The computation devices such as CPU 10 process the wave data with the waveform memory readout mode according to a simple algorithm in order to synthesize the actual waveform. In order to reproduce the wave data loaded in the RAM 13 or 20 by the sound source 22, the loaded basic wave data might be transferred again to the waveform memory 25 included in the sound source 22. In a conventional musical sound generator having a specific hardware sound source, the hardware sound source must have a temporary waveform memory to receive the loaded wave data. The CPU must execute the wave data transfer from the RAM to the waveform memory. However, in the present embodiment, the basic wave data is loaded to RAM 13 or 20 under the control of the CPU 10. Therefore, it is not necessary to provide the temporary memory to hold the wave data in the hardware. Further, it is not necessary to execute re-loading process in which the loaded wave data is transferred further to the external hardware device. The costs for the system hardware or software can be reduced, and the time from the end of the wave data loading to the reproduction of the sound can be shortened.

[0039] The sample value computing operation executed in step Sa15 (Figure 9) will be explained hereunder with reference to Figure 13. In step Sc1 of Figure 13, it is tested whether the co-processor 17 exists. In the present embodiment, the co-processor 17 is installed. However, this device is optional and may be lacked in some hardware setup. If the CPU 10 accommodates an arithmetic operation unit equivalent to the co-processor 17, the detection of the co-processor is not required. It is possible to regard the system as if the co-processor is installed, If the co-processor 17 is detected, the sample value calculation is done with the CPU 10 as well as the co-processor 17 in step Sc2. Otherwise, the sample value calculation is carried out only with the CPU 10 in step Sc3, if there is no co-processor 17. After step Sc2 or Sc3, the procedure returns. The sample value calculation is the same in steps Sc2 and Sc3, except that the co-processor 17 is used or not. Thus, the detail of the sample value calculation is explained with respect to step Sc3 here.

[0040] In step Sd1 of Figure 14, it is tested whether the current operating mode is set to the FM mode among the various CPU synthesizing modes. If the FM mode is set, a set of sample values corresponding to a number of channels are calculated at one sampling point according to the FM synthesizing method in step Sd2. For example, if polyphonic synthesizing is selected, a set of multiple sample values required for synthesizing a desired number of voices are calculated. In this case, the load of the CPU 10 is high, since various voices having various pitches may be concurrently created. Even worse, other processing such as graphics processing may be done in parallel. If the FM mode is not detected in step Sd1, it is detected whether the current operating mode is set to the waveform memory readout mode among the CPU synthesizing modes in step Sd3. If the waveform memory readout mode is detected here, a set of sample values required for one sampling point are read out from the waveform memory. In polyphonic synthesizing, multiple samples required for synthesizing a desired number of voices are read out. This data reading and data transfer is not carried out by the CPU 10, but by the DMAC 19. The wave data has been loaded to the area WAVE in the RAM 13 or 20, or provisionally stored in the ROM 11. If the waveform memory readout mode is not detected in step Sd3, it is detected whether the current operating mode is set to the harmonics synthesizing mode among the various CPU synthesizing modes in step Sd5. If the harmonics synthesizing mode is detected, a set of sample values required for one sampling process are calculated by the CPU 10 according to the harmonics synthesizing method in step Sd6. In polyphonic mode, a number of samples required for synthesizing a desired number of voices are calculated with the harmonics synthesizing method. In this case, the load of the CPU 10 is high, since various voices having various. pitches may be created concurrently. Even worse, other processings may be carried out in parallel, similarly to the situation under the FM mode. If the harmonics synthesizing method is not detected in step Sd5, it is tested whether the current operating mode is set to the physical model synthesizing mode among the various CPU synthesizing modes in step Sd7. If the physical model synthesizing mode is detected, samples required for one sampling point are calculated by the CPU 10 according to the physical model synthesizing method in step Sd8. In polyphonic mode, a number of samples required for synthesizing a desired number of voices are calculated with the physical model synthesizing method. If the physical model synthesizing mode is not detected in step Sd7, the current operating mode is out of the present embodiment, so that the warning for the wrong mode setting is executed in step Sd9 as well as other process. After steps Sd2, Sd4, Sd6, Sd8, or Sd9, the procedure returns, and then the following step Sa16 (Figure 9) is executed.

[0041] As for the waveform sample value calculation with the CPU 10 and the co-processor 17 executed in the step Sc2, the procedure shown in Figure 14 is executed with cooperation of the CPU 10 and the co-processor 17 so that the calculating operation can be made faster. The detailed explanation for the operation is omitted here, since the procedure is basically the same as the calculation only by the CPU.

[0042] Thus, in the waveform sample value calculation in step Sa15, the most influential processing on the quality of the synthesized sound is carried out by one of the CPU synthesizing modes. In the FM, harmonics synthesizing, or physical model synthesizing mode, the time required for the wave data calculation, as well as the number of voices to be reproduced simultaneously, is the bottle neck of the sound synthesizing. The wave data at each sample point is actually calculated in order