

Tone Waveform Generating Method and Apparatus Based on
Software

The present invention relates to a method of and apparatus for generating a tone waveform by a computer
5 executing a software program to calculate tone waveform sample values, to thereby allow a CPU or processor of a micro computer system to function as a tone generator.

Electronic musical instruments are conventionally known which implement a tone generator circuit by a tone generating
10 software program executed by a CPU. In order to reproduce a tone waveform with such known electronic musical instruments, it is necessary to supply tone waveform sample values to a digital-to-analog converter (DAC) regularly in response to clock pulses corresponding to a predetermined
15 sampling frequency. The conventionally-known electronic musical instruments are designed to generate tone waveform samples by performing tone generating calculation in response to generation of each clock pulse. But, calculating a tone waveform sample value in response to each such clock pulse
20 would undesirably result in inefficient operation of the CPU because preparing for and ending each tone waveform sample value calculation require a considerable overhead.

To provide a solution to the problem, it has been

9905000-7

04 OCT 1999

proposed to calculate a plurality of tone waveform sample values together for each relatively long, optional calculating clock pulse that is generated once every predetermined plurality of sampling clock pulses. However, 5 in cases where the tone generating software program is run concurrently with other software, such as business or game software, burdens on the CPU tend to be significantly great because tone waveform sample values for a plurality of samples have to be calculated in response to each calculating 10 clock pulse, thus resulting in operational delays of the other software.

It is therefore an object of the present invention to significantly improve the structure for buffering a plurality of tone waveform samples collectively generated through 15 arithmetic operations by a software tone generator, to thereby permit an efficient use of the buffer.

It is another object of the present invention to efficiently execute both a waveform generating process based on automatic performance information and a waveform 20 generating process based on real-time performance information, in a situation where a plurality of tone waveform samples are collectively generated through arithmetic operations by a software tone generator.

It is still another object of the present invention to 25 control image display and music performance together as a

unit, by incorporating an image control program and a music control program in single application software.

It is still another object of the present invention to provide a method of and apparatus for generating tone waveform samples by executing tone generating software, which can prevent or minimize operational delays of other software program executed concurrently with the tone generating software.

In order to accomplish the above-mentioned objects, the present invention provides a tone waveform synthesizer which comprises: a waveform generation section for collectively generating a plurality of tone waveform samples in advance of predetermined reproduction timing; a buffer section for temporarily storing the plurality of tone waveform samples generated by the waveform generation section, the buffer section being capable of writing and reading, independently of each other, in a parallel fashion; and a control section for controlling writing of the plurality of tone waveform samples into the buffer section and also performing control to read out the tone waveform samples from the buffer section at the reproduction timing. The control section controls the writing and reading into and from the buffer section independently of each other and also in such a manner that the advance of the writing does not outpace that of the reading.

In one preferred implementation, the buffer is capable

of writing and reading, independently of each other, in accordance with respective pointers or address signals. The write address advance is controlled so that the tone waveform samples are sequentially written into the buffer from the beginning of a predetermined storage region of the buffer and, upon arrival at the end of the storage region, the write address returns to the beginning of the storage region. Although indicated reading locations follow the advance of the writing operation, the advance of the writing operation is fast, and thus control is performed such that indicated wiring locations do not get ahead of the indicated reading locations advancing in accordance with a predetermined reproduction rate. This arrangement permits an efficient use of the storage region in the buffer. Namely, unless the advance of the writing operation outpaces the advance of the reading operation, unread (i.e., yet-to-be-read) tone waveform sample data can be written in a considerable proportion of the buffer's storage region, which thus achieves an efficient use of the storage region. By contrast, with the so-called "double buffer scheme" characterized by alternating the read/write mode between two buffer sections, it is only possible to write data into just a half of the entire buffer storage region, and hence an efficient use of the storage region is impossible.

