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

(11) Publication number: **0 463 411 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **91109140.3**

(51) Int. Cl.5: **G10H 7/00**, G10H 1/18

(22) Date of filing: **04.06.91**

(30) Priority: **28.06.90 JP 171215/90**
29.06.90 JP 172200/90

(43) Date of publication of application:
02.01.92 Bulletin 92/01

(84) Designated Contracting States:
DE FR GB IT

(71) Applicant: **CASIO COMPUTER COMPANY LIMITED**
6-1, 2-chome, Nishi-Shinjuku
Shinjuku-ku Tokyo(JP)

(72) Inventor: **Usami, Ryuji, c/o Patent Department**
Casio Computer Co., Ltd., 3-2-1, Sakae-cho
Hamura-machi, Nishitama-gun,
Tokyo190-11(JP)
Inventor: **Shiba, Kosuke, c/o Patent Department**
Casio Computer Co., Ltd., 3-2-1, Sakae-cho
Hamura-machi, Nishitama-gun,
Tokyo190-11(JP)
Inventor: **Daigo, Koichiro, c/o Patent Department**
Casio Computer Co., Ltd., 3-2-1, Sakae-cho
Hamura-machi, Nishitama-gun,
Tokyo190-11(JP)
Inventor: **Ogura, Kazuo, c/o Patent Department**
Casio Computer Co., Ltd., 3-2-1, Sakae-cho
Hamura-machi, Nishitama-gun,
Tokyo190-11(JP)
Inventor: **Hosoda, Jun, c/o Patent Department**
Casio Computer Co., Ltd., 3-2-1, Sakae-cho
Hamura-machi, Nishitama-gun,
Tokyo190-11(JP)

(74) Representative: **KUHNEN, WACKER & PARTNER**
Alois-Steinecker-Strasse 22 Postfach 1553
W-8050 Freising(DE)

(54) Musical tone waveform generation apparatus.

(57) IN a musical tone waveform generation apparatus for outputting musical tone signals generated by a software program at predetermined time intervals, sound source method can be selected in units of tone generation channels. In the musical tone waveform generation apparatus, the sound source method or the tone color of a musical tone signal to be output is determined in accordance with performance data (pitch data, touch data, music part data, and the like).

EP 0 463 411 A2

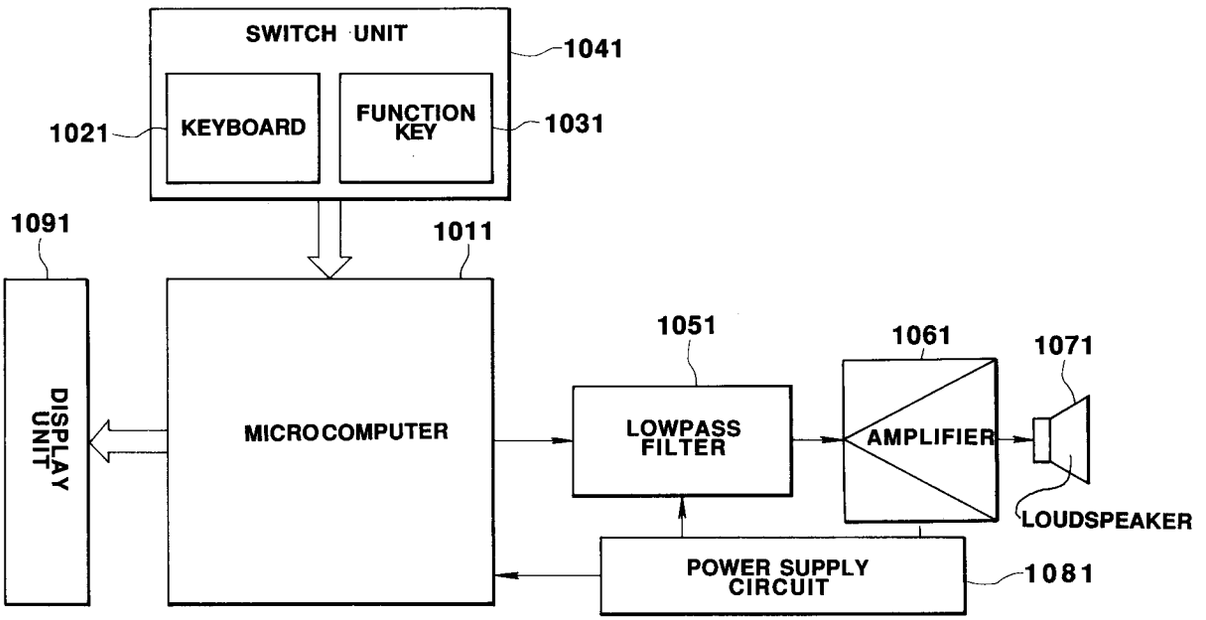
SWITCH UNIT
1041
KEYBOARD
FUNCTION KEY
1021
1031
1091
1011
DISPLAY UNIT
MICROCOMPUTER
1051
LOWPASS FILTER
1061
AMPLIFIER
1071
LOUDSPEAKER
POWER SUPPLY CIRCUIT
1081

FIG. 1

The present invention relates to a sound source processing method in a musical tone waveform generation apparatus and, more particularly, to a musical tone waveform generation apparatus capable of mixing a plurality of sound source methods.

Along with the development of the digital signal processing techniques and LSI processing techniques, various electronic musical instruments having good performance have been realized.

Since a musical tone waveform generation apparatus for an electronic musical instrument requires large-volume, high-speed digital calculations, a conventional apparatus is constituted by a special-purpose sound source circuit which realizes an architecture equivalent to a musical tone generation algorithm based on a required sound source method by hardware components. Such a sound source circuit generates a musical tone waveform on the basis of a PCM or modulation method.

The above-mentioned sound source circuit has a large circuit scale regardless of the sound source method adopted. When the sound source circuit is formed in an LSI, it has a scale about twice that of a versatile data processing microprocessor since the sound source circuit requires complicated address control for accessing waveform data on the basis of various performance data. Registers or the like for temporarily storing intermediate data obtained in the process of sound source generation processing must be arranged everywhere in the architecture corresponding to the sound source method. Furthermore, in order to realize a polyphonic arrangement capable of simultaneously generating a plurality of musical tones, shift registers or the like for time-divisionally executing sound source processing in a hardware manner must be arranged everywhere.

As described above, since the conventional musical tone waveform generation apparatus is constituted by the special-purpose sound source circuit corresponding to the sound source method, its hardware scale is undesirably increased. This results in an increase in manufacturing cost in terms of, e.g., a yield in the manufacture of LSI chips, when the sound source circuit is realized by an LSI. This also results in an increase in size of the musical tone waveform generation apparatus.

When a sound source method is to be changed, or when the number of polyphonic channels is to be increased, the sound source circuit must be considerably modified, resulting in an increase in development cost.

When the conventional musical tone waveform generation apparatus is realized as an electronic musical instrument, a control circuit, comprising, e.g., a microprocessor, for generating, based on performance data corresponding to a performance operation, data which can be processed by the sound source circuit, and for communicating performance data with another musical instrument, is required. The control circuit requires a sound source control program, corresponding to the sound source circuit, for supplying data corresponding to performance data to the sound source circuit in addition to a performance data processing program for processing performance data. In addition, these two programs must be synchronously operated. The development of such complicated programs causes a considerable increase in cost.

On the other hand, in recent years, a large number of high-performance microprocessors for performing versatile data processing have been developed, and a musical tone waveform generation apparatus for exectuting sound source processing in a software manner using such a microprocessor may be realized. However, no technique for synchronously operating a performance data processing program for processing performance data, and a sound source processing program for executing sound source processing on the basis of the performance data is available. In particular, since a processing time in the sound source processing program varies depending on the sound source method, a complicated timing control program for outputting generated musical tone data to a D/A converter is required. When the sound source processing is merely performed in a software manner, the processing programs are complicated very much, and processing of the high-speed sound source method such as a modulation method cannot be executed in terms of a processing speed and a program capacity. In particular, high-grade sound source processing for switching sound source methods in units of tone generation channels, and generating tones in different sound source methods in accordance with performance data so as to generate a real musical tone waveform having a complicated frequency structure like musical tones generated by an acoustic instrument cannot be performed.

Furthermore, a player sometimes wants to make a performance with a plurality of instrument tone colors by himself or herself to meet his or her requirements on a performance. In this case, the following processing is required. That is, a split point is determined for tone ranges or velocities of ON keys of an electronic musical instrument, so that musical tones in a plurality of instrument tone colors can be generated in accordance with a range having the split point as a boundary to which the tone range or velocity belongs, thus attaining complicated colorful musical expressions. However, simple software processing cannot attain such high-grade sound source method processing. It is

also difficult to execute processing for generating tones in different instrument tone colors in units of music parts.

It is an object of the present invention to attain high-grade sound source processing which can assign different sound source methods to a plurality of tone generation channels under the program control of a microprocessor without requiring a special-purpose sound source circuit.

It is another object of the present to allow generation of musical tone signals in different tones or different sound source methods in units of regions, or operation velocities, or music parts having a split point as a boundary under the program control of a microprocessor without requiring a special-purpose sound source circuit.

According to the first aspect of the present invention, there is provided a musical tone waveform generation apparatus comprising: storage means for storing a plurality of sound source processing programs corresponding to a plurality of types of sound source methods; musical tone signal generation means for generating musical tone signals in arbitrary sound source methods in tone generation channels by executing the plurality of sound source programs stored in the storage means; and musical tone signal output means for outputting the musical tone signals generated by the musical tone signal generation means at predetermined output time intervals.

According to the musical tone waveform generation apparatus of the first aspect of the present invention, high-grade sound source processing which can assign different sound source methods to a plurality of tone generation channels without using a special-purpose sound source circuit can be performed. Since a constant output rate of a musical tone signal can be maintained upon operation of the musical tone signal output means, a musical tone waveform will not be distorted.

According to the second aspect of the present invention, there is provided a musical tone waveform generation apparatus comprising: program storage means for storing a performance data processing program for processing performance data, and a plurality of sound source processing programs corresponding to a plurality of sound source methods for obtaining a musical tone signal; address control means for controlling an address of the program storage means; data storage means for storing musical tone generation data necessary for generating a musical tone signal by an arbitrary one of the plurality of sound source methods in units of tone generation channels; arithmetic processing means for performing a predetermined arithmetic operation; program execution means for executing the performance data processing program and the sound source processing program stored in the program storage means while controlling the address control means, the data storage means, and the arithmetic processing means, for normally executing the performance data processing program to control musical tone generation data on the data storage means, for executing the sound source processing program at predetermined time intervals, for executing the performance data processing program again upon completion of the sound source processing program, and for executing time-divisional processing on the basis of musical tone generation data on the data storage means upon execution of the sound source processing program so as to generate musical tone signals by the sound source methods assigned to the tone generation channels; and musical tone signal output means for holding the musical tone signals obtained upon execution of the sound source processing programs by the program execution means, and outputting the held musical tone signals at predetermined output time intervals.

In the musical tone waveform generation apparatus according to the second aspect of the present invention, the program storage means, the address control means, the data storage means, the arithmetic processing means, and the program execution means have the same arrangement as a versatile microprocessor, and no special-purpose sound source circuit is required at all. The musical tone signal output means is versatile in the category of a musical tone waveform generation apparatus although it has an arrangement different from that of a versatile microprocessor.

The circuit scale of the overall musical tone waveform generation apparatus can be greatly reduced, and when the apparatus is realized by an LSI, the same manufacturing technique as that of a normal processor can be adopted. Since the yield of chips can be increased, manufacturing cost can be greatly reduced. Since the musical tone signal output means can be constituted by simple latch circuits, addition of this circuit portion causes almost no increase in manufacturing cost.

When a modulation method is required to be switched, or when the number of polyphonic channels is required to be changed, a sound source processing program stored in the program storage means need only be changed to meet the above requirements. Therefore, the development cost of a new musical tone waveform generation apparatus can be greatly reduced, and a new modulation method can be presented to a user by means of, e.g., a ROM card.

The above-mentioned effects can be provided since the second aspect of the present invention can realize the following program and data architectures.

More specifically, the musical tone waveform

generation apparatus according to the second aspect of the present invention realizes a data architecture in which musical tone generation data necessary for generating musical tones are stored on the data storage means. When a performance data processing program is executed, corresponding musical tone generation data on the data storage means are controlled, and when a sound source processing program is executed, musical tone signals are generated on the basis of the corresponding musical tone generation data on the data storage means. In this manner, a data communication between the performance data processing program and the sound source processing program is performed via musical tone generation data on the data storage means, and access of one program to the data storage means can be performed regardless of an execution state of the other program. Therefore, the two programs can have substantially independent module arrangements, and hence, a simple and efficient program architecture can be attained.

In addition to the data architecture, the musical tone waveform generation apparatus according to the second aspect of the present invention realizes the following program architecture. That is, the performance data processing program is normally executed to execute, e.g., scanning of keyboard keys and various setting switches, demonstration performance control, and the like. During execution of this program, the sound source processing program is executed at predetermined time intervals, and upon completion of the processing, the control returns to the performance data processing program. Thus, the sound source processing program forcibly interrupts the performance data processing program on the basis of an interrupt signal generated from the interrupt control means at predetermined time intervals. For this reason, the performance data processing program and the sound source processing program need not be synchronized.

When the program execution means executes the sound source processing program, its processing time changes depending on sound source methods. However, the change in processing time can be absorbed by the musical tone signal output means. Therefore, no complicated timing control program for outputting musical tone signals to, e.g., a D/A converter is required.

As described above, the data architecture for attaining a data link between the performance data processing program and the sound source processing program via musical tone generation data on the data storage means, and the program architecture for executing the sound source processing program at predetermined time intervals while interrupting the performance data processing program are realized, and the musical tone signal output means is arranged. Therefore, sound source processing under the efficient program control can be realized by substantially the same arrangement as a versatile processor.

Furthermore, the data storage means stores musical tone generation data necessary for generating musical tone signals in an arbitrary one of a plurality of sound source methods in units of tone generation channels, and the program execution means executes the performance data processing program and the sound source processing program by time-divisional processing in correspondence with the tone generation channels. Therefore, the program execution means accesses the corresponding musical tone generation data on the data storage means at each time-divisional timing, and executes a sound source processing program of the assigned sound source method while simply switching the two programs. In this manner, musical tone signals can be generated by different sound source methods in units of tone generation channels.

In this manner, according to the second aspect of the present invention, musical tone signals can be generated by different sound source methods in units of tone generation channels under the simple control, i.e., by simply switching between time-divisional processing for musical tone generation data in units of tone generation channels on the data storage means, and a sound source processing program based on the musical tone generation data.

According to the third aspect of the present invention, there are provided a musical tone waveform generation apparatus comprising: storage means for storing a sound source processing program; musical tone signal generation means for executing the sound source processing program stored in the storage means to generate a musical tone signal; pitch designation means for designating a pitch of the musical tone signal generated by the musical tone signal generation means; tone color determination means for determining a tone color of the musical tone signal generated by the musical tone signal generation means in accordance with the pitch designated by the pitch designation means; control means for controlling the musical tone signal generation means to generate the musical tone signal having the pitch designated by the pitch designation means and the tone color determined by the tone color determination means; and musical tone signal output means for outputting the musical tone signal generated by the musical tone signal generation means at predetermined time intervals.

According to the fourth aspects of the present invention, there are provided a musical tone

waveform generation apparatus comprising: storage means for storing a sound source processing program; musical tone signal generation means for executing the sound source processing program stored in the storage means to generate a musical tone signal; a performance operation member for instructing the musical tone signal generation means to generate the musical tone signal; tone color determination means for determining a tone color of the musical tone signal to be generated by the musical tone signal generation means in accordance with an operation velocity of the performance operation member; control means for controlling the musical tone signal generation means to generate the musical tone signal having the tone color determined by the tone color determination means; and musical tone signal output means for outputting the musical tone signal generated by the musical tone signal generation means at predetermined time intervals.

According to the fifth aspect of the present invention, there are provided a musical tone waveform generation apparatus comprising: storage means for storing a sound source processing program; musical tone signal generation means for executing the sound source processing program stored in the storage means to generate a musical tone signal; output means for outputting performance data of a plurality of parts constituting a music piece; tone color determination means for determining a tone color of the musical tone signal to be generated by the musical tone signal generation means in accordance with one of the plurality of parts to which the performance data output from the output means belongs; control means for controlling the musical tone generation means to generate the musical tone signal having the tone color determined by the tone color determination means; and musical tone signal output means for outputting the musical tone signal generated by the musical tone signal generation means at predetermined time intervals.

According to the musical tone waveform generation apparatuses of the third, fourth, and fifth aspects of the present invention, musical tone signals can be generated in different tone colors in units of regions, or operation velocities, or musical parts having a split point as a boundary without using a special-purpose sound source circuit. Since a constant output rate of musical tone signals can be maintained upon operation of the musical tone signal output means, a musical tone waveform will not be distorted.

According to the sixth aspect of the present invention, there are provided a musical tone waveform generation apparatus comprising: program storage means for storing a performance data processing program for processing performance data, and a sound source processing program for obtaining a musical tone signal; address control means for controlling an address of the program storage means; split point designation means for causing a player to designate a split point to divide a range of a performance data value into a plurality of ranges; tone color designation means for designating tone colors of the plurality of ranges having the split point designated by the split point designation means as a boundary; data storage means for storing musical tone generation data necessary for generating the musical tone signal in correspondence with a plurality of tone colors; arithmetic processing means for processing data; program execution means for executing the performance data processing program and the sound source processing program stored in the program storage means while controlling the address control means, the data storage means, and the arithmetic processing means, for normally executing the performance data processing program to control musical tone generation data stored in the data storage means, for executing the sound source processing program at predetermined time intervals, for executing the performance data processing program again upon completion of the sound source processing program, and for generating, upon execution of the sound source processing program, the musical tone signal on the basis of the musical tone generation data on the data storage means corresponding to the tone color designated by the tone color designation means in correspondence with the range which has the split point designated by the split point designation means as a boundary, and to which the performance data value belongs; and musical tone signal output means for holding the musical tone signals in units of tone generation operations obtained upon execution of the sound source processing program by the program execution means, and outputting the held musical tone signals at predetermined output time intervals.

According to the seventh aspect of the present invention, there are provided a musical tone waveform generation apparatus comprising: program storage means for storing a performance data processing program for processing performance data, and a plurality of sound source processing programs corresponding to a plurality of sound source methods for obtaining a musical tone signal; address control means for controlling an address of the program storage means; split point designation means for causing a player to designate a split point to divide a range of a performance data value into a plurality of ranges; sound source method designation means for causing the player to designate the sound source methods for the divided ranges having the split point designated by the

split point designation means as a boundary; data storage means for storing musical tone generation data necessary for generating the musical tone signal in correspondence with the plurality of sound source methods; arithmetic processing means for processing data; program execution means for executing the performance data processing program or the sound source processing program stored in the program control means while controlling the address control means, the data storage means, and the arithmetic processing means, for normally executing the performance data processing program to control musical tone generation data on the data storage means, for executing the sound source processing program at predetermined time intervals, for executing the performance data processing program again upon completion of the sound source processing program, and for generating, upon execution of the sound source processing program, the musical tone signal on the basis of the musical tone generation data corresponding to the sound source method corresponding to the range to which the performance data value belongs, and by the sound source processing program corresponding to the sound source method; and musical tone signal output means for holding the musical tone signals obtained upon execution of the sound source processing programs by the program execution means, and outputting the held musical tone signals at predetermined output time intervals.

