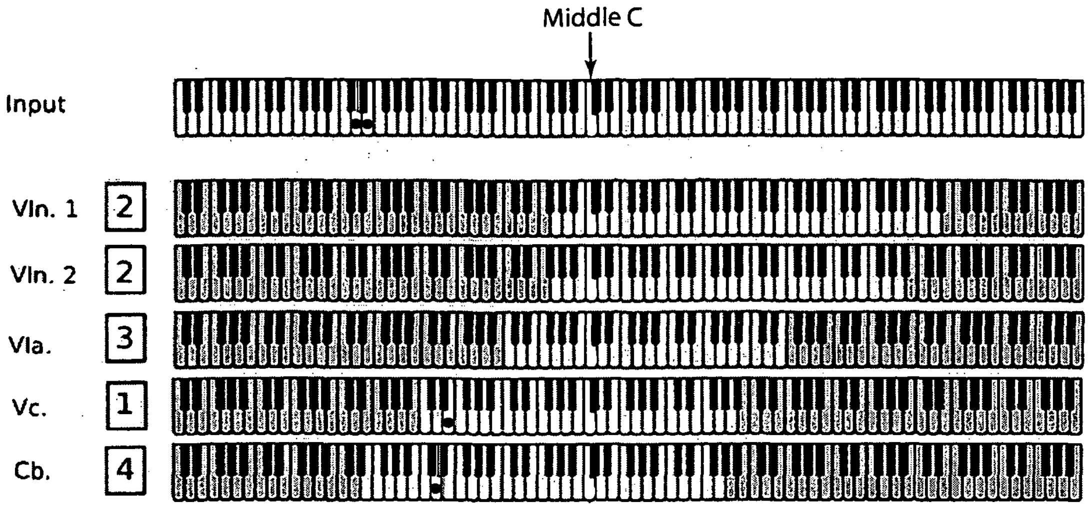

Fig. 33 of 33

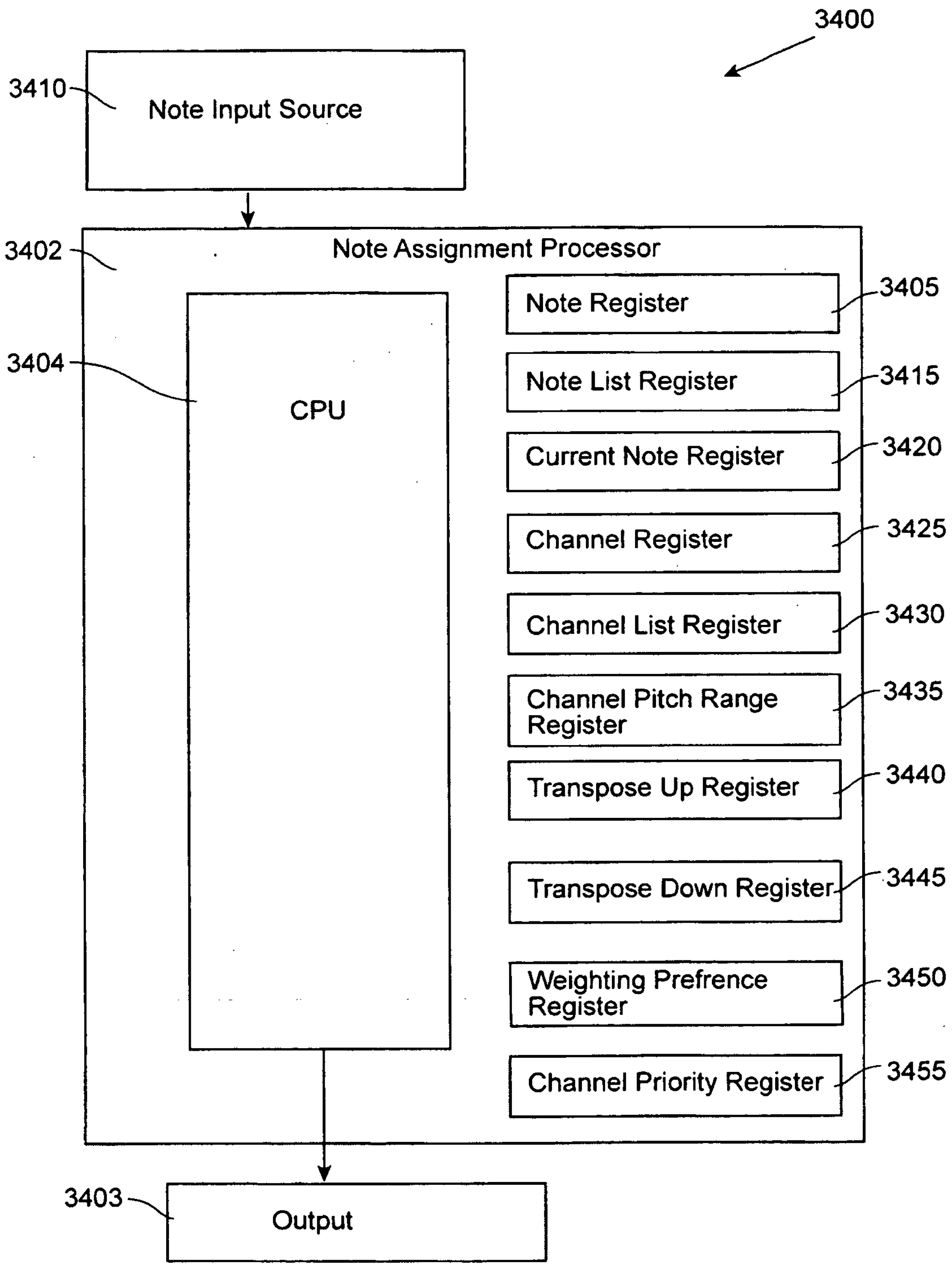
3400
3410
Note Input Source
3402
Note Assignment Processor
3404
CPU
Note Register
3405
Note List Register
3415
Current Note Register
3420
Channel Register
3425
Channel List Register
3430
Channel Pitch Range Register
3435
Transpose Up Register
3440
Transpose Down Register
3445
Weighting Prefrence Register
3450
Channel Priority Register
3455
3403
Output

Fig. 34

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 7109406 A **[0001] [0004]**
- US 0617757 W **[0001] [0004]**
- US 2006236848 A **[0008]**