The present invention also provides a method of generating a tone waveform based on performance information

using a processor executing a tone waveform forming process, which comprises: a step of receiving automatic performance information; a step of receiving real-time performance information generated in response to a real-time performance operation; a step of generating tone waveform samples using the processor, the step of generating including a step of collectively generating a plurality of tone waveform samples in advance of predetermined reproduction timing on the basis of the received automatic performance information and a step of collectively generating a plurality of tone waveform samples in advance of predetermined reproduction timing on the basis of the received real-time performance information, the step of generating being capable of generating the tone waveform samples based on the automatic performance information and the tone waveform samples based on the real-time performance information in a parallel fashion; and a step of outputting the tone waveform samples generated by the step of generating. This arrangement makes it possible to efficiently execute both a waveform generating process based on automatic performance information and a waveform generating process based on real-time performance information in a case where a plurality of tone waveform samples are collectively generated through arithmetic operations by a software tone generator.

25 The present invention also provides a method of generating a tone waveform using a processor capable of

executing a plurality of different programs on a time divisional basis, which comprises: a step of supplying the processor with application software including at least an image control program for controlling image display, a music control program for controlling tone generation and a general control program, the music control program including tone color data; and a step of causing the processor to execute, under control by the general control program, the image control program and the music control program in a parallel fashion, to output image data generated as a result of execution of the image control program and tone waveform data generated as a result of execution of the music control program. By thus incorporating the image control program and music control program in the single application software, the image control program and music control program can be executed in parallel and in correlation to each other and can be controlled together as a unit.

The present invention also provides a method of generating a tone waveform based on automatic performance information, which includes executing a waveform calculating process for calculating tone waveform samples by sharing an arithmetic processing section with another process, and which is characterized by comprising the steps of detecting a portion of processing capability of the arithmetic processing section that is not currently occupied by the other process as available processing capability for the waveform

calculating process, calculating a plurality of tone waveform samples based on the performance information in advance of predetermined generation timing thereof by executing the waveform calculating process using the available processing capability detected by the step of detecting, this step of calculating including a step of storing the calculated tone waveform samples in a memory, and generating a tone waveform by reading out the tone waveform samples from the memory.

The present invention is characterized in that a portion of processing capability of the arithmetic processing section that is not currently occupied by the other process is detected as available processing capability for the waveform calculating process and in that the waveform calculating process is executed using the thus-detected available processing capability so as to calculate tone waveform samples corresponding to the detected available processing capability. The calculated tone waveform samples are stored in a memory and then output from the memory at a predetermined sampling frequency. The calculation of the tone waveform samples is always executed in advance of predetermined generation timing when these samples are to be output. According to the present invention, tone waveform samples are calculated only to the amount corresponding to the detected available processing capability. In cases where a plurality of application programs are concurrently run on a multitask basis, this arrangement of the invention permits

extra tone waveform samples to be calculated and saved when sufficient processing capability of the arithmetic processing section is available for the waveform sample calculation, thus providing increased operational efficiency of the arithmetic processing section. Also, even when the arithmetic processing section is busy with another application program, i.e., even when the arithmetic processing section is used for the other application program with priority over the waveform sample calculation, tone waveform samples can continue to be output without a break by just reading out the tone waveform samples calculated and saved previously when sufficient processing capability of the arithmetic processing section was available for the waveform sample calculation, with the result that the other application program can be executed with increased efficiency.

For better understanding of various features of the present invention, the preferred embodiments of the invention will be described in detail hereinbelow with reference to the accompanying drawings, in which:

Fig. 1 is a block diagram illustrating an exemplary hardware structure of a microcomputer system which has a tone generating function based on the principle of the present invention;

Figs. 2A to 2B are diagrams illustrating exemplary

configurations of various software programs stored in the microcomputer system of Fig. 1;

Fig. 3 is a timing chart explanatory of reproduction processing carried out in the microcomputer system;

5 Fig. 4 is a flowchart illustrating exemplary operation of the microcomputer system;

Fig. 5 is a flowchart illustrating a part of a detailed control flow of the reproduction processing; and