According to the eighth aspects of the present invention, there are provided a musical tone waveform generation apparatus comprising: program storage means for storing a performance data processing program for processing performance data, and a sound source processing program for obtaining a musical tone signal; address control means for controlling an address of the program storage means; tone color designation means for causing a player to designate tone colors in units of music parts of musical tone signals to be played; data storage means for storing musical tone generation data necessary for generating a musical tone signal in an arbitrary one of the plurality of tone colors; arithmetic processing means for processing data; program execution means for executing the performance data processing program and the sound source processing program stored in the program control means while controlling the address control means, the data storage means, and the arithmetic processing means, for normally executing the performance data processing program to control musical tone generation data on the data storage means, for executing the sound source processing program at predetermined time intervals, for executing the performance data processing program again upon completion of the sound source processing program, and for generating, upon execution of the sound source processing program, the musical tone signal on the basis of the musical tone generation data on the data storage means corresponding to the tone color designated by the tone color designation means in correspondence with the music part of the musical tone signal generated by the sound source processing program; and musical tone signal output means for holding the musical tone signals in units of tone generation operations obtained upon execution of the sound source processing program by the program execution means, and outputting the held musical tone signals at predetermined output time intervals.

According to the ninth aspect of the present invention, there are provided a musical tone waveform generation apparatus comprising: program storage means for storing a performance data processing program for processing performance data, and a plurality of sound source processing programs corresponding to a plurality of sound source methods for obtaining a musical tone signal; address control means for controlling an address of the program storage means; sound source method designation means for causing a player to designate sound source methods in units of music parts of musical tone signals to be played; data storage means for storing musical tone generation data necessary for generating a musical tone signal by an arbitrary one of the plurality of sound source methods; arithmetic processing means for processing data; program execution means for executing the performance data processing program and the sound source processing program stored in the program control means while controlling the address control means, the data storage means, and the arithmetic processing means, for normally executing the performance data processing program to control musical tone generation data on the data storage means, for executing the sound source processing program at predetermined time intervals, for executing the performance data processing program again upon completion of the sound source processing program, and for generating, upon execution of the sound source processing program, the musical tone signal on the basis of the musical tone generation data corresponding to the sound source method corresponding to the music part of the musical tone signal generated by the sound source processing program, and by the sound source processing program corresponding to the sound source method; and musical tone signal output means for holding the musical tone signals obtained upon execution of the sound source processing programs by the program execution means, and outputting the held musical tone signals at predetermined output time intervals.

According to the musical tone waveform generation apparatuses according to the sixth and seventh aspects of the present invention, a player can designate a split point, and can also designate tone colors or sound source methods in units of ranges having the designated split point as a boundary, so that musical tone signals can be generated by switching the corresponding tone colors or sound source methods in accordance with the above-described ranges of predetermined performance data.

According to the musical tone waveform generation apparatuses according to the eighth and ninth aspects of the present invention, tone colors or sound source methods can also be switched in accordance with not a split point but music parts.

This invention can be more fully understood from the following detailed description when taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a block diagram showing the overall arrangement according to the first embodiment of the present invention;
Fig. 2 is a block diagram showing the internal arrangement of a microcomputer;
Fig. 3 is a block diagram of a conventional D/A converter unit;
Fig. 4 is a block diagram of a D/A converter unit according to the first embodiment;
Fig. 5 is a timing chart in D/A conversion;
Figs. 6 to 8 are flow charts showing the overall operations of the first embodiment;
Fig. 9 is a schematic chart showing the relationship between the main operation flow chart and interrupt processing;
Fig. 10 is a view showing storage areas in units of tone generation channels on a RAM;
Fig. 11 is a schematic chart when a sound source processing method of each tone generation channel is selected;
Fig. 12 shows a data format in units of sound source methods on the RAM;
Fig. 13 is an operation flow chart of sound source processing based on a PCM method;
Fig. 14 is an operation flow chart of sound source processing based on a DPCM method;
Figs. 15 and 16 are charts for explaining the principle when an interpolation value X_Q is calculated using a difference D and a present address A_F in the PCM and DPCM methods, respectively;
Fig. 17 is an operation flow chart of sound source processing based on an FM method;
Fig. 18 is a chart showing an algorithm of the sound processing method based on the FM method;
Fig. 19 is an operation flow chart of sound source processing based on a TM method;
Fig. 20 is a chart showing an algorithm of the sound source processing based on the TM method;
Fig. 21 is a view showing an arrangement of some function keys (Part 1);
Fig. 22 is a view showing a data architecture of tone color parameters;
Fig. 23 is a view showing an arrangement of a buffer B and registers X and Y on a RAM 2061;
Fig. 24 is an explanatory view of keyboard keys (64 keys);
Fig. 25 is an operation flow chart of an embodiment A of keyboard key processing;
Fig. 26 is an operation flow chart of an embodiment B of keyboard key processing;
Fig. 27 is a view showing an arrangement of some function keys (Part 2);
Fig. 28 is an operation flow chart of an embodiment C of keyboard key processing;
Fig. 29 is an operation flow chart of an embodiment D of keyboard key processing;
Fig. 30 is an operation flow chart of an embodiment A of demonstration performance processing;
Fig. 31 is an operation flow chart of an embodiment B of demonstration performance processing;
Figs. 32 and 33 are views showing assignment methods of X and Y tone colors to tone generation channels;
Fig. 34 is a block diagram showing the overall arrangement according to an embodiment of the present invention;
Fig. 35 is a block diagram showing an internal arrangement of a master CPU;
Fig. 36 is a block diagram showing an internal arrangement of a slave CPU;
Figs. 37 to 40 are flow charts showing operations of the overall arrangement of this embodiment;
Fig. 41 is a schematic view showing the relationship among the main operation flow charts and interrupt processing;
Fig. 42 is a diagram of a conventional D/A converter unit;
Fig. 43 is a diagram of a D/A converter unit according to this embodiment;
Fig. 44 is a timing chart in D/A conversion;
Fig. 45 illustrates an arrangement of a function key and a keyboard key;
Fig. 46 is an explanatory view of keyboard keys;
Fig. 47 shows storage areas in units of tone generation channels on a RAM;
Fig. 48 is a schematic diagram upon selection of a sound source processing method of each tone generation channel;
Fig. 49 shows an architecture of data formats in units of sound source methods on the RAM;

Fig. 50 shows buffer areas on the RAM;
Figs. 51 to 54 are charts showing algorithms in a modulation method;
Fig. 55 is an operation flow chart of sound source processing based on an FM method (Part 2);
Fig. 56 is an operation flow chart of sound source processing based on a TM method (Part 2);
Fig. 57 is an operation flow chart of a first modification of the modulation method;
Fig. 58 is an operation flow chart of operator 1 processing based on the FM method according to the first modification;
Fig. 59 is a chart showing an arithmetic algorithm per operator in the operator 1 processing based on the FM method according to the first modification;
Fig. 60 is an operation flow chart of operator 1 processing based on the TM method according to the first modification;
Fig. 61 is a chart showing an arithmetic algorithm per operator in the operator 1 processing based on the TM method according to the first modification;
Fig. 62 is an operation flow chart of algorithm processing according to the first modification;
Fig. 63 is an operation flow chart of a second modification of the modulation method;
Fig. 64 is an operation flow chart of algorithm processing according to the second modification;
Fig. 65 shows an arrangement of some function keys;
Figs. 66 and 67 show examples of assignments of sound source methods to tone generation channels;
Fig. 68 is an operation flow chart of function key processing;
Fig. 69 is an operation flow chart of an embodiment A of ON event keyboard key processing;
Fig. 70 is an operation flow chart of an second embodiment B of ON event keyboard key processing; and
Fig. 71 is an operation flow chart of an embodiment of OFF event keyboard key processing.

[First Embodiment]

The first embodiment of the present invention will be described below with reference to the accompanying drawings.

Arrangement of the First Embodiment

Fig. 1 is a block diagram showing the overall arrangement according to the first embodiment of the present invention.

In Fig. 1, the entire apparatus is controlled by a microcomputer 1011. In particular, not only control input processing for an instrument but also processing for generating musical tones are executed by the microcomputer 1011, and no sound source circuit for generating musical tones is required.

A switch unit 1041 comprising a keyboard 1021 and function keys 1031 serves as an operation/input section of a musical instrument, and performance data input from the switch unit 1041 are processed by the microcomputer 1011. Note that the function keys 1031 will be described in detail later.

A display unit 1091 includes red and green LEDs indicating which tone color on the function keys 1031 is designated when a player determines a split point and sets different tone colors to keys as will be described later. The display unit 1091 will be described in detail later in a description of Fig. 21 or 26.

An analog musical tone signal generated by the microcomputer 1011 is smoothed by a low-pass filter 1051, and the smoothed signal is amplified by an amplifier 1061. Thereafter, the amplified signal is produced as a tone via a loudspeaker 1071. A power supply circuit 1081 supplies a necessary power supply voltage to the low-pass filter 1051 and the amplifier 1061.

Fig. 2 is a block diagram showing the internal arrangement of the microcomputer 1011.

A control data/waveform data ROM 2121 stores musical tone control parameters such as target values of envelope values (to be described later), musical tone waveform data in respective sound source methods, musical tone difference data, modulated waveform data, and the like. A command analyzer 207 accesses the data on the control data/waveform data ROM 2121 while sequentially analyzing the content of a program stored in a control ROM 2011, thereby executing software sound source processing.

The control ROM 2011 stores a musical tone control program (to be described later), and sequentially outputs program words (commands) stored at addresses designated by a ROM address controller 2051 via a ROM address decoder 2021. More specifically, the word length of each program word is 28 bits, and a next address method is employed. In this method, a portion of each program word is input to the ROM address controller 2051 as lower bits (intra-page address) of an address to be read out next. Note that the control ROM 2011 may comprise a CPU of a conventional program counter type.

The command analyzer 2071 analyzes operation codes of commands output from the control ROM 2011, and supplies control signals to the respective units of the circuit so as to execute the

designated operations.

When an operand of a command from the control ROM 2011 designates a register, a RAM address controller 2041 designates an address of a corresponding register in a RAM 2061. The RAM 2061 stores various musical tone control data (to be described later with reference to Figs. 9 and 10) for eight tone generation channels, and various buffers (to be described later), and is used in sound source processing (to be described later).

When a command from the control ROM 2011 is an arithmetic command, an ALU unit 2081 and a multiplier 2091 respectively execute a subtraction/addition and logic arithmetic operation, and a multiplication on the basis of an instruction from the command analyzer 2071.

An interrupt controller 2031 supplies an interrupt signal to the ROM address controller 2051 and a D/A converter unit 2131 at predetermined time intervals on the basis of an internal hardware timer (not shown).

An input port 2101 and an output port 2111 are connected to the switch unit 1041 and the display unit 1091 (Fig. 1).

Various data read out from the control ROM 2011 or the RAM 2061 are supplied to the ROM address controller 2051, the ALU unit 2081, the multiplier 2091, the control data/waveform data ROM 2121, the D/A converter unit 2131, the input port 2101, and the output port 2111 via a bus. The outputs from the ALU unit 2081, the multiplier 2091, and the control data/waveform data ROM 2121 are supplied to the RAM 2061 via the bus.

Fig. 4 shows the internal arrangement of the D/A converter unit 2131 shown in Fig. 1. Data of musical tones for one sampling period generated by sound source processing are input to a latch 3011 via a data bus. When the clock input of the latch 3011 receives a sound processing end signal from the command analyzer 2071 (Fig. 2), the musical tone data for one sampling period on the data bus are latched by the latch 3011, as shown in Fig. 5.

Since a time required for the sound source processing changes depending on execution conditions of sound source processing software, a timing at which the sound source processing is ended, and the musical tone data are latched by the latch 3011 is not fixed. For this reason, as shown in Fig. 3, the output from the latch 301 cannot be directly input to a D/A converter 3031.

In the first embodiment, as shown in Fig. 4, the musical tone signals output from the latch 3011 are latched by a latch 3021 in response to interrupt signals equal to a sampling clock interval, which signals are output from the interrupt controller 2031 (Fig. 2), and are output to the D/A converter 3031 at predetermined time intervals.

Since a change in processing time in the respective sound source methods can be absorbed by using the two latches, a complicated timing control program for outputting musical tone data to the D/A converter can be omitted.

Overall Operation of the First Embodiment

The overall operation of the first embodiment will be described below.

In the first embodiment, the microcomputer 1011 repetitively executes a series of processing operations in steps S_{502} to S_{510}, as shown in the main flow chart of Fig. 6. Sound source processing is executed as interrupt processing in practice. More specifically, the program executed as the main flow chart shown in Fig. 6 is interrupted at predetermined time intervals, and a sound source processing program for generating musical tone signals for eight channels is executed based on the interrupt. Upon completion of this processing, the musical tone signals for eight channels are added to each other, and the sum signal is output from the D/A converter unit 2131 shown in Fig. 2. Thereafter, the control returns from the interrupt state to the main flow. Note that the above-described interrupt operation is periodically performed on the basis of the internal hardware timer in the interrupt controller 2031 (Fig. 2). This period is equal to the sampling period when musical tones are output.

The schematic operation of the first embodiment has been described. The overall operation of the first embodiment will be described in detail below with reference to Figs. 6 to 8.

The main flow chart of Fig. 6 shows a flow of processing operations other than the sound source processing, which are executed by the microcomputer 1011 in a non-interrupt state from the interrupt controller 2031.

The power switch is turned on, and the contents of the RAM 2061 (Fig. 2) in the microcomputer 1011 are initialized (S_{501}).

Switches of the function keys 1031 (Fig. 1) externally connected to the microcomputer 1011 are scanned (S_{502}), and states of the respective switches are fetched from the input port 2101 to a key buffer area in the RAM 2061. As a result of scanning, a function key whose state is changed is discriminated, and processing of a corresponding function is executed (S_{503}). For example, a musical tone number and an envelope number are set, and if a rhythm performance function is presented as an optional function, a rhythm number is set.

Thereafter, ON keyboard key data on the keyboard 1021 (Fig. 1) are fetched in the same manner as the function keys described above (S_{504}), and keys whose states are changed are discriminated, thereby executing key assignment process-

ing (S_{505}). The keyboard key processing is particularly associated with the present invention, and will be described later.

When a demonstration performance key (not shown) of the function keys 1031 (Fig. 1) is depressed, demonstration performance data (sequencer data) are sequentially read out from the control data/waveform data ROM 2121 to execute, e.g., key assignment processing (S_{506}). When a rhythm start key is depressed, rhythm data are sequentially read out from the control data/waveform data ROM 2121 to execute, e.g., key assignment processing (S_{507}). The demonstration performance processing (S_{506}) and the rhythm processing (S_{507}) are also particularly associated with the present invention, and will be described in detail later.

Thereafter, timer processing to be described below is executed (S_{508}). More specifically, a value of time data which is incremented by interrupt timer processing (S_{512}) (to be described later) is discriminated. The time data value is compared with time control sequencer data sequentially read out for demonstration performance control or time control rhythm data read out for rhythm performance control, thereby executing time control when a demonstration performance in step S_{506} or a rhythm performance in step S_{507} is performed.

In tone generation processing in step S_{509}, pitch envelope processing, and the like are executed. In this processing, an envelope is added to a pitch of a musical tone to be subjected to tone generation processing, and pitch data is set in a corresponding tone generation channel.

Furthermore, one flow cycle preparation processing is executed (S_{510}). In this processing, processing for changing a state of a tone generation channel of a note number corresponding to an ON event detected in the keyboard key processing in step S_{505} to an ON event state, and processing for changing a state of a tone generation channel of a note number corresponding to an OFF event to a muting state, and the like are executed.

Interrupt processing will be described below with reference to Fig. 7.

When the program corresponding to the main flow shown in Fig. 6 is interrupted by the interrupt controller 2031 shown in Fig. 2, processing of the program is interrupted, and execution of the interrupt processing program shown in Fig. 7 is started. In this case, control is made to inhibit contents of registers to be subjected to write access in the main flow program in Fig. 6 from being rewritten in the interrupt processing program. Therefore, register save/restoration processing normally executed at the beginning and end of interrupt processing can be omitted. Thus, transition between the processing of the main flow chart shown in Fig. 6 and the interrupt processing can be quickly performed.

Subsequently, in the interrupt processing, sound source processing is started (S_{511}). The sound source processing is shown in Fig. 8. As a result, musical tone waveform data obtained by accumulating tones for eight tone generation channels is obtained in a buffer B (to be described later) of the RAM 2061 (Fig. 2).

In step S_{512}, interrupt timer processing is executed. In this processing, the value of time data (not shown) on the RAM 2061 (Fig. 2) is incremented by utilizing the fact that the interrupt processing shown in Fig. 7 is executed for every predetermined sampling period. More specifically, a time elapsed from power-on can be detected based on the value of the time data. The time data obtained in this manner is used in time control in the timer processing in step S_{508} in the main flow chart shown in Fig. 6, as described above.

In step S_{513}, the content of the buffer area is latched by the latch 3011 (Fig. 4) of the D/A converter unit 2131.

Operations of the sound source processing executed in step S_{511} in the interrupt processing will be described below with reference to the flow chart shown in Fig. 8.

A waveform addition area on the RAM 2061 is cleared (S_{513}). Then, sound source processing is executed in units of tone generation channels (S_{514} to S_{521}). After the sound source processing for the eighth channel is completed, waveform data obtained by adding those for eight channels is obtained in a predetermined buffer area B. These processing operations will be described in detail later.

Fig. 9 is a schematic flow chart showing the relationship among the processing operations of the flow charts shown in Figs. 6 and 7. Given processing A (the same applies to B, C,..., F) is executed (S_{601}). This "processing" corresponds to, e.g., "function key processing", or "keyboard key processing" in the main flow chart of Fig. 6. Thereafter, the control enters the interrupt processing, and sound source processing is started (S_{602}). Thus, a musical tone signal for one sampling period obtained by accumulating waveform data for eight tone generation channels can be obtained, and is output to the D/A converter unit 2131. Thereafter, the control returns to some processing B in the main flow chart.

The above-mentioned operations are repeated while executing sound source processing for each of eight tone generation channels (S_{604} to S_{611}). The repetition processing continues as long as musical tones are being produced.

Data Architecture in Sound Source Processing

The sound source processing executed in step S_{511} in Fig. 7 will be described in detail below.

In the first embodiment, the microcomputer 1011 executes sound source processing for eight tone generation channels. The sound source processing data for eight channels are set in areas in units of tone generation channels of the RAM 2061 (Fig. 2), as shown in Fig. 10.

The waveform data accumulation buffer B and tone color No. registers X and Y are allocated on the RAM 2061, as shown in Fig. 23.

In this case, a sound source method is set in (assigned to) each tone generation channel area shown in Fig. 10 by operations to be described in detail later, and thereafter, control data from the control data/waveform data ROM 2121 are set in the area in data formats in units of sound source methods, as shown in Fig. 12. The data formats in the control data/waveform data ROM 2121 will be described in detail later with reference to Fig. 22. In the first embodiment, different sound source methods can be assigned to tone generation channels, as will be described later.