10 Fig. 6 is a flowchart illustrating the remaining part of the control flow of the reproduction processing.

Fig. 1 is a block diagram illustrating an exemplary hardware structure of a microcomputer system 21 which has a tone generating function based on the principle of the present invention. The microcomputer system 21 comprises a
15 CPU 10 as an arithmetic processing section, to which are connected, via a bus 22, a ROM 11, a RAM 12, a hard disk device 13, a timer 14, a serial interface 15, a keyboard 16, a display 17 and a reproduction section 18. The ROM 11 has prestored therein a basic operating program that is essential
20 to the operation of the microcomputer system 21. The RAM 12 buffers data and a program to be executed, as well as various data occurring during execution of a program by the CPU 10. The hard disk device 13 has prestored therein a multimedia application software set as shown in Fig. 2, etc. and the
25 multimedia application software set is read into the RAM 12

prior to its execution. At predetermined regular intervals, the timer 14 outputs an interrupt signal to the CPU 10 and sends sampling clock pulses to the reproduction section 18. The serial interface 15 is provided for transmitting and
5 receiving data and control signals to and from various external peripherals connected to the microcomputer system 21.

The keyboard 16 and display 17 may both be of any suitable type depending on a particular application of the
10 microcomputer system 21. Where the system 21 is applied for example as an electronic musical instrument, the keyboard 16 may be the one conventional with an electronic piano keyboard or the like, and the display 17 may be a conventional LCD (Liquid Crystal Display). Where the system 21 is applied as
15 a karaoke device, the keyboard 16 may be one specialized to permit a selection from among pre-recorded music pieces and desired tempo control, and the display 17 may be a large-size CRT monitor.

The reproduction section 18 is a so-call sound board,
20 which stores in its internal buffer waveform data for a plurality of samples received from the CPU 10 and outputs the waveform data to the D/A converter 19 in response to each sampling clock generated by the timer 14. The internal buffer of the reproduction section 18 is of a type where data
25 can be additionally written, so that each group (i.e., block) of waveform data received from the CPU 10 is stored in the

buffer immediately after another waveform data group being currently read out from the buffer. The buffer may be constructed as a FIFO, dual-port or other suitable memory structure. The D/A converter 19 converts each waveform data
5 supplied from the reproduction section 18 into an analog tone signal, which is then audibly reproduced through a sound system 20 comprised of amplifiers and speakers.

In the above-mentioned hard disk 13, there may be stored various other data such as automatic performance data and
10 chord progression data and the above-mentioned operating program. By prestoring the operating program in the hard disk 13 rather than in the ROM 11 and loading the operating program into the RAM 12, the CPU 10 can operate in exactly the same way as where the operating program is stored in the
15 ROM 11. This greatly facilitates version-up of the operation program, addition of an operating program, etc. A CD-ROM (compact disk) 23 may be used as a removably-attachable external recording medium for recording various data such as automatic performance data, chord progression data and tone
20 waveform data and an optional operating program similarly to the above-mentioned. Such an operating program and data stored in the CD-ROM 23 can be read out by a CD-ROM drive 24 to be transferred for storage into the hard disk 13. This facilitates installation and version-up of the operating
25 program. The removably-attachable external recording medium may of course be other than the CD-ROM, such as a floppy disk

and magneto optical disk (MO).

A communication interface 25 may be connected to the bus 22 so that the microcomputer system 21 can be connected via the interface 25 to a communication network 26 such as a LAN (Local Area Network), internet and telephone line network and can also be connected to an appropriate sever computer 27 via the communication network 26. Thus, where the operating program and various data are not contained in the hard disk 13, these operating program and data can be received from the server computer 27 and downloaded into the hard disk 13. In such a case, the microcomputer system 21, a "client", sends a command requesting the server computer 27 to download the operating program and various data by way of the communication interface 25 and communication network 26. In response to the command, the server computer 27 delivers the requested operating program and data to the microcomputer system 21 via the communication network 26. The microcomputer system 21 completes the necessary downloading by receiving the operating program and data via the communication network 25 and storing these into the hard disk 13.

It should be understood here that the microcomputer system 21 may be implemented by installing the operating program and various data corresponding to the present invention in any commercially available personal computer. In such a case, the operating program and various data

corresponding to the present invention may be provided to users in a recorded form on a recording medium, such as a CD-ROM or floppy disk, which is readable by the personal computer. Where the personal computer is connected to a communication network such as a LAN, the operating program and various data may be supplied to the personal computer via the communication network similarly to the above-mentioned.