In Table 1 showing the data formats of the respective sound source methods shown in Fig. 12, S indicates a sound source method No. as a number for identifying the sound source methods. A represents an address designated when waveform data is read out in the sound source processing, and A_I A_1, and A_2 represent integral parts of current addresses, and directly correspond to addresses of the control data/waveform data ROM 2121 (Fig. 2) where waveform data are stored. A_F represents a decimal part of the current address, and is used for interpolating waveform data read out from the control data/waveform data ROM 2121. A_E and A_L respectively represent end and loop addresses. P_I, P_1, and P_2 represent integral parts of pitch data, and P_F represents a decimal part of pitch data. For example, $P_I = 1$ and $P_F = 0$ express a pitch of an original tone, $P_I = 2$ and $P_F = 0$ express a pitch higher than the original pitch by one octave, and $P_I = 0$ and $P_F = 0.5$ express a pitch lower by one octave. X_P represents storage of previous sample data, and X_N represents storage of the next sample data. D represents a difference between magnitudes of two adjacent sample data, and E represents an envelope value. Furthermore, O represents an output value. Various other control data will be described later in descriptions of sound source methods.

In the first embodiment, when the main flow chart shown in Fig. 6 is executed, sound source method No. data, and control data necessary for sound source processing of the sound source method, e.g., pitch data, envelope data, and the like are set in a corresponding tone generation channel area. In the sound source processing shown in Fig. 8 executed as sound source processing in the interrupt processing shown in Fig. 7, musical tone generation processing is executed while using the control data set in the tone generation channel area. In this manner, a data communication between the main flow program and the sound source processing program is performed via control data (musical tone generation data) in the tone generation channel areas on the RAM 2061. For this reason, since access of one program to the tone generation channel area can be performed regardless of an execution state of the other program, the two programs can have substantially independent module arrangements, and hence, a simple and efficient program architecture can be attained.

The sound source processing operations of the respective sound source methods executed using the above-mentioned data architecture will be described below in turn. These sound source processing operations are realized by analyzing and executing a sound source processing program stored in the control ROM 2011 by the command analyzer 2071 of the microcomputer 1011. Assume that the processing is executed under this condition unless otherwise specified.

In the flow chart shown in Fig. 8, when the sound source processing (one of steps S_{517} to S_{524}) for each channel is started, the sound source method No. data S of the data in the data format (Table 1) shown in Fig. 12 stored in the corresponding tone generation channel area of the RAM 2061 is discriminated to determine sound source processing of a sound source method to be described below.

Sound Source Processing Based on PCM Method

When the sound source method No. data S indicates the PCM method, sound source processing based on the PCM method shown in the operation flow chart of Fig. 13 is executed. Variables in the flow chart are PCM data of Table 1 shown in Fig. 12, which data are stored in the corresponding tone generation channel area (Fig. 10) on the RAM 2061 (Fig. 2).

Of an address group on the control data/waveform data ROM 2121 (Fig. 2) where PCM waveform data are stored, an address where waveform data as an object to be currently processed is stored is assumed to be (A_I, A_F) shown in Fig. 15.

Pitch data (P_I, P_F) is added to the present address (S_{101}). The pitch data corresponds to the type of a key determined as an ON key of the keyboard 1021 shown in Fig. 1.

It is then checked if the integral part A_I of the sum address is changed (S_{1002}). If NO in step

S_{1002}, an interpolation data value O corresponding to the decimal part A_F of the address is calculated by arithmetic processing $D \times A_F$ using a difference D as a difference between sample data X_N and X_P at addresses $(A_I + 1)$ and A_I shown in Fig. 15 (S_{1007}). Note that the difference D has already been obtained by the sound source processing at the previous interrupt timing (see step S_{1006} to be described later).

The sample data X_P corresponding to the integral part A_I of the address is added to the interpolation data value O to obtain a new sample data value O (corresponding to X_Q in Fig. 15) corresponding to the current address (A_I, A_F) (S_{1008}).

Thereafter, the sample data is multiplied with the envelope value E (S_{1009}), and the content of the obtained interpolation data value O is added to the content of the waveform data buffer B (Fig. 23) in the RAM 2061 (Fig. 2) (S_{1010}).

Thereafter, the control returns to the main flow chart shown in Fig. 6. The control is interrupted in the next sampling period, and the operation flow chart of the sound source processing shown in Fig. 13 is executed again. Thus, pitch data (P_I, P_F) is added to the current address (A_I, A_F) (S_{1001}).

The above-mentioned operations are repeated until the integral part A_I of the address is changed (S_{1002}).

Before the integral part is changed, the sample data X_P and the difference D are left unchanged, and only the interpolation data value O is updated in with the address A_F. Thus, every time the address A_F is updated, new sample data X_Q is obtained.

If the integral part A_I of the current address is changed (S_{1002}) as a result of addition of the current address (A_I, A_F) and the pitch data (P_I, P_F) in step S_{1001}, it is checked if the address A_I has reached or exceeded the end address A_E (S_{1003}).

If YES in step S_{1003}, the next loop processing is executed. More specifically, a value (A_I - A_E) as a difference between the updated current address and the end address A_E is added to the loop address A_L to obtain a new current address (A_I, A_F). A loop reproduction is started from the integral part A_I of obtained new current address (S_{1004}). The end address A_E is an end address of an area of the control data/waveform data ROM 2121 (Fig. 2) where PCM waveform data are stored. The loop address A_L is an address of a position where a player wants to repeat an output of a waveform. With the above-mentioned operations, known loop processing is realized by the PCM method.

If NO in step S_{1003}, the processing in step S_{1004} is not executed.

Sample data is then updated. In this case, sample data corresponding to the new updated current address A_I and the immediately preceding address (A_I-1) are read out as X_N and X_P from the control data/waveform data ROM 2121 (Fig. 2) (S_{1005}).

Furthermore, the difference so far is updated with a difference D between the updated data X_N and X_P (S_{1006}).

The following operation is as described above.

In this manner, waveform data by the PCM method for one tone generation channel is generated.

Sound Source Processing Based on DPCM Method

The sound source processing based on the DPCM method will be described below.

The operation principle of the DPCM method will be briefly described below with reference to Fig. 16.

In Fig. 16, sample data X_P corresponding to an address A_I of the control data/waveform data ROM 2121 (Fig. 2) is obtained by adding sample data corresponding to an address (A_I-1) (not shown) to a difference between the sample data corresponding to the address (A_I-1) and sample data corresponding to the address A_I.

A difference D with sample data at the next address ($A_I + 1$) is written at the address A_I of the control data/waveform data ROM 2121. Sample data at the next address ($A_1 + 1$) is obtained by $X_P + D$.

In this case, if the current address is represented by A_F, as shown in Fig. 16, sample data corresponding to the current address $A_I + A_F$ is obtained by $X_P + D \times A_F$.

In this manner, in the DPCM method, a difference D between sample data corresponding to the current address and the next address is read out from the control data/waveform data ROM 2121, and is added to the current sample data to obtain the next sample data, thereby sequentially forming waveform data.

If the DPCM method is adopted, when a waveform such as a voice or a musical tone which generally has a small difference between adjacent samples is to be quantized, quantization can be performed by a smaller number of bits as compared to the normal PCM method.

The operation of the above-mentioned DPCM method will be described below with reference to the operation flow chart shown in Fig. 14. Variables in the flow chart are DPCM data in Table 1 shown in Fig. 12, which data are stored in the corresponding tone generation area (Fig. 10) on the RAM 2061 (Fig. 2).

Of addresses on the control data/waveform data ROM 2121 where DPCM differential waveform data are stored, an address where data as an object to be currently processed is stored is as-

sumed to be (A_I, A_F) shown in Fig. 16.

Pitch data (P_I, P_F) is added to the present address (A_I, A_F) (S_{1101}).

It is then checked if the integral part A_I of the sum address is changed (S_{1102}). If NO in step S_{1102}, an interpolation data value O corresponding to the decimal part A_F of the address is calculated by arithmetic processing $D \times A_F$ using a difference D at the address A_I in Fig. 16 (S_{1114}). Note that the difference D has already been obtained by the sound source processing at the previous interrupt timing (see steps S_{1106} and S_{1110} to be described later).

The interpolation data value O is added to sample data X_P corresponding to the integral part A_I of the address to obtain a new sample data value O (corresponding to X_Q in Fig. 16) corresponding to the current address (A_I, A_F) (S_{1115}).

Thereafter, the sample data value O is multiplied with an envelope value E (S_{1116}), and the obtained value is added to a value stored in the waveform data buffer B (Fig. 23) in the RAM 2061 (Fig. 2) (S_{1117}).

Thereafter, the control returns to the main flow chart shown in Fig. 6. The control is interrupted in the next sampling period, and the operation flow chart of the sound source processing shown in Fig. 14 is executed again. Thus, pitch data (P_I, P_F) is added to the current address (A_I, A_F) (S_{1101}).

The above-mentioned operations are repeated until the integral part AI of the address is changed.

Before the integral part is changed, the sample data X_P and the difference D are left unchanged, and only the interpolation data O is updated in accordance with the address A_F. Thus, every time the address A_F is updated, new sample data X_Q is obtained.

If the integral part A_I of the present address is changed (S_{1102}) as a result of addition of the current address (A_I, A_F) and the pitch data (P_I, P_F) in step S_{1101}, it is checked if the address A_I has reached or exceeded the end address A_E (S_{1103}).

If NO in step S_{1103}, sample data corresponding to the integral part A_I Of the updated present address is calculated by the following loop processing in steps S_{1104} to S_{1107}. More specifically, a value before the integral part A_I of the present address is changed is stored in a variable "old A_I" (see the column of DPCM in Table 1 shown in Fig. 12). This can be realized by repeating processing in step S_{1106} or S_{1113} (to be described later). The old A_I value is sequentially incremented in S_{1106}, and differential waveform data on the control data/waveform data ROM 2121 (Fig. 2) addressed by the incremented old A_I values are read, out as D in step S_{1107}. The readout data D are sequentially accumulated on sample data X_P in step S_{1105}. When the old A_I value becomes equal to the integral part A_I of the changed current address, the sample data X_P as a value corresponding to the integral part A_I of the changed current address.

When the sample data X_P corresponding to the integral part A_I of the current address is obtained in this manner, YES is determined in step S_{1104}, and the control starts the arithmetic processing of the interpolation value (S_{1114}) described above.

The above-mentioned sound source processing is repeated at the respective interrupt timings, and when the judgment in step S_{1103} is changed to YES, the control enters the next loop processing.

An address value (A_I-A_E) exceeding the end address A_E is added to the loop address A_L, and the obtained address is defined as an integral part A_I of a new current address (S_{1108}).

An operation for accumulating the difference D several times depending on an advance in address from the loop address A_L is repeated to calculate sample data X_P corresponding to the integral part A_I of the new current address. More specifically, sample data X_P is initially set as the value of sample data X_{PL} (see the column of DPCM in Table 1 shown in Fig. 12) at the current loop address A_L, and the old A_I is set as the value of the loop address A_L (S_{1109}). The following processing operations in steps S_{1110} to S_{1113} are repeated. More specifically, the old A_I value is sequentially incremented in step S_{1113}, and differential waveform data on the control data/waveform data ROM 2121 designated by the incremented old A_I values are read out as data D. The data D are sequentially accumulated on the sample data X_P in step S_{1112}. When the old A_I value becomes equal to the integral part A_I of the new current address, the sample data X_P has a value corresponding to the integral part A_I of the new current address after loop processing.

When the sample data X_P corresponding to the integral part A_I of the new current address is obtained in this manner, YES is determined in step S_{1111}, and the control enters the above-mentioned arithmetic processing of the interpolation value (S_{1114}).

As described above, waveform data by the DPCM method for one tone generation channel is generated.

Sound Source Processing Based on FM Method

The sound source processing based on the FM method will be described below.

In the FM method, hardware or software elements having the same contents, called "operators", are normally used, and are connected based on connection rules, called algorithms, thereby generating musical tones. In the first embodiment, the FM method is realized by a software

program.

The operation of one embodiment executed when the sound source processing is performed using two operators will be described below with reference to the operation flow chart shown in Fig. 17. The algorithm of the processing is shown in Fig. 18. Variables in the flow chart are FM data in Table 1 shown in Fig. 12, which data are stored in the corresponding tone generation channel area (Fig. 10) on the RAM 2061 (Fig. 2).

First, processing of an operator 2 (OP2) as a modulator is performed. In pitch processing (processing for accumulating pitch data for determining an incremental width of an address for reading out waveform data stored in the ROM 2121), since no interpolation is performed unlike in the PCM method, an address consists of only an integral address A_2. More specifically, modulation waveform data are stored in the control data/waveform data ROM 2121 (Fig. 2) at sufficiently fine incremental widths.

Pitch data P_2 is added to the current address A_2 (S_{1301}).

A feedback output F_{O2} is added to the address A_2 as a modulation input to obtain a new address A_{M2} (S_{1302}). The feedback output F_{O2} has already been obtained upon execution of processing in step S_{1305} (to described later) at the immediately preceding interrupt timing.

The value of a sine wave corresponding to the address A_{M2} (phase) is calculated. In practice sine wave data are stored in the control data/wave from data ROM 2121, and are obtained by addressing the ROM 2121 by the address A_{M2} to read out the corresponding data (S_{1303}).

Subsequently the sine wave data is multiplied with an envelope value E_2 to obtain an output O_2 - (S_{1304}).

Thereafter, the output O_2 is multiplied with a feedback level F_{L2} to obtain a feedback output F_{O2} (S_{1305}). In the first embodiment, this output F_{O2} serves as an input to the operator 2 (OP2) at the next interrupt timing.

The output O_2 is multiplied with a modulation level M_{L2} to obtain a modulation output M_{O2} - (S_{1306}). The modulation output M_{O2} serves as a modulation input to an operator 1 (OP1).

The control then enters processing of the operator 1 (OP1). This processing is substantially the same as that of the operator 2 (OP2) described above, except that there is no modulation input based on the feedback output.

The present address A_1 of the operator 1 (OP1) is added to pitch data P_1 (S_{1307}), and the sum is added to the above-mentioned modulation output M_{O2} to obtain a new address A_{M1} (S_{1308}).

The value of sine wave data corresponding to this address A_{M1} (phase) is read out from the control data/waveform data ROM 2121 (S_{1309}), and is multiplied with an envelope value E_1 to obtain a musical tone waveform output O_1 (S_{1310}).

This output O_1 is added to a value held in the buffer B (Fig. 23) in the RAM 2061 (S_{1311}), thus completing the FM processing for one tone generation channel.

Sound Source Processing Based on TM (Triangular Wave Modulation) Method (Part 1)

The sound source processing based on the TM method will be described below.

The principle of the TM method will be described below.

The FM method described above is based on the following formula:

$$e = A \cdot \sin\{\omega_c t + I(t) \cdot \sin\omega_m t\}$$

where $\omega_c t$ is the carrier wave phase angle (carrier signal), $\sin\omega_m t$ is the modulation wave phase angle (modulation signal), and I(t) is the modulation index.

In contrast to this, a phase modulation method called the TM method in the first embodiment is based on the following formula:

$$e = A \cdot f_T\{f_c(t) + I(t) \cdot \sin\omega_m t\}$$

where $f_T(t)$ is the triangular wave function, and is defined by the following functions in units of phase angle regions (where ω is the input):

$$f_T(\omega) = 2/\pi \cdot \omega$$
..(region : $0 \leqq \omega \leqq \pi/2$)
$$f_T(\omega) = -1 + 2/\pi(3\pi/2 - \omega)$$
..(region : $\pi/2 \leqq \omega \leqq 3\pi/2$)
$$f_T(\omega) = -1 + 2/\pi(\omega - 3\pi/2)$$
..(region : $3\pi/2 \leqq \omega \leqq 2\pi$)

f_c is called a modified sine wave, and is the carrier signal generation function obtained by accessing by the carrier phase angle $\omega_c t$ the control data/waveform data ROM 2121 (Fig. 2) for storing different sine waveform data in units of phase angle regions. f_c of each phase angle region is defined as follows:

$$f_c(t) = \pi/2 \sin\omega_c t$$
..(region : $0 \leqq \omega t \leqq \pi/2$)
$$f_c(t) = \pi - \pi/2 \sin\omega_c t$$
..(region : $\pi \leqq \omega t \leqq 3\pi/2$)
$$f_c(t) = 2\pi + \pi/2 \sin\omega_c t$$
..(region : $3\pi/2 \leqq \omega_c t \leqq 2\pi$)

(where n is an integer)

In the TM method, the above-mentioned trian-

gular wave function is modulated by a sum signal obtained by adding a carrier signal generated by the above-mentioned function $f_c(t)$ to the modulation signal $\sin\omega_m(t)$ at a ratio indicated by the modulation index I(t). In this manner, when the value of the modulation index I(t) is 0, a sine wave can be generated, and as the value I(t) is increased, a very deeply modulated waveform can always be generated. Various other may be used in place of the modulation signal $\sin\omega_m(t)$, and as will be described later, the same operator output in the previous arithmetic processing may be fed back at a predetermined feedback level, or an output from another operator may be input.

The sound source processing based on the TM method according to the abovementioned principle will be described below with reference to the operation flow chart shown in Fig. 19. The sound source processing is also performed using two operators like in the FM method shown in Figs. 17 and 18, and the algorithm of the processing is shown in Fig. 20. Variables in the flow chart are TM format data in Table 1 shown in Fig. 12, which data are stored in the corresponding tone generation channel area (Fig. 10) on the RAM 2061 (Fig. 2).

First, processing of an operator 2 (OP2) as a modulator is performed. In pitch processing, since no interpolation is performed unlike in the PCM method, an address consists of only an integral address A_2.

The present address A_2 is added to pitch data P_2 (S_{1401}).

Modified sine wave data corresponding to the address A_2 (phase) is read out from the control data/waveform waveform data ROM 2121 (Fig. 2) by the modified sine conversion f_c, and is output as a carrier signal O_2 (S_{1402}).

Subsequently, the carrier signal O_2 is added to a feedback output F_{O2} (S_{1406}) as a modulation signal, and the sum signal is output as a new address O_2 (S_{1403}).The feedback output F_{O2} has already been obtained upon execution of processing in step S_{1406} (to be described later) at the immediately preceding interrupt timing.

The value of a triangular wave corresponding to the carrier signal O_2 is calculated. In practice, the above-mentioned triangular wave data are stored in the control data/waveform data ROM 2121 (Fig. 2), and are obtained by addressing the ROM 2121 by the address O_2 to read out the corresponding triangular wave data (S_{1404}).

Subsequently, the triangular wave data is multiplied with an envelope value E_2 to obtain an output O_2 (S_{1405}).

Thereafter, the output O_2 is multiplied with a feedback level F_{L2} to obtain a feedback output F_{O2} (S_{1407}). In the first embodiment, the output F_{O2} serves as an input to the operator 2 (OP2) at the next interrupt timing.

The output O_2 is multiplied with a modulation level M_{L2} to obtain a modulation output M_{O2} - (S_{1407}). The modulation output M_{O2} serves as a modulation input to an operator 1 (OP1).

The control then enters processing of the operator 1 (OP1). This processing is substantially the same as that of the operator 2 (OP2) described above, except that there is no modulation input based on the feedback output.

The present address A_1 of the operator 1 is added to pitch data P_1 (S1408), and the sum is subjected to the above-mentioned modified sine conversion to obtain a carrier signal O_1 (S1409).

The carrier signal O_1 is added to the above-mentioned modulation output M_{O2} to obtain a new value O_1 (S1410), and the value O_1 is subjected to triangular wave conversion (S1411). The converted is multiplied with an envelope value E_1 to obtain a musical tone waveform output O_1 (S1412).

The output O_1 is added to a value held in the buffer B (Fig. 23) in the RAM 2061 (Fig. 2) (S1413), thus completing the TM processing for one tone generation channel.

The sound source processing operations based on four methods, i.e., the PCM, DPCM, FM, and TM methods have been described. Of these methods, the FM and TM methods are modulation methods, and, in the above examples, two-operator processing operations are executed based on the algorithms shown in Figs. 18 and 20. However, in sound source processing in an actual performance, more operators may be used, and the algorithms may be more complicated.

Summary of Keyboard Key Processing

The operations of keyboard key processing (S_{505}) in the main flow chart shown in Fig. 6 when an actual electronic musical instrument is played will be described in detail below.

In the above-described sound source processing, data in units of sound source methods (Fig. 12) are set in the corresponding tone generation channel areas (Fig. 10) on the RAM 2061 (Fig. 2) by the function keys 1031 (Fig. 1). The function keys 1031 are connected to, e.g., an operation panel of the electronic musical instrument via the input port 2101 (Fig. 2).

In the first embodiment, split points based key codes and velocities, and two tone colors are designated in advance, thus allowing characteristic assignment of tone colors to the tone generation channels.

The split points and the tone colors are designated, as shown in Fig. 21 or 27.

Fig. 21 shows an arrangement of some function keys 1031 (Fig. 1). A keyboard split point

designation switch 15011 comprises a slide switch which has a click feeling, and can designated a split point based on key codes of ON keys in units of keyboard key. When two tone colors, e.g., "piano" and "guitar" are designated as X and Y tone colors by tone color switches 15021, the X tone color is designated for a bass tone range, and the Y tone color is designated for a high tone range to have the above-mentioned split point as a boundary. In this case, a tone color designated first is set as the X tone color, and for example, a red LED is turned on. A tone color designated next is set as the Y tone color, and a green LED is turned on. The LEDs correspond to the display unit 1091 (Fig. 1).

A split point based on velocities is designated by a velocity split point designation switch 15031 shown in Fig. 27. For example, when the switch 15031 is set at velocity = 60, an X tone color is designated for ON events having a velocity of 60 or less, and a Y tone color is designated for ON events having a velocity faster than 60. In this case, the X and Y tone colors are designated by tone color switches 20021 (Fig. 27) in same manner as in Fig. 21 (the case of a split point based on key codes).

The arrangement shown in Fig. 21 or 27 can constitute an independent embodiment. However, an embodiment having both these functions may be realized. In order to allow the above-mentioned tone color setting operations, the control data/waveform data ROM 2121 (Fig. 2) stores various tone color parameters in data formats shown in Fig. 22. More specifically, tone color parameters for the four sound source methods, i.e., the PCM, DPCM, FM, and TM methods are stored in units of instruments corresponding to the tone color switches 15021 of "piano" as the tone color No. 1, "guitar" as the tone color No. 2, and the like shown in Fig. 21. The tone color parameters for the respective sound source methods are stored in the data formats in units of sound source methods shown in Fig. 12. On the other hand, the buffer B for accumulating waveform data for eight tone generation channels, and the tone color No. registers for holding the tone color Nos. of the X and Y tone colors are allocated on the RAM 2061 (Fig. 2).

Tone color parameters in units of sound source methods, which have the data formats shown in Fig. 22, are set in the tone generation channel areas (Fig. 10) for the eight channels of the RAM 2061, and sound source processing is executed based on these parameters. Processing operations for assigning tone color parameters to the tone generation channels in accordance with ON events on the basis of the split point and the two, i.e., X and Y tone colors designated by the function keys shown in Fig. 21 or 27 will be described below in turn.

Embodiment A of Keyboard key Processing

The embodiment A of keyboard key processing will be described below.

The embodiment A is for an embodiment having the arrangement shown in Fig. 21 as some function keys 1031 shown in Fig. 1. Based on an operation of the keyboard split point designation switch 15011 shown in Fig. 21 by a player, key codes of ON keys are split into two groups at the split point. Then, musical tone signals in two, i.e., X and Y tone colors designated upon operation of the tone color switches 15021 (Fig. 21) by the player are generated. Furthermore, one of the four sound source methods is selected in accordance with the magnitude of a velocity (corresponding to an ON key speed) obtained upon an ON event of a key on the keyboard 1021 (Fig. 1). Tone color generation is performed on the basis of the tone colors and the sound source method determined in this manner.

In the embodiment A, as shown in Fig. 32, musical tone signals in the X tone color are generated using the first to fourth tone generation channels (ch1 to ch4), and musical tone signals in the Y tone color are generated using the fifth to eighth tone generation channels (ch5 to ch8).

Note that operations of the keyboard split point designation switch 15011 and the tone color switches 15021 shown in Fig. 21 by the player are detected in the function key scanning processing in step S_{502} in the main flow chart of Fig. 6, and in the function key processing in step S503 in Fig. 6, key codes corresponding to the operation states are held in registers (not shown) on the RAM 2061. In addition, the X and Y tone colors are held in the X and Y tone color No. registers (Fig. 23) in the RAM 2061.

Fig. 25 is an operation flow chart of the embodiment A of the keyboard key processing in step S_{505} in the main flow chart shown in Fig. 6.

It is checked if a key code of a key determined as an "ON key" in step S_{504} in the main flow chart shown in Fig. 6 is equal to or smaller than that at the split point designated in advance (S_{1801}).

If YES in step S_{1801}, tone color parameters of the X tone color designated beforehand by the player are set in one of the first to fourth tone generation channels (Fig. 32) by the following processing operations in steps S_{1802} to S_{1805} and S_{1810} to S_{1813}. It is checked if the first to fourth tone generation channels include an empty channel (S_{1802}).

If it is determined that there is no empty channel, and NO is determined in step S_{1802}, no assignment is performed.

If it is determined that there is an empty channel, and YES in step S_{1802}, tone color parameters for the X tone color, and corresponding to one of the PCM, DPCM, TM, and FM methods are set in the empty channel in accordance with the velocity value as follows.

It is checked if the velocity value of a key determined as an "ON key" in step S_{504} in the main flow chart in Fig. 6 is equal to or smaller than 63 (almost corresponding to mezzo piano mp) (S1803).

If YES in step $S1_{803}$, e.i., if it is determined that the velocity value is equal to or smaller than 63, it is then checked if the value is equal to or smaller than 31 (almost corresponding to piano p) (S_{1805}).

If YES in step S_{1805}, e.i., if it is determined that the velocity value V falls within a range of $0 \leqq V \leqq 31$, the tone color parameters for the X tone color are set in the FM format shown in Fig. 12 in one tone generation channel area (empty channel area) of the first to fourth channels (Fig. 2) to which the ON key is assigned on the RAM 2061. More specifically, sound source method No. data S representing the FM method is set in the first area of the corresponding tone generation channel area (see the column of FM in Fig. 12). Then, the tone color parameters corresponding to the tone color of the tone color No. presently stored in the X tone color No. register (Fig. 23) on the RAM 2061 are read out from a data architecture portion shown in Fig. 22 of the control data/waveform data ROM 2121, and are set in the second and subsequent areas of the tone generation channel area (S_{1813}).

If YES in step S_{1805}, e.i., if it is determined that the velocity value falls within a range of $31 \leqq V$ 63, tone color parameters for the X tone color are set in the TM format shown in Fig. 12 in the tone generation channel area on the RAM 2061 to which the ON key is assigned (S_{1812}). In this case, the parameters set in the same manner as in step S_{1813}.

If NO in step S_{1803}, it is then checked if the velocity value is equal to or smaller than 95 (almost corresponding to piano p) (S_{1804}).

If YES in step S_{1804}, i.e., if it is determined that the velocity value V falls within a range of $63 \leqq V \leqq 95$, tone color parameters for the X tone color are set in the DPCM format shown in Fig. 12 in the tone generation channel area on the RAM 2061 to which the ON key is assigned (S_{1811}). In this case, the parameters set in the same manner as in step S_{1813}.

If NO in step S_{1804}, i.e., if it is determined that the velocity value V falls within a range of $95 \leqq V \leqq 127$, tone color parameters for the X tone color are set in the PCM format shown in Fig. 12 in the tone generation channel area on the RAM 2061 to which the ON key is assigned (S_{1810}). In this case, the parameters are set in the same manner as in step S_{1813}.

On the other hand, if NO in first step S_{1801}, tone color parameters for the Y tone color designated in advance by the player are set in one of the fifth to eighth tone generation channels (Fig. 32) by the following processing in steps S_{1806} to S_{1809} and S_{1814} to S_{1817}.

It is checked if the fifth to eighth tone generation channels include an empty channel (S_{1806}).

It it is determined that there is no empty channel, and NO is determined in step S_{1806}, no assignment is performed.

If it is determined that there is an empty channel, and YES is determined in step S_{1806}, tone color parameters for the Y tone color, and corresponding to one of the PCM, DPCM, TM, and FM methods are set in the empty channel in accordance with the velocity value as follows.

First, it is checked if the velocity value of an ON key is equal to or smaller than 63 (S_{1807}).

If YES in step S_{1807}, i.e., if it is determined that the velocity value is equal to or smaller than 63, it is then checked if the value is equal to or smaller than 31 (S_{1808}).

If YES in step S_{1808}, i.e., if it is determined that the velocity value V falls within a range of $0 \leqq V \leqq 31$, tone color parameters for the Y tone color are set in the FM format in Fig. 12 in one of the fifth to eighth channels to which the ON key is assigned. More specifically, sound source method No. data S representing the FM method is set in the first area of the corresponding tone generation channel area (see the column of FM in Fig. 12). Then, the tone color parameters corresponding to the tone color of the tone color No. presently stored in the Y tone color No. register (Fig. 23) on the RAM 2061 are read out from a data architecture portion shown in Fig. 22 of the control data/waveform data ROM 2121, and are set in the second and subsequent areas of the tone generation channel area (S_{1814}).

If YES in step S_{1808}, i.e., if it is determined that the velocity value falls within a range of $31 \leqq V \leqq 63$, tone color parameters for the Y tone color are set in the TM format shown in Fig. 12 in the tone generation channel area on the RAM 2061 to which the ON key is assigned (S_{1815}). In this case, the parameters are set in the same manner as in step S_{1814}.

If NO in step S_{1807}, it is checked if the velocity value is equal to or smaller than 95 (S_{1809}).

If YES in step S_{1809}, i.e., if it is determined that the velocity value V falls within a range of $63 < V \leqq 95$, tone color parameters for the Y tone color are set in the DPCM format shown in Fig. 12 in the tone generation channel area on the RAM 2061 to which the ON key is assigned (S_{1816}). In this case,

the parameters are set in the same manner as in step S_{1814}.

If NO in step S_{1816}, i.e., if it is determined that the velocity value V falls within a range of $95 < V \leqq 127$, tone color parameters for the Y tone color are set in the PCM format shown in Fig. 12 in the tone generation channel area on the RAM 2061 to which the ON key is assigned (S_{1817}). In this case, the parameters are set in the same manner as in step S_{1814}.

As described above, one of the X and Y tone colors is selected in accordance with whether the key code is lower or higher than the split point, and one of the four sound source methods is selected in accordance with the magnitude of an ON key velocity, thus generating musical tones.

Embodiment B of Keyboard Processing

The embodiment B of the keyboard key processing will be described below.

In the embodiment A described above, as shown in Fig. 32, the tone generation channels to which the X and Y tone colors are assigned are fixed as the first to fourth tone generation channels and the fifth to eighth tone generation channels, respectively. In the embodiment B, channels to which each tone color is assigned are not fixed, and the X and Y tone colors are sequentially assigned to empty channels, as shown in Fig. 33.

Fig. 26 is an operation flow chart of the embodiment B of the keyboard key processing in step S_{505} in the main flow chart shown in Fig. 6. As shown in Fig. 26, it is checked if the first to eighth channels include an empty channel (S_{1901}). If there is an empty channel, tone color assignment is performed. The processing operations in steps S_{1902} to S_{1916} the same as those in steps S_{1801}, S_{1803} to S_{1805}, and S_{1806} to S_{1817} in the embodiment A.

According to the embodiment B, flexible tone color assignment to the tone generation channels can be performed.

Embodiment C of Keyboard Key Processing

The embodiment C of the keyboard key processing will be described below.

The embodiment C corresponds to a case wherein processing for a key code and processing for a velocity in the embodiment A are replaced.

More specifically, the embodiment C is for an embodiment having an arrangement shown in Fig. 27 as some function keys 1031 shown in Fig. 1, and velocities of ON keys are split into two groups at the split point upon operation of the velocity split point designation switch 20011 (Fig. 27) by the player. Then, musical tone signals are generated in the two, i.e., X and Y tone colors designated upon operation of the tone color switches 20021 (Fig. 27) by the player. In this case one of the four sound source methods is selected in accordance with a key code value of an ON key on the keyboard 1021 (Fig. 1) by the player. Tone color generation is performed in accordance with the tone colors and the sound source method determined in this manner. The X and Y tone colors are assigned to the tone generation channels, as shown in Fig. 32, in the same manner as in the embodiment A.

Fig. 28 is an operation flow chart of the embodiment C of the keyboard key processing in step S_{505} in the main flow chart of Fig. 6.

It is checked if the velocity of a key determined as an "ON key" in step S_{504} in the main flow chart in Fig. 6 is equal to or smaller than the velocity at the split point designated in advance by the player (S_{2101}).

If YES in step S_{2101}, tone color parameters for the X tone color designated in advance by the player are set in one of the first to fourth tone generation channels (Fig. 32) by the following processing in steps S_{2102} to S_{2105} and S_{2110} to S_{2113}.

It is checked if the first to fourth tone generation channels include an empty channel (S_{2102}).

If it is determined that there is no empty channel, and NO is determined in step S_{2102}, no assignment is performed.

If it is determined that there is an empty channel, and YES is determined in step S_{2102}, tone color parameters for the X tone color, and corresponding to one of the PCM, DPCM, TM, and FM methods are set in the empty channel in accordance with the key code value as follows.

It is checked if the key code value of a key determined as an "ON key" in step S_{504} in the main flow chart in Fig. 6 is equal to or larger than 32 (S_{2103}).

If YES in step S_{2103}, i.e., if it is determined that the key code value is equal to or larger than 32, it is then checked if the value is equal to or larger than 48 (S_{2105}).

If YES in step S_{2105}, i.e., if it is determined that the key code value K falls within a range of $48 \leqq K \leqq 63$ (63 = maximum value), tone color parameters for the X tone color are set in the FM format shown in Fig. 12 in one of the first to fourth channels area on the RAM 2061 to which the ON key is assigned (Fig. 2). In this case, the parameters are set in the same manner as in step S_{1813} in the embodiment A.

If YES in step S_{2105}, i.e., if the key code value falls within a range of $32 \leqq V < 48$, tone color parameters for the X tone color are set in the TM format shown in Fig. 12 in the tone generation channel area on the RAM 2061 to which the ON

key is assigned (S_{2112}). In this case, the parameters are set in the same manner as in step S_{1813} in the embodiment A.

If NO in step S_{2103}, it is checked if the key code value is equal to or larger than 16 (S_{2104}).

If YES in step S_{2104}, i.e., if it is determined that the key code value K falls within a range of $16 \leqq K \leqq 32$, tone color parameters for the X tone color are set in the DPCM format shown in Fig. 12 in the tone generation channel area on the RAM 2061 to which the ON key is assigned (S_{2111}). In this case, the parameters are set in the same manner as in step S1813 in the embodiment A.

Furthermore, if NO in step S_{2104}, i.e., if it is determined that the key code value K falls within a range of $0 \leqq V < 16$, tone color parameters for the X tone color are set in the PCM format shown in Fig. 12 in the tone generation channel area on the RAM 2061 to which the ON key is assigned (S_{2110}). In this case, the parameters are set in the same manner as in step S_{1813} in the embodiment A.

If NO in first step S_{2101}, tone color parameters for the Y tone color designated in advance by the player are set in one of the fifth to eighth tone generation channels (Fig. 32) by the following processing in steps S_{2106} to S_{2109} and S_{2114} to S_{2117}.

It is checked if the fifth to eighth tone generation channels include an empty channel (S_{2106}).

If it is determined that there is no empty channel, and NO is determined in step S_{2106}, no assignment is performed.

If there is an empty channel, and YES is determined in step S_{2106}, it is checked in the processing in steps S_{2107} to S_{2109} having the same judgment conditions as those in steps S_{2103} to S_{2105} if the key code value falls within a range of $48 \leqq K \leqq 63$, $32 \leqq K < 48$, $16 \leqq K < 32$, or $0 \leqq K < 16$. Thus, in steps S_{2114} to S_{2117}, tone color parameters for the Y color and corresponding to one of the FM, TM, DPCM, and PCM methods are set in an empty channel.

Embodiment D of Keyboard Key Processing

Furthermore, the embodiment D of the keyboard key processing will be described below.

In the embodiment C, as shown in Fig. 32, the tone generation channels to which the X and Y tone colors are assigned are fixed as the first to fourth tone generation channels and the fifth to eighth tone generation channels, respectively. In the embodiment D, channels to which each tone color is assigned are not fixed, and the X and Y tone colors are sequentially assigned to empty channels, as shown in Fig. 33 like in the embodiment B.

Fig. 29 is an operation flow chart of the embodiment D of the keyboard key processing in step S_{505} in the main flow chart shown in Fig. 6. As shown in Fig. 29, it is checked if the first to eighth channels include an empty channel (S_{2201}). If there is empty channel, tone color assignment is performed. The processing operations in steps S_{2202} to S_{2216} are the same as those in steps S_{2201}, S_{2203} to S_{2205}, and S_{2206} to S_{2217} in the embodiment C shown in Fig. 28.

Demonstration Performance Processing

The operations of the demonstration performance processing (S_{506}) in the main flow chart shown in Fig. 6 when a demonstration performance (automatic performance) is executed in some electronic musical instruments in addition to the keyboard key processing described above, will be described in detail below.

In the first embodiment, different tone colors and sound source methods can be assigned to the tone generation channels in accordance with whether the ON key plays a melody or accompaniment part.

Fig. 30 is an operation flow chart of an embodiment A of the demonstration performance processing in step S_{506} in the main flow chart shown in Fig. 6. In the embodiment A, X and Y tone colors are assigned to the tone generation channels, as shown in Fig. 32, in the same manner as the embodiment A or C of the keyboard key processing.

It is checked whether or not an ON key designated by automatic performance data read out from the control data/waveform data ROM 2121 (Fig. 2) plays a melody (or accompaniment part) (S_{2301}).

If YES in step S_{2301}, i.e., if it is determined that the key plays the melody part, it is checked if the first to fourth tone generation channels include an empty channel (S_{2302}).

If there is no empty channel, and NO is determined in step S_{2302}, no assignment is performed.

If there is an empty channel, and YES is determined in step S_{2302}, tone color parameters for the X tone color are set in the FM format shown in Fig. 12 in one tone generation channel area of the first to fourth channels on the RAM 2061 (Fig. 2) to which the ON key is assigned. More specifically, sound source method No. data S representing the FM method is set in the first area of the corresponding tone generation channel area (see the column of FM in Fig. 12). Then, the tone color parameters corresponding to the tone color of the tone color No. presently stored in the X tone color No. register (Fig. 23) on the RAM 2061 are read out from a data architecture portion shown in Fig.

22 of the control data/waveform data ROM 2121, and are set in the second and subsequent areas of the tone generation channel area (S_{2303}).

If NO in step S_{2301}, it is checked if the fifth to eighth tone generation channels include an empty channel (S_{2304}).

If there is no empty channel, and NO is determined in step S_{2304}, no assignment is performed.