Fig. 2 shows an exemplary configuration of the multimedia-type application software set stored in the RAM 12 of Fig. 1. As shown in Fig. 2A, this multimedia-type application software set, which may for example be for karaoke performance or for game playing, comprises a header, overall control software, image control software, music control software and other control software. The header contains various data indicative of a version, memory size, etc. of the multimedia-type application software set. The overall control software controls the concurrent or parallel operation of the above-mentioned software in such a manner that all these software operates smoothly, and the image control software controls various images to be presented on the display 17. The music control software controls automatic performance based on automatic musical performance data and generation of a tone waveform based on performance data supplied on a real-time basis. The other control software controls input by a user and other operations.

Fig. 2B shows a detail of the music control software,

which includes reproduction processing software, a tone color data section, an automatic performance data section and a working area. The reproduction processing software performs various operations, as flowcharted in Figs. 4, 5 and 6, to simulate the function of a hardware tone generator. The tone color data section contains various data to actually drive the reproduction processing software, including parameters for calculating waveform sample values and controlling envelopes, filter controlling parameters, etc. Where the reproduction processing software is designed to simulate a PCM (Pulse Code Modulation) tone generator, the tone color data section contains waveform data themselves. The automatic performance data section contains sequence data for automatically performing background and karaoke music. The working area provides various registers to store data that become necessary as the reproduction processing software runs as well as data produced during various processes, and the working area corresponds to internal registers of a conventional hardware tone generator and registers of various peripherals (e.g., of an interface) connected thereto.

Fig. 2C shows a detail of the working area, Fig. 2D shows a detail of a sample buffer contained in the working area, and Fig. 3 is a timing chart explanatory of tone waveform generating operations of the microcomputer system 21. As shown, the sample buffer is a ring buffer comprising "n" blocks BLK(0) to BLK(n - 1), and according to the present

invention, waveform data for 128 samples (128 waveform sample values) are written in each of the blocks.

In Fig. 3, the reproduction processing software is activated by a calculation triggering clock pulse BC generated every 128 sampling clock pulses. The reproduction processing software sends stored waveform data of each of the blocks of the sample buffer to the reproduction section 18, and also, while checking for the operational availability of the CPU 10, writes waveform data into the block so far as the data writing does not adversely influence other operations. Real-time performance data entered by the user or player via the keyboard 16 are input in response to a first calculation triggering clock pulse BC generated after the entry, so that corresponding waveform sample values are calculated. The reproduction section 18 buffers the waveform data that are input via the CPU 10 in response to the calculation triggering clock pulse BC, and the buffered waveform data are then read out, one sample for each sampling clock, and supplied to the D/A converter 19.

As shown in Fig. 2D, each of the blocks in the sample buffer in which waveform data are to be written is pointed to by a writing block pointer WP, while each of the blocks in the sample buffer from which waveform data are to be read out is pointed to by a reading block pointer RP. In Fig. 2C, reference character WF represents a write-enable flag and reference character RF represents a read-enable flag. The

read-enable flag RF is kept set throughout a period from the start to end of an automatic performance, and the write-enable flag WF is kept set throughout a period from writing of first data to writing of last data of an automatic performance. Because the data writing takes place in advance of the data reading in accordance with the present invention, the data writing and reading periods, i.e., the periods during which the write-enable flag WF and the read-enable flag RF are set, do not conform to each other.

10 In instructing data and instructing flag sections of the working area are written various instructions given from the overall control software, and these sections correspond to an instruction register within a conventional hardware tone generator. Specifically, data written in the instructing data and instructing flag sections are for example:

data designating a specific music piece number for which automatic performance data are to be reproduced, because the music control software contains automatic performance data for a plurality of music pieces;

20 data designating a range to be reproduced in a case where only part of performance data of a specific music piece are to be reproduced;

data designating a tone color for each performance part;
and

25 real-time performance data input via the keyboard 16 and serial I/O 15, which may be suitably received by the above-

mentioned other control software.