If there is an empty channel, and YES is determined in step S_{2304}, tone color parameters for the Y tone color are set in the DPCM format shown in Fig. 12 in one tone generation channel area of the fifth to eighth channels on the RAM 2061 (Fig. 2) to which the ON key is assigned. More specifically, sound source method No. data S representing the DPCM method is set in the first area of the corresponding tone generation channel area (see the column of DPCM in Fig. 12). Then, the tone color parameters corresponding to the tone color of the tone color No. presently stored in the X tone color No. register (Fig. 23) on the RAM 2061 are read out from a data architecture portion shown in Fig. 22 of the control data/waveform data ROM 2121, and are set in the second and subsequent areas of the tone generation channel area (S_{2305}).

Fig. 31 is an operation flow chart of an embodiment B of demonstration performance processing in step S_{506} in the main flow chart of Fig. 6. In the embodiment B, channels to which each tone color is assigned are not fixed, and the X and Y tone colors are sequentially assigned to empty channels, as shown in Fig. 33 like in the embodiment B or D of the keyboard key processing.

In Fig. 31, it is checked if the first to eighth channels include an empty channel (S_{2401}). If there is an empty channel, tone color assignment is performed. The processing operations in steps S_{2402} to S_{2404} arethe same as those in steps S_{2302} to S_{2304} in the embodiment A of the demonstration performance processing shown in Fig. 30.

Other Embodiments

In the embodiments A to D of the keyboard key processing described above, two tone colors are switched to have a split point for key code or velocity values as a boundary, and sound source methods are switched in units of tone colors in accordance with the velocity or key code values. Contrary to this, the sound source methods may be switched to have a split point as a boundary, and tone colors may be switched in units of sound source methods in accordance with, e.g., velocity values.

The number of split points is not limited to one, and a plurality of tone colors or sound source methods may be switched in regions having two or more split points as boundaries.

Furthermore, performance data associated with the split point is not limited to a key code or a velocity.

On the other hand, in the embodiments A and B of the demonstration performance processing, different tone colors and sound source methods can be assigned to tone generation channels in accordance with a melody or accompaniment part in a demonstration performance (automatic performance) mode. However, the present invention is not limited to this. For example, tone colors and sound source methods may be switched in accordance with whether a player plays a melody or accompaniment part.

In the embodiments A and B of the demonstration performance processing, an assignment state of tone generation is changed in a permanent combination of tone colors and sound source methods in accordance with a melody or accompaniment part. However, like in the keyboard key processing, only tone colors or sound source methods may be changed, and the kind of parameters may be desirably selected.

Summary of the Second Embodiment

The summary of this embodiment will be described below.

Fig. 34 is a block diagram showing the overall arrangement of this embodiment. In Fig. 34, components other than an external memory 1162 are constituted in one chip. Of these components, two, i.e., master and slave CPUs (central processing units) exchange data to share sound source processing for generating musical tones.

In, e.g., a 16-channel polyphonic system, 8 channels are processed by a master CPU 1012, and the remaining 8 channels are processed by a slave CPU 1022.

The sound source processing is executed in a software manner, and sound source methods such as PCM (Pulse Code Moduration) and DPCM (Differential PCM) methods, and sound source methods based on modulation methods such as FM and phase modulation methods are assigned in units of tone generation channels.

A sound source method is automatically designated for tone colors of specific instruments, e.g., a trumpet, a tuba, and the like. For tone colors of other instruments, a sound source method can be selected by a selection switch, and/or can be automatically selected in accordance with a performance tone range, a performance strength such as a key touch, and the like.

In addition, different sound source methods can be assigned to two channels for one ON event of a key. That is, for example, the PCM method

can be assigned to an attack portion, and the FM method can be assigned to a sustain portion.

Furthermore, in, e.g., the FM method, when software processing is executed by a versatile CPU according to a sound source processing algorithm, it requires too much time. However, this embodiment can also solve this problem.

Arrangement of The Second Embodiment

The second embodiment will be described below with reference to the accompanying drawings.

In Fig. 34, the external memory 1162 stores musical tone control parameters such as target values of envelope values, a musical tone waveform in the PCM (pulse code modulation) method, a musical tone differential waveform in the DPCM (differential PCM) method, and the like.

The master CPU (to be abbreviated to as an MCPU hereinafter) 1012 and the slave CPU (to be abbreviated to as an SCPU hereinafter) 1022 access the data on the external memory 1162 to execute sound source processing while sharing processing operations. Since these CPUs 1012 and 1022 commonly use waveform data of the external memory 1162, a contention may occur when data is loaded from the external memory 1162. In order to prevent this contention, the MCPU 1013 and the SCPU 1022 outputan address signal for accessing the external memory, and external memory control data from output terminals 1112 and 1122 of an access address contention prevention circuit 1052 via an external memory access address latch unit 1032 for the MCPU, and an external memory access address latch unit 1042 for the SCPU. Thus, a contention between addresses from the MCPU 1012 and the SCPU 1022 can be prevented.

Data read out from the external memory 1162 on the basis of the designated address is input from an external memory data input terminal 1152 to an external memory selector 1062. The external memory selector 1062 separates the readout data into data to be input to the MCPU 1012 via a data bus MD and data to be input to the SCPU 1022 via a data bus SD on the basis of a control signal from the address contention prevention circuit 1052, and inputs the separated data to the MCPU 1012 and the SCPU 1022. Thus, a contention between readout data can also be prevented.

After the MCPU 1012 and the SCPU 1022 perform corresponding sound source processing operations of the input data by software, musical tone data of all the tone generation channels are accumulated, and a left-channel analog output and a right-channel analog output are then output from a left output terminal 1132 of a left D/A converter unit 1072 and a right output terminal 1142 of a right D/A converter unit 1082, respectively.

Fig. 35 is a block diagram showing an internal arrangement of the MCPU 1012.

In Fig. 35, a control ROM 2012 stores a musical tone control program (to be described later), and sequentially outputs program words (commands) addressed by a ROM address controller 2052 via a ROM address decoder 2022. This embodiment employs a next address method. More specifically, the word length of each program word is, e.g., 28 bits, and a portion of a program word is input to the ROM address controller 2052 as a lower bit portion (intra-page address) of an address to be read out next. Note that the SCPU 1012 may comprise a conventional program counter type CPU insted of control ROM 2012.

A command analyzer 2072 analyzes operation codes of commands output from the control ROM 2012, and sends control signals to the respective units of the circuit so as to execute designated operations.

When an operand of a command from the control ROM 2012 designates a register, the RAM address controller 2042 designates an address of a corresponding internal register of a RAM 2062. The RAM 206 stores various musical tone control data (to be described later with reference to Figs. 49 and 50) for eight tone generation channels, and includes various buffers (to be described later) or the like. The RAM 2062 is used in sound source processing (to be described later).

When a command from the control ROM 2012 is an arithmetic command, an ALU unit 2082 and a multiplier 2092 respectively execute an addition/subtraction, and a multiplication on the basis of an instruction from the command analyzer 2072.

On the basis of an internal hardware timer (not shown), an interrupt controller 2032 supplies a reset cancel signal $\overline{A}$ to the SCPU 2012 (Fig. 34) and an interrupt signal $\overline{I}$ to the D/A converter units 1072 and 1082 (Fig. 34) at predetermined time intervals.

In addition to the above-mentioned arrangement, the MCPU 1012 shown in Fig. 35 comprises the following interfaces associated with various buses: an interface 2152 for an address bus MA for addressing the external memory 1162 to access it; an interface 2162 for the data bus MD for exchanging the accessed data with the MCPU 1012 via the external memory selector 1062; an interface 2122 for a bus Ma for addressing the internal RAM of the SCPU 1022 so as to execute data exchange with the SCPU 1022; an interface 2132 for a data bus D_{OUT} used by the MCPU 1012 to write data in the SCPU 1022; an interface 2142 for a data bus D_{IN} used by the MCPU 1012 to read data from the SCPU 1022; an interface 2172 for a D/A data transfer bus for transferring final output waveforms to the left and right D/A converter units 1072 and

1082; and input and output ports 2102 and 2112 for exchanging data with an external switch unit or a keyboard unit (Figs. 45, and 46).

Fig. 36 shows the internal arrangement of the SCPU 1022.

Since the SCPU 1022 executes sound source processing upon reception of a processing start signal from the MCPU 1012, it does not comprise an interrupt controller corresponding to the controller 2032 (Fig. 35), I/O ports, corresponding to the ports 2102 and 2112 (Fig. 35) for exchanging data with an external circuit, and an interface, corresponding to the interface 2172 (Fig. 35) for outputting musical tone signals to the left and right D/A converter units 1072 and 1082. Other circuits 3012, 3022, and 3042 to 3092 have the same functions as those of the circuits 2012, 2022, and 2042 to 2092 shown in Fig. 35. Interfaces 3032, and 3102 to 3132 are arranged in correspondence with the interface 2122 to 2162 shown in Fig. 35. Note that the internal RAM address of the SCPU 1022 designated by the MCPU 1012 is input to the RAM address controller 3042. The RAM address controller 3042 designates an address of the RAM 3062. Thus, accumulated waveform data for eight tone generation channels generated by the SCPU 1022 and held in the RAM 3062 are output to the MCPU 1012 via the data bus D_{IN}. This will be described later.

In addition to the above-mentioned arrangement, in this embodiment, function keys 8012, keyboard keys 8022, and the like shown in Figs. 45 and 46 are connected to the input port 2102 of the MCPU 1012. Theses portions substantially constitute an instrument operation unit.

The D/A converter unit as one characteristic feature of the present invention will be described below.

Fig. 43 shows the internal arrangement of the left or right D/A converter unit 1027 or 1082 (the two converter units have the same contents) shown in Fig. 34. One sample data of a musical tone generated by sound source processing is input to a latch 6012 via a data bus. When the clock input terminal of the latch 6012 receives a sound source processing end signal from the command analyser 2072 (Fig. 35) of the MCPU 1012, musical tone data for one sample on the data bus is latched by the latch 6012, as shown in Fig. 44.

A time required for the sound source processing changes depending on the sound source processing software program. For this reason, a timing at which each sound source processing is ended, and musical tone data is latched by the latch 6012 is not fixed. For this reason, as shown in Fig. 42, an output from the latch 6012 cannot be directly input to a D/A converter 6032.

In this embodiment, as shown in Fig. 43, the output from the latch 6012 is latched by a latch 6022 in response to an interrupt signal equal to a sampling clock interval output from the interrupt controller 2032, and is output to the D/A converter 603 at predetermined time intervals.

Since a change in processing time can be absorbed using the two latches 6012 and 6022, no complicated control program for outputting musical tone data to a D/A converter 6032 is required.

Overall Operation of The Second Embodiment

The overall operation of this embodiment will be described below.

In this embodiment, basically, the MCPU 1012 is mainly operated, and repetitively executes a series of processing operations in steps S402 to S410, as shown in the main flow chart of Fig. 37. The sound source processing is performed by interrupt processing. More specifically, the MCPU 1012 and the SCPU 1022 are interrupted at predetermined time intervals, and each CPU executes sound source processing for generating musical tones for eight channels. Upon completion of this processing, musical tone waveforms for 16 channels are added, and are output from the left and right D/A converter units 1072 and 1082. Thereafter, the control returns from the interrupt state to the main flow. Note that the above-mentioned interrupt processing is periodically executed on the basis of the internal hardware timer in the interrupt controller 2032 (Fig. 35). This period is equal to a sampling period when a musical tone is output.

The schematic operation of this embodiment has been described. The operation of this embodiment will be described in detail below with reference to Figs. 37 to 40.

When the interrupt controller 2032 interrupts repetitively executed processing operations in steps S402 to S410 in the main flow chart of Fig. 37, MCPU interrupt processing shown in Fig. 38 and SCPU interrupt processing shown in Fig. 39 are simultaneously started. "Sound source processing" in Figs. 38 and 39 is shown in Fig. 40.

The main flow chart of Fig. 37 shows a processing flow executed by the MCPU 1012 in a state wherein no interrupt signal is supplied from the interrupt controller 2032.

When the power switch is turned on, the system e.g., the contents of the RAM 2062 in the MCPU 1012 are initialized (S401).

The function keys externally connected to the MCPU 1012, e.g., tone color switches, and the like (Fig. 65), are scanned (S402) to fetch respective switch states from the input port 2102 to a key buffer area in the RAM 2062. As a result of scanning, a function key whose state is changed is discriminated, and processing of a corresponding

function is executed (S403). For example, a musical tone number or an envelope number is set, or if optional functions include a rhythm performance function, a rhythm number is set.

Thereafter, states of ON keyboard keys are fetched in the same manner as the function keys (S404), and keys whose states are changed are discriminated, thus executing key assignment processing (S405).

When a demonstration performance key of the function keys 8012 (Figs. 45 and 46) is depressed, demonstration performance data (sequencer data) are sequentially read out from the external memory 1162 to execute, e.g., key assignment processing (S406). When a rhythm start key is depressed, rhythm data are sequentially read out from the external memory 1162 to execute, e.g., key assignment processing (S407).

Thereafter, timer processing is executed (S408). More specifically, time data which is incremented by interrupt timer processing (S412) (to be described later) is compared with time control sequencer data sequentially read out for demonstration performance control or time control rhythm data read out for rhythm performance control, thereby executing time control when a demonstration performance in step S406 or a rhythm performance in step S407 is performed.

In tone generation processing in step S409, pitch envelope processing, and the like are executed. In this processing, an envelope is added to a pitch of a musical tone to be generated, and pitch data is set in a corresponding tone generation channel.

Furthermore, one flow cycle preparation processing is executed (S410). In this processing, processing for changing a state of a tone generation channel assigned with a note number corresponding to an ON event detected in the keyboard key processing in step S405 to an "ON event" state, and processing for changing a state of a tone generation channel assigned with a note number corresponding to an OFF event to a "muting" state, and the like are executed.

The MCPU interrupt processing shown in Fig. 38 will be described below.

When the interrupt controller 2032 of the MCPU 1012 interrupts the MCPU 1012, the processing in the main flow chart shown in Fig. 37 is interrupted, and the MCPU interrupt processing in Fig. 38 is started. In this case, control is made to avoid contents of registers to be subjected to write access in the main flow program in Fig. 37 from being rewritten in the MCPU interrupt processing program. For this reason, the MCPU interrupt processing uses registers different from those used in the main flow program. As a result, register save/restoration processing normally executed at the beginning and end of interrupt processing can be omitted. Thus, transition between the processing of the main flow chart shown in Fig. 37 and the MCPU interrupt processing can be quickly performed.

Subsequently, in the MCPU interrupt processing, sound source processing is started (S411). The sound source processing is shown in Fig. 40.

Simultaneously with the above-mentioned operations, the interrupt controller 2032 of the MCPU 1012 outputs the SCPU reset cancel signal $\overline{A}$ (Fig. 34) to the ROM address controller 3052 of the SCPU 1022, and the SCPU 1022 starts execution of the SCPU interrupt processing (Fig. 39).

Sound source processing (S415) is started in the SCPU interrupt processing almost simultaneously with the source processing (S411) in the MCPU interrupt processing. In this manner, since each of the MCPU 1012 and the SCPU 1022 simultaneously executes sound source processing of eight tone generation channels, the sound source processing for 16 tone generation channels can be executed in a processing time for eight tone generation channels, and a processing speed can be almost doubled (the interrupt processing will be described later with reference to Fig. 41).

In the interrupt timer processing in step S412, the value of time data (not shown) on the RAM 2062 (Fig. 35) is incremented by utilizing the fact that the interrupt processing shown in Fig. 38 is executed for every predetermined sampling period. More specifically, a time elapsed from power-on can be detected based on the value of the time data. The time data obtained in this manner is used in time control in the timer processing in step S408 in the main flow chart shown in Fig. 37.

The MCPU 1012 then waits for an SCPU interrup processing end signal $\overline{B}$ from the SCUP 1022 after interrupt timer processing in step S412 (S413).

Upon completion of the sound source processing in step S415 in Fig. 39, the command analyzer 3072 of the SCPU 1022 supplies an SCPU processing end signal $\overline{B}$ (Fig. 34) to the ROM address controller 2052 of the MCPU 1012. In this manner, YES is determined in step S413 in the MCPU interrupt processing in Fig. 38.

As a result, waveform data generated by the SCPU 1022 are written in the RAM 2062 of the MCPU 1012 via the data bus D_{IN} shown in Fig. 34 (S414). The waveform data are stored in a predetermined buffer area (a buffer $\overline{B}$ to be described later) on the RAM 3062 of the SCPU 1022. The command analyzer 2072 of the MCPU 1012 designates addresses of the buffer area to the RAM address controller 3042, thus reading the waveform data.

In step S414', the contents of the buffer area $\overline{B}$

are latched by the latches 6012 (Fig. 43) of the left and right D/A converter units 1072 and 1082.

The operation of the sound source processing executed in step S411 in the MCPU interrupt processing or in step S415 in the SCPU interrupt processing will be described below with reference to the flow chart of Fig. 40.

A waveform addition area on the RAM 2062 or 3062 is cleared (S416). Then, sound source processing is executed in units of tone generation channels (S417 to S424). After the sound source processing for the eighth channel is completed, waveform data obtained by adding those for eight channels is obtained in the buffer area B. These processing operations will be described in detail later.

Fig. 41 is a schematic flow chart showing the relationship among the processing operations of the flow charts shown in Figs. 37, 38, and 39. As can be seen from Fig. 41, the MCPU 1012 and the SCPU 1022 share the sound source processing.

Given processing A (the same applies to B, C,..., F) is executed (S501). This "processing" corresponds to, for example, "function key processing", or "keyboard key processing" in the main flow chart shown in Fig. 37. Thereafter, the MCPU interrupt processing and the SCPU interrupt processing are executed, so that the MCPU 1012 andthe SCPU 1022 simultaneously start sound source processing (S502 and S503). Upon completion of the SCPU interrupt processing of the SCPU 1022, the SCPU processing end signal B is input to the MCPU 1012. In the MCPU interrupt processing, the sound source processing is ended earlier than the SCPU interrupt processing, and the MCPU waits for the end of the SCPU interrupt processing the SCPU processing end signal B is discriminated in the MCPU interrupt processing, waveform data generated by the SCPU 1022 is supplied to the MCPU 1012, and is added to the waveform data generated by the MCPU 1012. The waveform data is then output to the left and right D/A converter units 1072 and 1082. Thereafter, the control returns to some processing B in the main flow chart.

The above-mentioned operations are repeated (S504 to S516) while executing the sound source processing for all the tone generation channels (16 channels as a total of those of the MCPU 1012 and the SCPU 1022). The repetition processing continues as long as musical tones are being produced.

Data Architecture in Source Processing

The sound source processing executed in step S411 (Fig. 38) and step S415 (Fig. 39) will be described in detail below.

In this embodiment, as described above, the two CPUs, i.e., the MCPU 1012 and the SCPU 1022 share the sound source processing in units of eight channels. Data for the sound source processing for eight channels are set in areas corresponding to the respective tone generation channels in the RAMs 2062 and 3062 of the MCPU 1012 and the SCPU 1022, as shown in Fig. 47.

Buffers BF, BT, B, and M are allocated on the RAM, as shown in Fig. 50.

In each tone generation channel area shown in Fig. 47, an arbitrary sound source method can be set by an operation (to be described in detail later), as schematically shown in Fig. 48. When the sound source method is set, data are set in each tone generation channel area in Fig. 47 in a data format of the corresponding sound source method, as shown in Fig. 49. In this embodiment, as will be described later, different sound methods can be assigned to the tone generation channels.

In Table 1 showing the data formats of the respective sound source methods shown in Fig. 49, G indicates a sound source method number for identifying the sound source methods. A represents an address designated when waveform data is read out in the sound source processing, and A_I, A_1, and A_2 represent integral parts of current addresses, and directly correspond to addresses of the external memory 1162 (Fig. 34) where waveform data are stored. A_F represents a decimal part of the current address, and is used for interpolating waveform data read out from the external memory 1162.

A_E and A_L respectively represent end and loop addresses. P_I, P_1 and P_2 represent integral parts of pitch data, and P_F represents a decimal part of pitch data. For example, $P_I = 1$ and $P_F = 0$ express a pitch of an original tone, PI = 2 and PF = 0 express a pitch higher than the original pitch by one octave, and $P_I = 0$ and $P_F = 0.5$ express a pitch lower by one octave.

X_P represents previous sample data, and X_N represents the next sample data. D represents a difference between two adjacent sample data, and E represents an envelope value. Furthermore, O represents an output value, and C rePresents a flag which is used when a sound source method to be assigned to a tone generation channel is changed in accordance with performance data, as will be described later.

Various other control data will be described in descriptions of the respective sound source methods.

When data shown in Fig. 49 are stored in the RAMs 2062 and 3062 of the MCPU 1012 and the SCPU 1022, and the sound source methods (to be described later) are determined, data are set in units of channels shown in Fig. 47 in the format shown in Fig. 49.

The sound source processing operations of the

respective sound source methods executed using the above-mentioned data architecture will be described below in turn. These sound source processing operations are realized by analyzing and executing a sound source processing program stored in the control ROM 2012 or 3012 by the command analyzer 2072 or 3072 of the MCPU 1012 or the SCPU 1022. Assume that the processing is executed under this condition unless otherwise specified.

In the flow chart shown in Fig. 40, in the sound source processing (one of steps S417 to S424) for each channel, the sound source method No. data G of the data in the data format (Table 1) shown in Fig. 49 stored in the corresponding tone generation channel of the RAM 2062 or 3062 is discriminated to determine sound source processing of a sound source method to be described below.

Sound Source Processing Based on PCM Method

When the sound source method No. data G indicates the PCM method, sound source processing based on the PCM method shown in the operation flow chart of Fig. 13 is executed. Variables in the flow chart are data in a PCM format of Table 1 shown in Fig. 49, which data are stored in the corresponding tone generation channel area (Fig. 47) of the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022.

Of an address group of the external memory 1162 (Fig. 34) where PCM waveform data are stored, an address where waveform data as an object to be currently processed is stored is assumed to be (A_I, A_F) shown in Fig. 15.

Pitch data (P_I, P_F) is added to the current address (S1001). The pitch data corresponds to the type of an ON key of the keyboard keys 8012 shown in Figs. 45 and 46.

It is then checked if the integral part A_I of the sum address is changed (S1002). If NO in step S1002, an interpolation data value O corresponding to the decimal part A_F of the address (Fig. 15) is calculated by arithmetic processing $D \times A_F$ using a difference D as a difference between sample data X_N and X_P at addresses ($A_I + 1$) and A_I (S1007). Note that the difference D has already been obtained by the sound source processing at previous interrupt timing (see step S1006 to be described later).

The sample data X_P corresponding to the integral part A_I of the address is added to the interpolation data value O to obtain a new sample data value O (corresponding to X_Q in Fig. 15) corresponding to the current address (A_I, A_F) (S1008).

Thereafter, the sample data is multiplied with the envelope value E (S1009), and the content of the obtained data O is added to a value held in the waveform data buffer B (Fig. 50) in the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022 (S1010).

Thereafter, the control returns to the main flow chart shown in Fig. 37. The control is interrupted in the next sampling period, and the operation flow chart of the sound source processing shown in Fig. 13 is executed again. Thus, pitch data (P_I, P_F) is added to the current address (A_I, A_F) (S1001).

The above-mentioned operations are repeated until the integral part A_I of the address is changed (S1002).

Before the integral part is changed, the sample data X_P and the difference D are left unchanged, and only the interpolation data O is updated in accordance with the address A_F. Thus, every time the address A_F is updated, new sample data X_Q is obtained.

If the integral part A_I of the current address is changed (S1002) as a result of addition of the current address (A_I, A_F) and the pitch data (P_I, P_F) in step S1001, it is checked if the address A_I has reached or exceeded the end address A_E (S1003).

If YES in step S1003, the next loop processing is executed. More specifically, a value (A_I - A_E) as a difference between the updated current address A_T and the end address A_E is added to the loop address A_L to obtain a new current address (A_I, A_F). A loop reproduction is started from the obtained new current address A_I (S1004). The end address A_E is an end address of an area of the external memory 1162 (Fig. 34) where PCM waveform data are stored. The loop address A_L is an address of a position where a player wants to repeat an output of a waveform, and known loop processing is realized by the PCM method.

If NO in step S1003, the processing in step S1004 is not executed.

Sample data is then updated. In this case, sample data corresponding to the new updated current address A_T and the immediately preceding address (A_I-1) are read out as X_N and X_P from the external memory 1162 (Fig. 34) (S1005).

Furthermore, the difference so far is updated with a difference D between the updated data X_N and X_P (S1006).

The following operation is as described above.

In this manner, waveform data by the PCM method for one channel is generated.

Sound Source Processing Based on DPCM Method

The sound source processing based on the DPCM method will be described below.

The operation principle of the DPCM method will be briefly described below with reference to Fig. 16.

In Fig. 16, sample data X_P corresponding to an

address A_I of the external memory 1162 (Fig. 34) is obtained by adding sample data corresponding to an address (A_I-1) (not shown) to a difference between the sample data corresponding to the address (A_I-1) and sample data corresponding to the address A_I.

A difference D with the next sample data is written at the address A_I of the external memory 1162 (Fig. 34). Sample data at the next address ($A_I + 1$) is obtained by $X_P + D$.

In this case, if the decimal part of the current address is represented by A_F, as shown in Fig. 16, sample data corresponding to the current address A_F is obtained by $X_P + D \times A_F$.

In this manner, in the DPCM method, a difference D between sample data corresponding to the current address and the next address is read out from the external memory 1162 (Fig. 34), and is added to the current sample data to obtain the next sample data, thereby sequentially forming waveform data.

The operation of the above-mentioned DPCM method will be described below with reference to the operation flow chart shown in Fig. 14. Variables in the flow chart are DPCM data in Table 1 shown in Fig. 49, which data are stored in the corresponding tone generation channel area (Fig. 49) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022.

Of addresses on the external memory 1162 (Fig. 34) where DPCM differential waveform data are stored, an address where waveform data as an object to be currently processed is stored is assumed to be (A_I, A_F) shown in Fig. 16.

Pitch data (P_I, P_F) is added to the current address (A_I, A_F) (S1101).

It is then checked if the integral part A_I of the sum address is changed (S1102). If NO in step S1102, an interpolation data value O corresponding to the decimal part A_F of the address is calculated by arithmetic processing $D \times A_F$ using a difference D at the address A_I in Fig. 16 (S1114). Note that the difference D has already been obtained by the sound source processing at the previous interrupt timing (see steps S1106 and S1110 to be described later).

The interpolation data value O is added to sample data X_P corresponding to the integral part A_I of the address to obtain a new sample data value O (corresponding X_Q in Fig. 16) corresponding to the current address (A_I, A_F) (S1115).

Thereafter, the sample data value O is multiplied with an envelope value E (S1116), and the obtained value is added to a value stored in the waveform data buffer B (Fig. 50) in the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022 (S1117).

Thereafter, the control returns to the flow chart shown in Fig. 37. The control is interrupted in the next sampling period, and the operation flow chart of the sound source processing shown in Fig. 14 is executed again. Thus, pitch data (P_I, P_F) is added to the current address (A_I, A_F) (S1101).

The above-mentioned operations are repeated until the integral part A_I of the address is changed.

Before the integral part is changed, the sample data X_P and the difference D are left unchanged, and only the interpolation data O is updated in accordance with the address A_F. Thus, every time the address A_F is updated, new sample data X_Q is obtained.

If the integral part A_I of the present address is changed (S1102) as a result of addition of the current address (A_I, A_F) and the pitch data (P_I, P_F) in step S1101, it is checked if the address A_I has reached or exceeded the end address A_E (S1103).

If NO in step S1103, sample data corresponding to the integral part A_I of the updated current address is calculated by the loop processing in steps S1104 to S1107. More specifically, a value before the integral part A_I of the current address is changed is stored in a variable "old A_I" (see the column of DPCM in Table 1 shown in Fig. 49). This can be realized by repeating processing in step S1106 or S1113 (to be described later). The old A_I value is sequentially incremented in S1106, and differential waveform data in the external memory 1162 (Fig. 34) addressed by the old A_I values are read out as D in step S1107. The readout data D are sequentially accumulated on sample data X_P in step S1105. When the old A_I value becomes equal to the integral part A_I of the changed current address, the sample data X_P has a value corresponding to the integral part A_I of the changed current address.

When the sample data X_P corresponding to the integral part A_I of the current address is obtained in this manner, YES is determined in step S1104, and the control starts the arithmetic processing of the interpolation value (S1114) described above.

The above-mentioned sound source processing is repeated at the respective interrupt timings, and when the judgment in step S1103 is changed to YES, the control enters the next loop processing.

An address value (A_I-A_E) exceeding the end address A_E is added to the loop address A_L, and the obtained address is defined as an integral part A_I of a new current address (S1108).

An operation for accumulating the difference D several times depending on an advance in address from the loop address A_L is repeated to calculate sample data X_P corresponding to the integral part A_I of the new current address. More specifically, sample data X_P is initially set as the value of sample data X_{PL} (see the column of DPCM in Table 1 shown in Fig. 49) at the preset loop address A_L

and the old A_I is set as the value of the loop address A_L (S1110). The following processing operations in steps S1110 to S1113 are repeated. More specifically, the old A_I value is sequentially incremented in step S1113, and differential waveform data on the external memory 1162 (Fig. 34) designated by the incremented old A_I values read out as data D. The data D are accumulated on the sample data X_P in step S1112. When old A_I value becomes equal to the integral part A_I of the new current address, the sample data X_P has a value corresponding to the integral part A_I of the new current address after loop processing.

When the sample data Xp corresponding to the integral part A_I of the new current address is obtained in this manner, YES is determined in step S1111, and the control enters the above-mentioned arithmetic processing of the interpolation value (S1114).

As described above, waveform data by the DPCM method for one tone generation channel is generated.

Sound Source Processing Based on FM Method (Part 1)

The sound source processing based on the FM method will be described below.

In the FM method, hardware or software elements having the same contents, called "operators", as indicated by OP1 to OP4 in Figs. 51 to 54, are normally used, and are connected based on connection rules indicated by algorithms 1 to 4 in Figs. 51 to 54, thereby generating musical tones. In this embodiment, the FM method is realized by a software program.

The operation of this embodiment executed when the sound source processing is performed using two operators will be described below with reference to the operation flow chart shown in Fig. 17. The algorithm of the processing is shown in Fig. 18. Variables in the flow chart are FM format data in Table 1 shown in Fig. 49, which data are stored in the corresponding tone generation channel area (Fig. 47) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022.

First, processing of an operator 2 (OP2) as a modulator is performed. In pitch processing (processing for accumulating pitch data for determining an incremental width of an address for reading out waveform data stored in the waveform memory 1162), since no waveform data interpolation is performed unlike in the PCM method, an address consists of an integral address A_2, and has no decimal address. Further, modulation waveform data are stored in the external memory 1162 (Fig. 34) at sufficiently fine incremental widths.

Pitch data P_2 is added to the present address A_2 (S1301).

A feedback output F_{O2} is added to the address A_2 as a modulation input to obtain a new address A_{M2} which corresponds to phase of a sine wave (S1302). The feedback output F_{O2} has already been obtained upon execution of processing in step S1305 (to be described later) at the immediately preceding interrupt timing.

The value of a sine wave corresponding to the address A_{M2} is calculated. In practice, sine wave data are stored in the external memory 1162 (Fig. 34), and are obtained by addressing the external memory 1162 by the address A_{M2} to read out the corresponding data (S1303).

Subsequently, the sine wave data is multiplied with an envelope value E_2 to obtain an output O_2 - (S1304).

Thereafter, the output O_2 is multiplied with a feedback level F_{L2} to obtain a feedback output F_{O2} (S1305). This output F_{O2} serves as an input to the operator 2 (OP2) at the next interrupt timing.

The output O_2 is multiplied with a modulation level M_{L2} to obtain a modulation output M_{O2} - (S1306). The modulation output M_{O2} serves as a modulation input to an operator 1 (OP1).

The control then enters processing of the operator 1 (OP1). This processing is substantially the same as that of the operator 2 (OP2) described above, except that there is no modulation input based on the feedback output.

The current address A_1 of the operator 1 is added to pitch data P_1 (S1307), and the sum is added to the above-mentioned modulation output M_{O2} to obtain a new address A_{M1} (S1308).

The value of sine wave data corresponding to this address A_{M1} (phase) is read out from the external memory 1162 (Fig. 34) (S1309), and is multiplied with an envelope value E_1 to obtain a musical tone waveform output O_1 (S1310).

The output O_1 is added to a value held in the buffer B (Fig. 50) in the RAM 2062 (Fig. 35) or the RAM 3062 (Fig. 36) (S1311), thus completing the FM processing for one tone generation channel.

Sound Source Processing Based on TM (Triangular Wave Modulation) Method (Part 1)

The sound source processing based on the TM method will be described below.

The principle of the TM method is already described in the first embodiment. Therefore, the description of the TM method itself is omitted.

The sound source processing based on the TM method will be described below with reference to the operation flow chart shown in Fig. 19. In this case, the sound source processing is also performed using two operators like in the FM method shown in Figs. 17 and 18, and the algorithm of the

processing is shown in Fig. 20. Variables in the flow chart are TM format data in Table 1 shown in Fig. 49, which data are stored in the corre-sponding tone generation channel area (Fig. 47) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022.

First, processing of an operator 2 (OP2) as a modulator is performed. In pitch processing, since no waveform data interpolation is performed unlike in the PCM method, an address for addressing the external memory 1162 consists of only an integral address A_2.

The current address A_2 is added to pitch data P_2 (S1401).

A modified sine wave corresponding to the address A_2 (phase) is read out from the external memory 1162 (Fig. 34) by the modified sine conversion f_c, and is output as a carrier signal O_2 - (S1402).

Subsequently, a feedback output F_{O2} (S1460) as a modulation signal, is added to the carrier signal O_2, and the sum signal is output as a new address O_2 (S1403). The feedback output F_{O2} has already been obtained upon execution of processing in step S1406 (to be described later) at the immediately preceding interrupt timing.

The value of a triangular wave corresponding to the address O_2 is calculated. In practice, triangular wave data are stored in the external memory 1162 (Fig. 34), and are obtained by addressing the external memory 1162 by the address O_2 to read out the corresponding data (S1404).

Subsequently, the triangular wave data is multiplied with an envelope value E_2 to obtain an output O_2 (S1405).

Thereafter, the output O_2 is multiplied with a feedback level F_{L2} to obtain a feedback output F_{O2} (S1407). In this embodiment, the output F_{O2} serves as an input to the operator 2 (OP2) at the next interrupt timing.

The output O_2 is multiplied with a modulation level M_{L2} to obtain a modulation output M_{O2} - (S1407). The modulation output M_{O2} serves as a modulation input to an operator 1 (OP1).

The control then enters processing of the operator 1 (OP1). This processing is substantially the same as that of the operator 2 (OP2) described above, except that there is no modulation input based on the feedback output.

The current address A_1 of the operator 1 is added to pitch data P_1 (S1408), and the sum is subjected to the above-mentioned modified sine conversion to obtain a carrier signal O_1 (S1409).

The carrier signal O_1 is added to the modulation output M_{O2} to obtain a new value O_1 (S1410), and the value O_1 is subjected to triangular wave conversion (S1411). The converted value is multiplied with an value E_1 to obtain a musical tone waveform output O_1 (S1412).

The output O_1 is added to a value held in the buffer B (Fig. 50) in the RAM 2062 (Fig. 36) or the RAM 3062 (Fig. 36), thus completing the TM processing for one tone generation channel.

The sound source processing operations based on four methods, i.e., the PCM, DPCM, FM, and TM methods have been described. The FM and TM methods are modulation methods, and, in the above examples, two-operator processing operations are executed based on the algorithms shown in Figs. 18 and 20. However, in sound source processing in an actual performance, more operators are used, and the algorithms are more complicated. Figs. 51 to 54 show examples. In an algorithm 1 shown in Fig. 51, four modulation operations including a feedback input are performed, and a complicated waveform can be obtained. In each of algorithms 2 and 3 shown in Figs. 52 and 53, two sets of algorithms each having a feedback input are arranged parallel to each other, and these algorithms are suitable for expressing a change in tone color during, e.g., transition from an attack portion to a sustain portion. An algorithm 4 shown in Fig. 59 has a feature close to a sine wave synthesis method.

The sound source processing operations based on the FM and TM methods using four operators shown in Figs. 51 to 54 will be described below in turn with reference to Figs. 55 and 56.

Sound Source Processing Based on FM Method (Part 2)

Fig. 55 is an operation flow chart of normal sound source processing based on the FM method corresponding to the algorithm 1 shown in Figs. 55 to 54. Variables in the flow chart are stored in the corresponding tone generation channel area (Fig. 47) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022. Although the variables used in Fig. 55 are not the same as data in the FM format of Table 1 in Fig. 49, they are obtained by expanding the concept of the data format shown in Fig. 49, and only have different suffixes.

First, the present address A_4 of an operator 4 (OP4) is added to pitch data P_4 (S1901). The address A_4 is added to a feedback output F_{O4} - (S1905) as a modulation input to obtain a new address A_{M4} (S1902). Furthermore, the value of a sine wave corresponding to the address M4 (phase) is read out from the external memory 1162 (Fig. 34) (S1903), and is multiplied with an envelope value E_4 to obtain an output O_4 (S1904). Thereafter, the output O_4 is multiplied with a feedback level F_{L4} to obtain a feedback output F_{O4} - (S1905). The output O_4 is multiplied with a modulation level M_{L4} to obtain a modulation output M_{O4} -

(S1906). The modulation output M_{O4} serves as a modulation input to the next operator 3 (OP3).

The control then enters processing of the operator 3 (OP3). This processing is substantially the same as that of the operator 4 (OP4) described above, except that there is no modulation input based on the feedback output. The current address A_3 of the operator 3 (OP3) is added to pitch data P_3 to obtain a new current address A_3 (S1907). The address A_3 is added to a modulation output M_{O4} as a modulation input, thus obtaining a new address A_{M3} (S1908). Furthermore, the value of a sine wave corresponding to the address A_{M3} - (phase) is read out from the external memory 1162 (Fig. 34) (S1909), and is multiplied with an envelope value E_3 to obtain an output O_3 (S1910). Thereafter, the output O_3 is multiplied with a modulation level M_{L3} to obtain a modulation output M_{O3} - (S1911). The modulation output M_{O3} serves as a modulation input to the next operator 2 (OP2).

Processing of the operator 2 (OP2) is then executed. However, this processing is substantially the same as that of the operator 3, except that a modulation input is different, and a detailed description thereof will be omitted.

Finally, the control enters processing of an operator 1 (OP1). In this case, the same processing operations as described above are performed up to step S1920. A musical tone waveform output O_1 obtained in step S1920 is added to data stored in the buffer B as a carrier (S1921).

Sound Source Processing Based on TM Method (Part 2)

Fig. 50 is an operation flow chart of normal sound source processing based on the TM method corresponding to the algorithm 1 shown in Fig. 51. Variables in the flow chart are stored in the corresponding tone generation channel area (Fig. 47) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022. Although the variables used in Fig. 55 are not the same as data in the TM format of Table 1 in Fig. 49, they are obtained by expanding the concept of the data format shown in Fig. 49, and only have different suffixes.