Part control data section of the working area contains tone color selecting data, tone volume level controlling data, etc. for each performance part in a case where
5 multipart automatic performance data are to be reproduced.

Channel control data section contains control data on a per-channel basis since the reproduction processing software is designed to deal with simultaneous sounding of tones in a plurality of tone generating channels.
10 Specifically, the channel control data include various data necessary for each of the channels to generate a tone signal, such as data designating a musical scale, a current address counter value, data determining a shape and current value of an envelope. Namely, this channel control data section
15 corresponds to registers within a conventional hardware tone generator.

Exemplary operation of the microcomputer system 21 will be described in greater detail hereinbelow with reference to flowcharts of Figs. 4, 5 and 6. Fig. 4 is a flowchart
20 illustrating an example of a main routine of the reproduction processing software used in the microcomputer system 21, in which reproduction processing is executed in a repetitive manner at step S2 after initialization of step S1.

Figs. 5 and 6 illustrate a detailed control flow of the reproduction processing. First, at step S10, the
25 reproduction processing waits for a calculation triggering

clock pulse BC to be generated. Assume that during this wait period, another process is being executed with the control returned to the overall control software. Upon generation of the calculation triggering clock pulse BC, the reproduction processing proceeds from step S10 to step S11, where a determination is made as to whether the read-enable flag RF is currently set to a value of "1" or not. If answered in the affirmative at step S11, this means that an automatic performance is currently in progress, so that the reproduction processing goes to step S12 to calculate waveform data for the automatic performance. At step S12, a comparison is made between current values of the writing block pointer WP and the reading block pointer RP. If the current value of the writing block pointer WP equals that of the reading block pointer RP ($WP = RP$), this means that waveform data of a block to be read out (to be sent to the reproduction section 18) this time have not yet been calculated, and thus arithmetic operations are performed at steps S13 to S17 to calculate waveform data of that block.

At step S13, the writing block pointer WP is incremented or advanced by one; however, because the sample buffer is a ring buffer as mentioned earlier, it is reset to "0" upon reaching its maximum value (this is true with incrementing operations of the writing and reading block pointers WP and RP as will be described later). Automatic performance data corresponding to one block are reproduced at next step S14.

Then, at step S15, it is determined whether there is sufficient time for calculating waveform data corresponding to the automatic performance data. If answered in the affirmative at step S15, calculation of waveform sample data is effected at step S16 with relatively high precision (for example, 48 kHz in calculating frequency and 32 bits in data size). If, however, there is no sufficient time for such high-precision waveform data calculation as determined at step S15, the calculation is effected at step S17 with the calculating precision lowered by a degree corresponding to the shortage of the calculating time. The lowering of the calculating precision may be effected by lengthening the sampling clock period and/or reducing the number of bits to be calculated at a time. After step S6 or S17, the reproduction processing proceeds to step S18.

If the value of the writing block pointer WP is greater than that of the reading block pointer RP as determined at step S12, this means that waveform data of the block to be read out this time have already calculated, and thus the reproduction processing goes from step S12 to step S18 directly.

At step S18, a determination is made as to whether the automatic performance has come to an end, by ascertaining whether the block to be read out this time is a last block (i.e., a block containing an end point). If the block to be read out this time is the last block, generation and readout

of waveform data based on the automatic performance data are no longer necessary, so that the write-enable flag WF and read-enable flag RF are both reset to "0" at step S19. After the resetting of the write-enable and read-enable flags WF
5 and RF, the reproduction processing proceeds to step S20 for calculation of a waveform corresponding to a real-time performance and transmission of the waveform data to the reproduction section 18.

When an automatic performance is not under way and hence
10 the read-enable flag RF is at a value of "0", the reproduction processing goes directly from step S11 to step S20 in order to increment the reading block pointer RP. Waveform data of the block specified by the reading block pointer RP (BLK(RP)) are transmitted to the reproduction
15 section 18 at step S23 as will be later described. A tone waveform is calculated in response to the player's real-time performance data input at step S21, and the calculated waveform data are added into the block specified by the reading block pointer RP at step S22. The waveform data are
20 then supplied from the block to the reproduction section 18 at step S23. After this, the block is cleared at step S24 now that the data of the block BLK(RP) are no longer necessary due to the data supply to the reproduction section 18. Namely, "0" is written into every location of that
25 block.

After step S24, the reproduction processing performs

operations at and after step S30 to effect data writing in advance of data reading. At step S30, a determination is made as to whether the write-enable flag WF is currently set to "1" ($WF = 1$). If answered in the affirmative at step S30, 5 this means that there is any other automatic performance data to be written, so that data writing operations are performed at and after steps S31. If, on the other hand, the write-enable flag WF is not currently set to "1" ($WF = 0$), this means that predetermined automatic performance have been 10 calculated to the end, so that the reproduction processing is brought to an end without performing any other operations.

At step S31, a determination is made as to whether there is time available for the "advanced" data writing, i.e., whether the CPU 10 is not busy with any other software 15 concurrently run with the reproduction processing software. If there is time available for the data writing, the writing block pointer WP is incremented by one at step S32 so as to point to a block in which data are to be written after this, and performance data corresponding to one block are read out 20 for reproduction at step S33. Then, waveform data for one block are calculated on the basis of the read-out performance data at step S34, and the calculated waveform data are written into a block pointed to by the writing block pointer WP. In this case, the waveform data calculation is executed 25 with high precision for all the tone generating channels because sufficient time can be spent on the calculation.

Thereafter, the reproduction processing goes to step S35 to make a determination as to whether the block for which the data writing has been executed this time is the last block, or to step S36 to make a determination as to whether the sample buffer is now full of unread waveform data. If the block is the last block as determined at step S35, the write-enable flag WF is reset to "0" at step S37 because it is no longer necessary to write waveform data, and then the reproduction processing is brought to an end. The determination as to whether the sample buffer is now full of unread waveform data is made by checking whether the writing block pointer WP has catch up with the reading block pointer RP after making a round through the ring buffer in advance of the reading block pointer RP ($WP = RP - 1$). Writing new data when the sample buffer is full of unread data will result in overwriting unread data, and thus the reproduction processing is brought to an end after the affirmative determination at step S36. If the sample buffer is not full of unread data, i.e., has any other block available for writing data, as determined at step S36, the reproduction processing loops from step S36 back to step S31. If it is determined at step S31 that there is still time left, then the data writing is performed for a next block.

In the above-described reproduction processing, the reading block pointer RP is constructed as a so-called "free-running" counter because the sample buffer may be used for

real-time performance data input when an automatic performance is not executed and the pointer RP is incremented whenever the reproduction processing steps through steps S20 to S24. Therefore, when waveform data are to be written into
5 the sample buffer prior to a start of an automatic performance (i.e., before the read-enable flag RF is set to "1"), it is determined how earlier than the start of an automatic performance the writing of waveform data should be initiated (namely, a determination is made of a specific
10 number of counts of the calculation triggering clock pulses BC by which the writing of waveform data should precede the automatic performance start), and then the writing block pointer WP is set to a value equivalent to a sum of a current value of the writing block pointer WP and the number of
15 counts by which the data writing should precede the data reading. After this, the write-enable flag is set to "1" so that leading waveform data start being written into the block pointed to by the writing block pointer WP.

Once the specific number of the calculation triggering
20 clock pulses B are generated and hence the reading block pointer RP reaches the block where the waveform data writing was initiated, the automatic performance is started (i.e., the read-enable flag RF is set to "1") and the waveform data are read out properly from the beginning. By virtue of the
25 arrangement that the data writing is executed (i.e., the write-enable flag WF is set to "1") prior to the start of the

automatic performance (i.e., the read-enable flag RF is set to "1"), the function of executing the data writing (calculation) in advance of the data reading can be utilized effectively from the starting point of the automatic
5 performance.