The current address A_4 of the operator 4 (OP4) is added to pitch data P_4 (S2061). A modified sine wave to the above-mentioned address A_4 (phase) is read out from the external memory 1162 (Fig. 34) by the modified sine conversion f_c, and is output as a carrier signal O_4 (S2002). A feedback output F_{O4} (see S2007) as a modulation signal is added to the carrier signal O_4, and the sum signal is output as a new address O_4 (S2003). The value of a triangular wave corresponding to the address O_4 (phase) is read out from the external memory 1162 (Fig. 34) (to be referred to as a triangular wave conversion hereinafter) (S2004), and is multiplied with an envelope value E_4, thus obtaining an output O_4 (S2005). Thereafter, the output O_4 is multiplied with a modulation level M_{L4} to obtain a modulation output M_{O4} (S2006). The output O_4 is multiplied with a feedback level F_{L4} to obtain a feedback output F_{O4} (S2007). The modulation output M_{O4} serves as a modulation input to the next operator 3 (OP3).

The control then enters processing of the operator 3 (OP3). This processing is substantially the same as that of the operator 4 (OP4) described above, except that there is no modulation input based on the feedback output. The current address A_3 of the operator 3 (OP3) is added to pitch data P_3 (S2008) and the sum is subject to modified sine conversion to obtain a carrier signal O_3 (S2009). The carrier signal O_3 is added to the above-mentioned modulation output M_{O4} to obtain a new value O_3 (S2010), and the value O_3 is subject to triangular wave conversion (S2011). The converted value is multiplied with an envelope value E_3 to obtain an aoutput O_3 (S2012). The output O3 is multiplied with a modulation level M_{L3} to obtain a modulation output M_{O3} (S2013). The modulation output M_{O3} serves as a modulation input to the next operator 2 (OP2).

Processing of the operator 2 (OP2) is then executed. However, this processing is substantially the same as that of the operator 3, except that a modulation input is different, and a detailed description thereof will be omitted.

Finally, the control enters processing of an operator 1 (OP1). In this case, the same processing operations as described above are performed up to step S2024. A musical tone waveform output O_1 obtained in step S2024 is accumulated in the buffer B (Fig. 50) as a carrier (S2025).

The embodiment of the normal sound processing operations based on the modulation methods has been described. However, the above-mentioned processing is for one tone generation channel, and in practice, the MCPU 1012 and the SCPU 1022 each execute processing for eight channels (Fig. 40). If a modulation method is designated in a given tone generation channel, the above-mentioned sound source processing based on the modulation method is executed.

Modification of Modulation Method (Part 1)

The first modulation of the sound source processing based on the modulation method will be described below.

The basic concept of this processing is shown in the flow chart of Fig. 57.

In Fig. 57, operator 1, 2, 3, and 4 processing operations have the same program architecture al-

though they have different variable names to be used.

Each operator processing cannot be executed unless a modulation input is determined. This is because a modulation input to each operator processing varies depending on the algorithm, as shown in Figs. 51 to 54. Which operator processing output is used as a modulation input or whether or not an output from its own operator processing is fed back, and is used as its own modulation input in place of another operator processing must be determined. In the operation flow chart shown in Fig. 57, such determinations are simultaneously performed in algorithm processing (S2105), and the connection relationship obtained by this processing determine modulation inputs to the respective operator processing operations (S2102 to S2104). Note that a given initial value is set as an input to each operator processing at the beginning of tone generation.

When the operator processing and the algorithm processing are separated in this manner, the program of the operator processing can remain the same, and only the algorithm processing can be modified in correspondence with algorithms. Therefore, the program size of the overall sound source processing based on the modulation method can be greatly reduced.

A modification of the FM method based on the above-mentioned basic concept will be described below. The operator 1 processing in the operation flow chart showing operator processing based on the FM method in Fig. 57 is shown in Fig. 58, and an arithmetic algorithm per operator is shown in Fig. 59. The remaining operator 2 to 4 processing operations are the same except for different suffix numbers of variables. Variables in the flow chart are stored in the corresponding tone generation channel (Fig. 47) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022.

An address A_1 corresponding to a phase angle is added to pitch data P_1 to obtain a new address A_1 (S2201). The address A_1 is added to a modulation input M_{I1}, thus obtaining an address A_{M1} - (S2202). The modulation input M_{I1} is determined by the algorithm processing in step S2105 (Fig. 57) at the immediately preceding interrupt timing, and may be a feed back output F_{O1} of its own operator, or an output M_{O2} from another operator, e.g., an operator 2 depending on the algorithm. The value of a sine wave corresponding to this address (phase) A_{M1} is read out from the external memory 1162 (Fig. 34), thus obtaining an output O_1 - (S2203). Thereafter, a value obtained by multiplying the output O_1 with envelope data E_1 serves as an output O_1 of the operator 1 (S2204). The output O_1 is multiplied with a feedback level F_{L1} to obtain a feedback output F_{O1} (S2205). The output 01 is multipled with a modulation level M_{L1}, thus obtaining a modulation output M_{O1} (S2206).

A modification of the FM method based on the above-mentioned basic concept will be described below. The operator 1 processing in the operation flow chart showing operator processing based on the FM method in Fig. 57 is shown in Fig. 58, and an arithmetic algorithm per operator is shown in Fig. 59. The remaining operator 2 to 4 processing operations are the same except for different suffix numbers of variables. Variables in the flow chart are stored in the corresponding tone generation channel (Fig. 47) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022.

The current address A_1 is added to pitch data P_1 (S2301). A modified sine wave corresponding to the above-mentioned address A_1 (phase) is read out from the external memory 1162 (Fig. 34) by the modified sine conversion f_c, and is generated as a carrier signal O_1 (S2302). The output O_1 is added to a modulation input M_{I1} as a modulation signal, and the sum is defined as a new address O_1 - (S2303). The value of a triangular wave corresponding to the address O_1 (phase) is read out from the external memory 1162 (S2304), and is multiplied with an envelope value E_1 to obtain an output O_1 (S2306). Thereafter, the output O_1 is multiplied with a feedback level F_{L1} to obtain a feedback output F_{O1} (S2306). The output O_1 is multiplied with a modulation level M_{L1} to obtain a modulation output M_{O1} (S2307).

The algorithm processing in step S2105 in Fig. 57 for determining a modulation input in the operator processing in both the above-mentioned modulation methods, i.e., the FM and TM methods will be described in detail below with reference to the operation flow chart of Fig. 62. The flow chart shown in Fig. 62 is common to both the FM and TM methods, and the algorithms 1 to 4 shown in Figs. 51 to 54 are selectively processed. In this case, choices of the algorithms 1 to 4 are made based on an instruction (not shown) from a player (S2400).

The algorithm 1 is of a series four-operator (to be abbreviated to as an OP hereinafter) type, and only the OP4 has a feedback input. More specifically, in the algorithm 1,

a feedback output F_{O4} of the OP4 serves as the modulation input M_{I4} of the OP4 (S2401),

a modulation output M_{O4} of the OP4 serves as a modulation input M_{I3} of the OP3 (S2402),

a modulation output OP3 of the OP3 serves as a modulation input M_{I2} of the OP2 (S2403),

a modulation output M_{O2} of the OP2 serves as a modulation input M_{I1} of the OP1 (S2404), and

an output O_1 from the OP1 is added to the value held in the buffer B (Fig. 50) as a carrier output (S2405).

In the algorithm 2, as shown in Fig. 52, the OP2 and the OP4 have feedback inputs. More specifically, in the algorithm 2,

a feedback output F_{O4} of the OP4 serves as a modulation input M_{I4} of the OP4 (S2406),

a modulation output M_{O4} of the OP4 serves as a modulation input M_{I3} of the OP3 (S2407),

a feedback output F_{O2} of the OP2 serves as a modulation input M_{I2} of the OP2 (S2408),

modulation outputs M_{O2} and M_{O3} of the OP2 and serve as a modulation input M_{I1} of the OP1 (S2409), and

an output O_1 from the OP1 is added to the value held in the buffer B as a carrier output (S2410).

In the algorithm 3, the OP2 and OP4 have feedback inputs, and two modules in which two operators are connected in series with each other are connected in parallel with each other. More specifically, in the algorithm 3,

a feedback output F_{O4} of the OP4 serves as a modulation input M_{I4} of the OP4 (S2411),

a modulation output M_{O4} of the OP4 serves as a modulation input M_{I3} of the OP3 (S2412),

a feedback output F_{O2} of the OP2 serves as a modulation input M_{I2} of the OP2 (S2413),

a modulation output M_{O2} of the OP2 serves as a modulation input M_{I1} of the OP1 (S2414), and

outputs O_1 and O_3 from the OP1 and OP3 are added to the value held in the buffer B as carrier outputs (S2415).

The algorithm 4 is of a parallel four-OP type, and all the OPs have feedback inputs. More specifically, in the algorithm 4,

a feedback output F_{O4} of the OP4 serves as a modulation input M_{I4} of the OP4 (S2416),

a feedback output F_{O3} of the OP3 serves as a modulation input M_{I3} of the OP3 (S2417),

a feedback output F_{O2} of the OP2 serves as a modulation input M_{I2} of the OP2 (S2418),

a feedback output F_{O1} of the OP1 serves as a input M_{I1} of the OP1 (S2419), and

outputs O_1, O_2, O_3, and O_4 from all the OPs are added to the value held in the buffer B (S2420).

The sound source processing for one channel is completed by the above-mentioned operator processing and algorithm processing, and tone generation (sound source processing) continues in this state unless the algorithm is changed.

Modification of Modulation Method (Part 2)

The second modification of the sound source processing based on the modulation method will be described below.

In the various modulation methods described above, processing time is increased as the complicated algorithms are programmed, and as the number of tone generation channels (the number of polyphonic channels) is increased.

In the second modification to be described below, the first modification shown in Fig. 57 is further developed, so that only operator processing is performed at a given interrupt timing, and only algorithm processing is performed at the next interrupt timing. Thus, the operator processing and the algorithm processing are alternately executed. In this manner, a processing load per interrupt timing can be greatly reduced. As a result, one sample data per two interrupts is output.

This operation will be described below with reference to the operation flow chart shown in Fig. 63.

In order to alternately execute the operator processing and the algorithm processing, whether or not a variable S is zero is checked (S2501). The variable is provided for each tone generation channel, and is stored in the corresponding tone generation channel area (Fig. 47) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022.

If S = 0 at a given interrupt timing, the process enters an operator processing route, and sets the variable S to a value "1" (S2502). Subsequently, operator 1 to 4 processing operations are executed (S2503 to S2506). This processing is the Same as that in Figs. 58 and 59, or 60 and 61.

The process exits from the operator processing route, and executes output processing for setting a value of the buffer BF (for the FM method) or the buffer BT (for the TM method) (S2510). The buffer BF or BT is provided for each tone generation channel, and is stored in the corresponding tone generation channel area (Fig. 47) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022. The buffer BF or BT stores a waveform output value after the algorithm processing. At the current interrupt timing, however, no algorithm processing been executed, and the content of the buffer BF or BT is not updated. For this reason, the same waveform output value as that at the immediately preceding interrupt timing is output.

With the above processing, sound source processing for one tone generation channel at the current interrupt timing is completed. In this case, data obtained by the current operator 1 to 4 processing operations are stored in the corresponding tone generation channel area (Fig. 47) on the RAM 2062 or 3062 of the MCPU 1012 or the SCPU 1022.

At the next interrupt timing, since the variable S is set to be 1 at the immediately preceding interrupt timing, the flow advances to step S2507. The process then enters an algorithm processing route, and sets the variable S to be a value "0". Subsequently, the algorithm processing is execut-

ed (S2508).

In this processing, the data processed in the operator 1 to 4 processing operations at the immediately preceding interrupt timing and stored in the corresponding tone generation channel area (Fig. 47) are used, and processing for determining a modulation input for the next operator processing is executed. In this processing, the content of the buffer BF or BT is rewritten, and a waveform output value at that interrupt timing can be obtained. The algorithm processing is shown in detail in the operation flow chart of Fig. 64. In this flow chart, the same processing operations as in Fig. 62 are executed in steps denoted by the same reference numerals as in Fig. 62. A difference between Figs. 62 and 64 is an output portion in steps S2601 to S2604. In the case of algorithms 1 and 2, the content of the output O_1 of the operator 1 processing is directly stored in the buffer BF or BT (S2601 and S2602). In the case of the algorithm 3, a value as a sum of the outputs O_1 and O_3 is stored in the buffer BF or BT (S2603). Furthermore, in the case of the algorithm 4, a value as a sum of the output O_1 and the outputs O_2, O_3, and O_4 is stored in the buffer BF or BT (S2604).

As described above, since the operator processing and the algorithm processing are alternately executed at every other interrupt timing, a processing load per interrupt timing of the sound source processing program can be remarkably decreased. In this case, since an interrupt period need not be prolonged, the processing load can be reduced without increasing an interrupt time of the main operation flow chart shown in Fig. 37, i.e., without influencing the program operation. Therefore, a keyboard key sampling interval executed in Fig. 37 will not be prolonged, and the response performance of an electronic musical instrument will not be impaired.

The operations for generating musical tone data in units of tone generation channels by the software sound source processing operations based on various sound source methods have been described.

Function Key Processing

The operation of the function key processing (S403) in the main operation flow chart shown in Fig. 37 when an actual electronic musical instrument is played will be described in detail below.

In the above-mentioned sound source processing executed for each tone generation channel, parameters corresponding to sound source methods are set in the formats shown in Fig. 49 in the corresponding tone generation channel area (Fig. 47) on the RAM 2062 or 3062 (Figs. 35 and 36) by one of the function keys 8012 (Fig. 45) connected to the operation panel of the electronic musical instrument via the input port 2102 (Fig. 35) of the MCPU 1012.

Fig. 65 shows an arrangement of some function keys 8012 shown in Fig. 45. In Fig. 65, some function keys 8012 are realized as tone color switches. When one of switches "piano", "guitar",..., "koto" in a group A is depressed, a tone color of the corresponding instrument tone is selected, and a guide lamp is turned on. Whether the tone color of the selected instrument tone is generated in the DPCM method or the TM method is selected by a DPCM/TM switch 27012.

On the other hand, when a switch "tuba" in a group B is depressed, a tone color based on the FM method is designated; when a switch "bass" is depressed, a tone color on both the PCM and TM methods is designated; and when a switch "trumpet" is depressed, a tone color based on the PCM method is designated. Then, a musical tone based on the designated sound source method is generated.

Figs. 66 and 67 show of sound source methods to the respective tone generation channel region (Fig. 47) on the RAM 2062 or 3062 when the switches "piano" and "bass" are depressed. When the switch "piano" is depressed, the DPCM method is assigned to all the 8-tone polyphonic tone generation channels of the MCPU 1012 and the SCPU 1022, as shown in Fig. 66. When the switch "bass" is depressed, the PCM method is assigned to the odd-numbered tone generation channels, and the TM method is assigned to the even-numbered tone generation channels, as shown in Fig. 67. Thus, a musical tone waveform for one musical tone can be obtained by mixing tone waveforms generated in the two tone generation channels based on the PCM and TM methods. In this case, a 4-tone polyphonic system per CPU is attained, and an 8-tone polyphonic system as a total of two CPUs is attained.

Fig. 68 is a partial operation flow chart of the function key processing in step S403 in the main operation flow chart shown in Fig. 37, and shows processing corresponding to the tone color designation switch group shown in Fig. 65.

It is checked if a player operates the DPCM/TM switch 27012 (S2901). If YES in step S2901, it is checked if a variable M is zero (S2902). The variable M stored on the RAM 2062 (Fig. 35) of the MCPU 1012, and has a value "0" for the DPCM method; a value "1" for the TM method. If YES in step S2902, i.e., if it is determined that the value of the variable M is 0, the variable M is set to be a value "1" (S2903). This means that the DPCM/TM switch 27012 is depressed in the DPCM method selection state, and the selection state is changed to the TM method

selection state. However, if NO in step S2902, i.e., if it is determined that the value of the variable M is "1", the variable M is set to be a value "0" (S2904). This means that the DPCM/TM switch 27012 is depressed in the TM method selection state, and the selection state is changed to the DPCM method selection state.

It is checked if a tone color in the group A shown in Fig. 65 is currently designated (S2905). Since the DPCM/TM switch 27012 is valid for tone co]ors of only group A, only when a tone color in the group A is designated, and YES is determined in step S2905, operations corresponding to the DPCM/TM switch 27012 in steps S2906 to S2908 are executed.

It is checked if the variable M is "0" (S2906).

If YES in step S2906, since the DPCM method is selected by the DPCM/TM switch 27012, DPCM data are set in the DPCM format shown in Fig. 49 in the corresponding tone generation channel areas on the RAMs 2062 and 3062 (Figs. 35 and 36). More specifically, sound source method No. data G indicating the DPCM method is set in the start area of the corresponding tone generation channel area (see the column of DPCM in Fig. 49). Subsequently, various parameters corresponding to currently designated tone colors are respectively set in the second and subsequent areas of the corresponding tone generation channel area (S2907).

If NO in step S2906, since the TM method is selected by the DPCM/TM switch 27012, TM data are set in the TM format shown in Fig. 49 in the corresponding generation channel areas. More specifically, sound source method No. data G indicating the TM method is set in the start area of the corresponding tone generation channel area. Subsequently, various parameters corresponding to currently designated tone colors are respectively set in the second and subsequent areas of the corresponding tone generation channel area (S2908).

A case has been exemplified wherein the DPCM/TM switch 27012 shown in Fig. 65 is operated. If the switch 27012 is not operated and NO is determined in step S2901, or if tone color of the group A is not designated and NO is determined in step S2905, processing from step S2909 is executed.

It is checked in step S2909 if a change in tone color switch shown in Fig. 65 is detected.

If NO in step S2909, since processing for the tone color switches need not be executed, the function key processing (S403 in Fig. 37) is ended.

If it is determined that a change in tone color switch is detected, and YES is determined in step S2909, it is checked if a tone color in the group B is designated (S2910).

If a tone color in the group B is designated, and YES is determined in step S2910, data for the sound source method corresponding to the designated tone color are set in the predetermined format in the corresponding tone generation channel areas on the RAMs 2062 and 3062 (Figs. 35 and 36). More specifically, sound source method No. data G indicating the sound source method is set in the start area of the corresponding tone generation channel area (Fig. 49). Subsequently, various parameters corresponding to the currently designated tone color are respectively set in the second and subsequent areas of the corresponding tone generation channel area (S2911). For example, when the switch "bass" in Fig. 65 is selected, data corresponding to the PCM method are set in the odd-numbered tone generation channel areas, and data corresponding to the TM method are set in the even-numbered tone generation channel areas.

If it is determined that the tone color switch in the group A is designated, and NO is determined in step S2910, it is checked if the variable M is "1" (S2912). If the TM method is currently selected, and YES is determined in step S2912, data are set in the TM format (Fig. 49) in the corresponding tone generation channel area (S2913) like in step S2908 described above.

If the DPCM method is selected, and NO is determined in step S2912, data are set in the DPCM format (Fig. 49) in the corresponding tone generation channel area (S2914) like in step S2907 described above.

Embodiment A of ON Event Keyboard Key Processing

The operation of the keyboard key processing (S405) in the main operation flow chart shown in Fig. 37 executed when an actual electronic musical instrument is played will be described below.

The first embodiment of ON event keyboard key processing will be described below.