In summary, according to the above-described embodiment, there are provided a plurality of storage regions (blocks) for storing waveform data (waveform sample values), waveform data to be supplied to the D/A converter 19 are generated
10 collectively in advance during a period when the amount of calculation necessary for another software processing executed in parallel with the reproduction processing software is small (i.e., when the CPU 10 is not busy with the other software processing), and the thus-generated waveform
15 data are stored in the blocks of the sample buffer. Thus, when the calculation amount of the other software processing increases temporarily, waveform data generating operations to be performed at that time can be skipped without any trouble since waveform data to be supplied to the
20 reproduction section 18 at that timing have been generated and saved previously. This prevents operational delays in the other software processing. Because generation and supply to the reproduction section 18 of waveform data are conducted on a block-by-block basis, a determination as to whether
25 there is time available is greatly facilitated and a plurality of waveform data can be generated collectively,

which provide increased operating efficiency.

Each of the blocks of the sample buffer has been described as storing 128 waveform samples, but, where the system is designed to not accept real-time performance data input, each of the blocks may be designed to store a greater number of samples, e.g., 1,024 or 4,096 samples, so as to permit the CPU 10 to operate more efficiently. However, where the system accepts real-time performance data input, designing each of the blocks to store such a greater number of samples is not preferable in that intervals between the calculation triggering clock pulses BC become longer and hence greater time lags will occur from the real-time performance data input to actual sounding of a tone corresponding thereto.

15 The present invention arranged in the above-described manner achieves the following benefits.

According to the present invention, a detection is made of calculating capability of the CPU 10 or arithmetic processing section that is unoccupied by other software processing and hence available for the reproduction processing and a specific number of tone waveform sample values corresponding to the unoccupied or available calculating capability are generated prior to predetermined readout timing of the sample values. Thus, when a plurality of applications are run concurrently on a multitask basis, it is allowed to calculate waveform data collectively for

memory storage utilizing the unoccupied or available calculating capability, which permits the CPU or arithmetic processing section to operate with greatly increased efficiency. Because it is just sufficient that the
5 previously stored waveform data be read out to be transmitted to the reproduction section 18 when the arithmetic processing section is busy with another application, processing by the other application can be performed efficiently without being influenced by the waveform sample calculation.

10 Further, because a predetermined number of tone waveform sample values are set as a basic calculating unit and tone waveform sample values are actually calculated on a unit-by-unit basis, it is possible to reduce the overhead spent in preparing for the waveform value calculating processing etc.

15 Furthermore, because a predetermined number of tone waveform sample values are already prepared and stored in memory at a starting point of an automatic performance, additional adjusting functions by the advanced calculation can be performed efficiently from the starting point of an automatic
20 performance.

Moreover, even when the advanced calculation of tone waveform sample values can not be conducted due to great loads placed on the arithmetic processing section by other processing, the present invention can prevent an unwanted
25 break in generated sounds because the waveform calculating steps are always taken to prepare tone waveform sample

values.

In addition, the present invention is characterized in that when calculating tone waveform sample values, a detection is made of an amount of calculation necessary for the arithmetic processing section to conduct other processing and the waveform sample value calculation is executed with different calculating precision which is selectable in accordance with the detected calculation amount for the other processing. Even when the arithmetic processing section is busy with the other processing, loads on the arithmetic processing section can be effectively lessened by selecting low calculating precision to reduce the amount of tone waveform sample calculation. As a result, generation of tone waveform data can be continued with no break and without influencing the other processing.

Moreover, according to the present invention, tone generating processing based on an automatic performance is executed in advance during a period when tone generating processing based on a real-time performance is not placing heavy burdens on the arithmetic processing section. As a result, the burdens on the arithmetic processing section can be distributed timewise, which achieves greatly increased operational efficiency of the arithmetic processing section.



- 28 -

C L A I M S

1. A tone waveform synthesizer comprising:

5 waveform generation means for collectively generating
a plurality of tone waveform samples in advance of
predetermined reproduction timing;

buffer means for temporarily storing the plurality of
tone waveform samples generated by said waveform generation
means, said buffer means being capable of writing and
10 reading, independently of each other, in a parallel fashion;
and

control means for controlling writing of the plurality
of tone waveform samples into said buffer means and also
performing control to read out the tone waveform samples from
15 said buffer means at said reproduction timing, said control
means controlling the writing and reading into and from said
buffer means independently of each other and also in such a
manner that an advance of the writing does not outpace an
advance of the reading.

20 2. A tone waveform synthesizer as recited in claim 1
wherein said buffer means performs the writing and reading,
independently of each other, in accordance with separate
write instruction and readout instruction signals,
respectively, and

25 wherein said control means controls the write



- 29 -

instruction signal so that the tone waveform samples are sequentially written into said buffer means from a beginning of a predetermined storage region of said buffer means and, upon arrival at an end of the predetermined storage region, the writing of the tone waveform samples returns to the beginning of the predetermined storage region and also that a writing location, in the storage region, indicated by the write instruction signal does not get ahead of a reading location, in the storage region, indicated by the read instruction signal.

3. A method of generating a tone waveform based on performance information, using a processor executing a tone waveform forming process, said method comprising:

- a step of receiving automatic performance information;
- 15 a step of receiving real-time performance information generated in response to a real-time performance operation;
- a step of generating tone waveform samples using said processor, said step of generating including a step of collectively generating a plurality of tone waveform samples
- 20 in advance of predetermined reproduction timing on the basis of the received automatic performance information and a step of collectively generating a plurality of tone waveform samples in advance of predetermined reproduction timing on the basis of the received real-time performance information,
- 25 said step of generating being capable of generating the tone



- 30 -

waveform samples based on the automatic performance information and the tone waveform samples based on the real-time performance information in a parallel fashion; and

5 a step of outputting the tone waveform samples generated by said step of generating.

4. A method as recited in claim 3 wherein said step of outputting includes a step of mixing the tone waveform samples based on the automatic performance information and
10 the tone waveform samples based on the real-time performance information to thereby provide mixtures of the samples, each of the mixtures being composed of the tone waveform samples to be reproduced at same reproduction timing, and a step of writing the mixtures into an output buffer.

15 5. A method as recited in claim 3 wherein said step of generating performs arithmetic operations for collectively generating the tone waveform samples based on the automatic performance information and the tone waveform samples based on the real-time performance information, independently of
20 each other, in such a manner that an advance of the arithmetic operations for the tone waveform samples based on the automatic performance information and an advance of the arithmetic operations for the tone waveform samples based on the real-time performance information differ from each other.



- 31 -

6. A method as recited in claim 5 wherein the arithmetic operations for the tone waveform samples based on the automatic performance information is advanced within an extent of an available processing capability taking into
5 account a current processing capability of said processor.

7. A method of generating a tone waveform using a processor capable of executing a plurality of different programs on a time divisional basis, said method comprising:

a step of supplying said processor with application
10 software including at least an image control program for controlling image display, a music control program for controlling tone generation and a general control program, said music control program including tone color data; and

a step of causing said processor to execute, under
15 control by the general control program, the image control program and the music control program in a parallel fashion, to output image data generated as a result of execution of the image control program and tone waveform data generated as a result of execution of the music control program.

20 8. A method as recited in claim 7 wherein the tone color data included in the music control program contains waveform data pertaining to at least one given tone color.



- 32 -

9. A method as recited in claim 7 wherein the music control program includes a tone waveform generating program for generating tone waveform samples on the basis of performance information.

5 10. A method as recited in claim 7 wherein the application software is software directed to a karaoke or game.

11. A method as recited in claim 7 wherein said step of supplying said processor with application software supplies the application software to said processor via a
10 communication network.

12. A method as recited in claim 7 wherein said step of supplying said processor with application software supplies the application software to said processor by setting, in said processor, a transportable medium storing the
15 application software.

13. A method as recited in claim 7 wherein said step of causing said processor to execute includes:

a step of detecting an available time portion in which said processor is not currently used for any other process
20 than a tone waveform generating process, as an available processing capability for the tone waveform generating process; and



- 33 -

a step of causing said processor to execute the tone waveform generating process based on the music control program, using the available processing capability detected by said step of detecting.