In this embodiment, when a tone color in the group A shown in Fig. 65 is designated, the sound source method to be set in the corresponding tone generation channel area of the RAM 2062 or 3062 (Figs. 35 and 36) is automatically switched in accordance with an ON key position, i.e., a tone range of a musical tone. This embodiment has a boundary between key code numbers 31 and 32 on the keyboard shown in Fig. 46. That is, when a key code of an ON key falls within a bass tone range equal to or lower than the 31st key code, the DPCM method is assigned to the corresponding tone generation channel. On the other hand, when a key code of an ON key falls within a high tone range equal to or higher than the 32nd key code, the TM method is assigned to the corresponding

tone generation channel. When a tone color in the group B in Fig. 65 is designated, no special keyboard key processing is executed.

Fig. 69 is a partial operation flow chart of the keyboard key processing in step S405 in the main operation flow chart of Fig. 37.

It is checked if a tone color in the group A is currently designated (S3001).

If NO in step S3001, and a tone color in the group B is currently designated, special processing in Fig. 69 is not performed.

If YES in step S3001, and a tone color in the group A is currently designated, it is checked if a key code of a key which is detected as an "ON key" in the keyboard key scanning processing in step S404 in the main operation flow chart shown in Fig. 37 is equal to or lower than the 31st key code (S3002).

If a key in the bass tone range equal to or lower than the 31st key code is depressed, and YES is determined in step S3002, it is checked if the variable M is "1" (S3003). The variable M is set in the operation flow chart shown in Fig. 68 as a part of the function key processing in step S403 in the main operation flow chart shown in Fig. 37, and is "0" for the DPCM method; "1" for the TM method, as described above.

If YES (M = "1") in step S3003, i.e., if it is determined that the TM method is currently designated as the sound source method, DPCM data in Fig. 49 are set in a tone generation channel area of the RAM 2062 or 3062 (Figs. 35 and 36) where the ON key is assigned so as to change the TM method to the DPCM method as a sound source method for the bass tone range (see the column of DPCM in Fig. 49). More specifically, sound source method No. data G indicating the DPCM method is set in the start area of the corresponding tone generation channel area. Subsequently, various parameters corresponding to the currently designated tone color are respectively set in the second and subsequent areas of the corresponding tone generation channel area (S3004). Thereafter, a value "1" is set in a flag C (S3005). The flag C is a variable (Fig. 49) stored in each tone generation channel area on the RAM 2062 (Fig. 35) of the MCPU 1012, and is used in OFF event processing to be described later with reference to Fig. 71.

If it is determined that a key in the high tone range equal to or higher than the 31st key code is depressed, and NO is determined in step S3002, it is checked if the variable M is "1" (S3006).

If NO (M = "0") in step S3006, i.e., if it is determined that the DPCM method is currently designated as the sound source method, TM data in Fig. 49 are set in a tone generation channel area of the RAM 2062 or 3062 (Figs. 35 and 36) where the ON key is assigned so as to change the DPCM method to the TM method as a sound source method for the high tone range (see the column of TM in Fig. 49). More specifically, sound source method No. data G indicating the TM method is set in the start area of the corresponding tone generation channel area. Subsequently, various parameters corresponding to the currently designated tone color are respectively set in the second and subsequent areas of the corresponding tone generation channel area (S3007). Thereafter, a value "2" is set in a flag C (S3008).

In the above-mentioned processing, if NO in step S3003 and if YES in step S3006, since the desired sound source method is originally selected, no special is executed.

Embodiment B of ON Event Keyboard Key Processing

The second embodiment of the ON event keyboard key processing will be described below.

In the embodiment B of the ON event keyboard key processing, when a tone color in the group A in Fig. 65 is designated, a sound source method to be set in the corresponding tone generation channel area (Fig. 47) on the RAM 2062 or 2062 (Figs. 35 and 36) of the MCPU 1012 or the SCPU 1022 is automatically switched in accordance with an ON key speed, i.e., a velocity. In this case, a switching boundary is set at a velocity value "64" half the maximum value "127" of the MIDI (Musical Instrument Digital Interface) standards. That is, when the velocity value of an ON key is equal to or larger than 64, the DPCM method is assigned; when the velocity of an ON key is equal to or smaller than 64, the TM method is assigned. When a tone color in the group B in Fig. 65 is designated, no special keyboard key processing is executed.

Fig. 70 is a partial operation flow chart of the keyboard key processing in step S405 in the main operation flow chart shown in Fig. 37.

It is checked if a tone color in the group A in Fig. 65 is currently designated (S3101).

If NO in step S3101, and a tone color in the group B is presently selected, the special processing in Fig. 69 is not executed.

If YES in step S3101, and a tone color in the group A is presently selected, it is checked if the velocity of a key which is detected as an "ON key" in the keyboard key scanning processing in step S404 in the main operation flow Chart Shown in Fig. 37 is equal to or larger than 64 (S3102). Note that the velocity value "64" corresponds to "mp (mezzo piano)" of the MIDI standards.

If it is determined that the velocity value is equal to or larger than 64, and YES is determined in step S3102, it is checked if the variable M is "1"

(S3102). The variable M is set in the operation flow chart shown in Fig. 68 as a part of the function key processing in step S403 in the main operation flow chart shown in Fig. 37, and is "0" for the DPCM method; "1" for the TM method, as described above.

If YES (M = "1") in step S3103, and the TM method is currently designated as the sound source method, DPCM data in Fig. 49 are set in a tone generation channel area of the RAM 2062 or 3062 (Figs. 35 and 36) where the ON key is assigned so as to change the TM method to the DPCM method as a sound source method for a fast ON key operation (S3104), and a value "1" is set in the flag C (S3105).

If it is determined that the velocity value is smaller than 64 and NO is determined in step S3102, it is further checked if the variable M is "1" (S3106).

NO (M = "0") in step S3106, and the DPCM method is currently designated as the sound source method, TM data in Fig. 49 are set in a tone generation channel area of the RAM 2062 or 3062 where the ON key is assigned so as to change the DPCM method to the TM method as a sound source method for a slow ON key operation (S3107). Thereafter, a value "2" is set in the flag C (S3108).

In the above-mentioned processing, if NO in step S3103 and if YES in step S3106, since the desired sound source method is originally selected, no special processing is executed.

Embodiment of OFF Event Keyboard Processing

The embodiment of the OFF event keyboard key processing will be described below.

According to the above-mentioned ON event keyboard key processing, the sound source method is automatically set in accordance with a key range (tone range) or a velocity. Upon an OFF event, the set sound source method must be restored. The embodiment of the OFF event keyboard key processing to be described below can realize this processing.

Fig. 71 is a partial operation flow chart of the keyboard key processing in step S405 in the main operation flow chart shown in Fig. 37.

The value of the flag C set in the tone generation channel area on the RAM 2062 or 3062 (Figs. 35 and 36), where the key determined as an "OFF key" in the keyboard key scanning processing in step S404 in the main operation flow chart of Fig. 37 is assigned, is checked. The flag C is set in steps S3005 and S3008 in Fig. 69, or in step S3105 or S3108 in Fig. 70, has an initial value "0", is set to be "1" when the sound source method is changed from the TM method to the DPCM method upon an ON event, and is set to be "2" when the sound source method is changed from the DPCM method to the TM method. When the sound source method is left unchanged upon an ON event, the flag C is left at the initial value "0".

If it is determined in step S3201 in the OFF event processing in Fig. 71 that the value of the flag C is "0", since the sound source method is left unchanged in accordance with a key range or a velocity, no special processing is executed, and normal OFF event processing is performed.

If it is determined in step S3201 that the value of the flag C is "1", the sound source method is changed from the TM method to the DPCM method upon an ON event. Thus, TM data in Fig. 49 is set in the tone generation channel area on the RAM 2062 or 3062 (Fig. 35 or 36) where the ON key is assigned to restore the sound source method to the TM method. More specifically, sound source No. data G indicating the TM method is set in the start area of the corresponding tone generation channel area. Subsequently, various parameters corresponding to the presently designated tone color are respectively set in the second and subsequent areas of the corresponding tone generation channel area (S3202).

If it is determined in step S3201 that the value of the flag C is "2", the sound source method is changed from the DPCM method to the TM method. Thus, DPCM data in Fig. 49 is set in the tone generation channel area on the RAM 2062 or 3062 where the ON key is assigned to restore the sound source method from the TM method to the DPCM method. More specifically, sound source method No. data G indicating the DPCM method is set in the start area of the corresponding tone generation channel area. Subsequently, various parameters corresponding to the presently designated tone color are respectively set in the second and subsequent areas of the corresponding tone generation channel area (S3203).

After the above-mentioned operation, the value of the flag C is reset to "0", and the processing in Fig. 71 is completed. Subsequently, normal OFF event processing (not shown) is executed.

Other Embodiments

In the above embodiments of the present invention described above, as shown in Fig. 34, the two CPUs, i.e., the MCPU 1012 and the SCPU 1022 share processing of different tone generation channels. However, the number of CPUs may be one or three or more.

If the control ROMs 2012 and 3012 shown in Figs. 35 and 36, and the external memory 1162 are constituted by, e.g., ROM cards, various sound source methods can be presented to a user by

means of the ROM cards.

Furthermore, the input port 2102 of the MCPU 1012 shown in Fig. 35 can be connected to various other operation units in addition to the instrument operation unit shown in Fig. 45. Thus, various other electronic musical instruments can be realized. In addition, the present invention may be realized as a sound source module for executing only the sound source processing while receiving performance data from another electronic musical instrument.

Various methods of assigning sound source methods to tone generation channels by the function keys 8012 or the keyboard keys 8022 in Fig. 45 including those based on tone colors, tone ranges, and velocities, may be proposed.

In addition to the FM and TM methods, the present invention may be applied to various other modulation methods.

In the modulation method, the above embodiment exemplifies a 4-operator system. However, the number of operators is not limited to this.

In this manner, according to the present invention, a musical tone waveform generation apparatus can be constituted by versatile processors without requiring a special-purpose sound source circuit at all. For this reason, the circuit scale of the overall musical tone waveform generation apparatus can be reduced, and the apparatus can be manufactured in the same manufacturing technique as a conventional microprocessor when the apparatus is constituted by an LSI, thus improving the yield of chips. Therefore, manufacturing cost can be greatly reduced. Note that a musical tone signal output unit can be constituted by a simple latch circuit, resulting in almost no increase in manufacturing cost after the output unit is added.

When the modulation method is required to be changed between a phase modulation method and a frequency modulation method, or when the number of polyphonic channels is required to be changed, a sound source processing program to be stored in a program storage means need only be changed to meet the above requirements. Therefore, development cost of a new musical tone waveform generation apparatus can be greatly decreased, and a new sound source method can be presented to a user by means of, e.g., a ROM card.

In this case, since a data architecture for attaining a data link between a performance data processing program and a sound source processing program via musical tone generation data on a data storage means, and a program architecture for executing the sound source processing program at predetermined time intervals while interrupting the performance data processing program are realized, two processors need not be synchronized, and the programs can be greatly simplified. Thus, complicated sound source processing such as the modulation method can be executed with a sufficient margin.

Furthermore, since a change in processing time depending on the type of modulation method or a selected musical tone generation algorithm in the modulation method can be absorbed by a musical tone signal output means, no complicated timing control program for outputting a musical tone signal to, e.g., a D/A converter can be omitted.

Furthermore, the present invention has, as an architecture of the sound source processing program, a processing architecture for simultaneously executing algorithm processing operations as I/O processing among operator processing operations before or after simultaneous execution of at least one operator processing as a modulation processing unit. For this reason, when one of a plurality of algorithms is selected to execute sound source processing, a plurality of types of algorithm processing portions are prepared, and need only be switched as needed. Therefore, the sound source processing program can be rendered very compact. The small program size can greatly contribute to a compact, low-cost musical tone waveform generation apparatus.

Claims

1. A musical tone waveform generation apparatus characterized by comprising:

storage means (2012, 3021) for storing a plurality of sound source processing programs corresponding to a plurality of types of sound source methods;

musical tone signal generation means (1012, 1022) for generating musical tone signals in arbitrary sound source methods in tone generation channels by executing the plurality of sound source programs stored in said storage means (2012, 3012);and

musical tone signal output means (1072, 1082) for outputting the musical tone signals generated by said musical tone signal generation means (1012, 1022) at predetermined output time intervals.

2. An apparatus according to claim 1, characterized in that said musical tone signal output means (1072, 1082) comprises:

timing signal generating means (2032) for generating a timing signal for each predetermined sampling period;

first latch means (6012) for latching a digital musical tone signal generated by said musical tone signal generation means (1012, 1022) at an outputting timing of the digital musical

tone signal from said musical tone signal generation means (1012, 1022); and

second latch means (6022) for outputting the digital musical tone signal by latching an output signal of said first latch means when the timing signal is generated from said timing signal generating means (2032).

3. A musical tone waveform generation apparatus comprising:

program storage means (2012, 3012) for storing a performance data processing program for processing generation data, and a plurality of sound source processing program corresponding to a plurality of sound source methods for obtaining a musical tone signal;

address control means (2052, 3052) for controlling an address of said program storage means (2012, 3012);

data storage means (2062, 3062) for storing musical tone generation data necessary for generating a musical tone signal by an arbitrary one of said plurality of sound source methods in units of tone generation channels;

arithmetic processing means (2082, 2092, 3082, 3092) for performing a predetermined arithmetic operation;

program execution means (1012, 1022) for executing the performance data processing program or the sound source processing program stored in said program storage means (2012, 3012) while controlling said address control means (2052, 3052), said data storage means (2062, 3062), and said arithmetic processing means (2082, 2092, 3082, 3092),for normally executing the performance data processing program to control musical tone generation data on said data storage means (2062, 3062), for executing the sound source processing program at predetermined time intervals, for executing the performance data processing program again upon completion of the sound source processing program, and for executing time-divisional processing on the basis of musical tone generation data on said data storage means (2062, 3062) upon execution of the sound source processing program so as to generate musical tone signals by the sound source methods assigned to the tone generation channels; and

musical tone signal output means (1072, 1082) for holding the musical tone signals obtained upon execution of the sound source processing programs by said program execution means (1012, 1022), and outputting the held musical tone signals at predetermined output time intervals.

4. A musical tone waveform generation apparatus comprising:

storage means (2011) for storing a sound source processing program;

musical tone signal generation means (1011) for executing the sound source processing program stored in said storage means (2011) to generate a musical tone signal;

pitch designation means (1021) for designating a pitch of the musical tone signal to be generated by said musical tone signal generation means (1011);

tone color determination means (1011) for determining a tone color of the musical tone signal to be generated by said musical tone signal generation means (1011) in accordance with the pitch designated by said pitch designation means (1021);

control means (1011) for controlling said musical tone signal generation means (2011) to generate the musical tone signal having the pitch designated by said pitch designation means and the tone color determined by said tone color determination means (1021); and

musical tone signal output means (2131) for outputting the musical tone signal generated by said musical tone signal generation means (1011) at predetermined time intervals.

5. An apparatus according to claim 4, characterized in said musical tone signal output means (2131) comprises:

timing signal generating means (2031) for generating a timing signal for each predetermined sampling period;

first latch means (3011) for latching a digital musical tone signal generated by said musical tone signal generation means (2031) at an outputting timing of the digital musical tone signal from said musical tone signal generation means; and

second latch means (3021) for outputting the digital muscal tone signal by latching an output signal of said first latch means (3011) when the timing signal is generated from said timing signal generating means (2031).

6. A musical tone waveform generation apparatus comprising:

storage means (2011) for storing a sound source processing program;

musical tone signal generation means (1011) for executing the sound source processing program stored in said storage means (2011) to generate a musical tone signal;

a performance operation member (1021) for instructing said musical tone signal generation means (1011) to generate the musical tone

signal;

tone color determination means (1011) for determining a tone color of the musical tone signal to be generated by said musical tone signal generation means (1011) in accordance with an operation velocity of said performance operation member(1021);

control means (1011) for controlling said musical tone signal generation means (1011) to generate the musical tone signal having the tone color determined by said tone color determination means (1011); and

musical tone signal output means (2131) for outputting the musical tone signal generated by said musical tone signal generation means (1011) at predetermined time intervals.

7. An apparatus according to claim 6, characterized in that said musical tone signal output means (2131) comprises:

timing signal generating means (2031) for generating a timing signal for each predetermined sampling period;

first latch means (3011) for latching a digital musical tone signal generated by said musical tone signal generation means (1011) at an outputting timing of the digital musical tone signal from said musical tone signal generation means (1011); and

second latch means (3021) for outputting the digital musical tone signal by latching an output signal of said first latch means (3011) when the timing signal is generated from said timing signal generating means (2031).

8. A musical tone waveform generation apparatus comprising:

storage means (2011) for storing a sound source processing program;

musical tone signal generation means (1011) for executing the sound source processing program stored in said storage means (2011) to generate a musical tone signal;

output means (1011) for outputting performance data of a plurality of parts constituting a music piece;

tone color determination means (1011) for determining a tone color of the musical tone signal to be generated by said musical tone signal generation means (1011) in accordance with one of the plurality of parts to which the performance data output from said output means (1011) belongs;

control means (1011) for controlling said musical tone generation means (1011) to generate the musical tone signal having the tone color determined by said tone color determination means (1011); and

musical tone signal output means (2131) for outputting the musical tone signal generated by said musical tone signal generatiuon means (1011) at predetermined time intervals.

9. An apparatus according to claim 8, characterized in that said musical tone signal output means (2131) comprises:

timing signal generating means (2031) for generating a timing signal for each predetermined sampling period;

first latch means (3011) for latching a digital musical tone signal generated by said musical tone signal generation means (1011) at an outputting timing of the digital musical tone signal from said musical tone signal generation means (1011); and

second latch means (3021) for outputting the digital musical tone signal by latching an output signal of said first latch means (3011) when the timing signal is generated from said timing signal generating means (2031).

10. A musical tone waveform generation apparatus characterized by comprising:

program storage means (2011) for storing a performance data processing program for processing performance data, and a sound source processing program for obtaining a musical tone signal;

address control means (2051) for controlling an address of said program storage means;

split point designation means (15011, 20011) for causing a player to designate a split point to divide a range of a performance data value into a plurality of ranges;

tone color designation means (15021, 20021) for designating tone colors of the plurality of ranges having the split point designated by said split point designation means (15011, 20011) as a boundary;

data storage means (2061) for storing musical tone generation data necessary for generating the musical tone signal in correspondence with a plurality of tone colors;

arithmetic processing means (2081, 2091) for processing data;

program execution means for executing the performance data processing program and the sound source processing program stored in said program storage means while controlling said address control means (205), said data storage means (2061), and said arithmetic processing means (2081, 2091), for normally executing the performance data processing program to control musical tone generation data stored in said data storage means (2061),

for executing the sound source processing program at predetermined time intervals, for executing the performance data processing program again upon completion of the sound source processing program, and for generating, upon execution of the sound source processing program, the musical tone signal on the basis of the musical tone generation data on said data storage means (2061) corresponding to the tone color designated by said tone color designation means (15021, 20021) in correspondence with the range which has the split point designated by said split point designation means (15011, 20011) as a boundary, and to which the performance data value belongs; and

musical tone signal output means (2131) for holding the musical tone signals in units of tone generation operations obtained upon execution of the sound source processing program by said program execution means, and outputting the held musical tone signals at predetermined output time intervals.

11. An apparatus according to claim 10, characterized in that the predetermined performance data is data indicating a pitch.

12. An apparatus according to claim 10, characterized in that the predetermined performance data is data indicating a touch of an operation member in a performance operation.

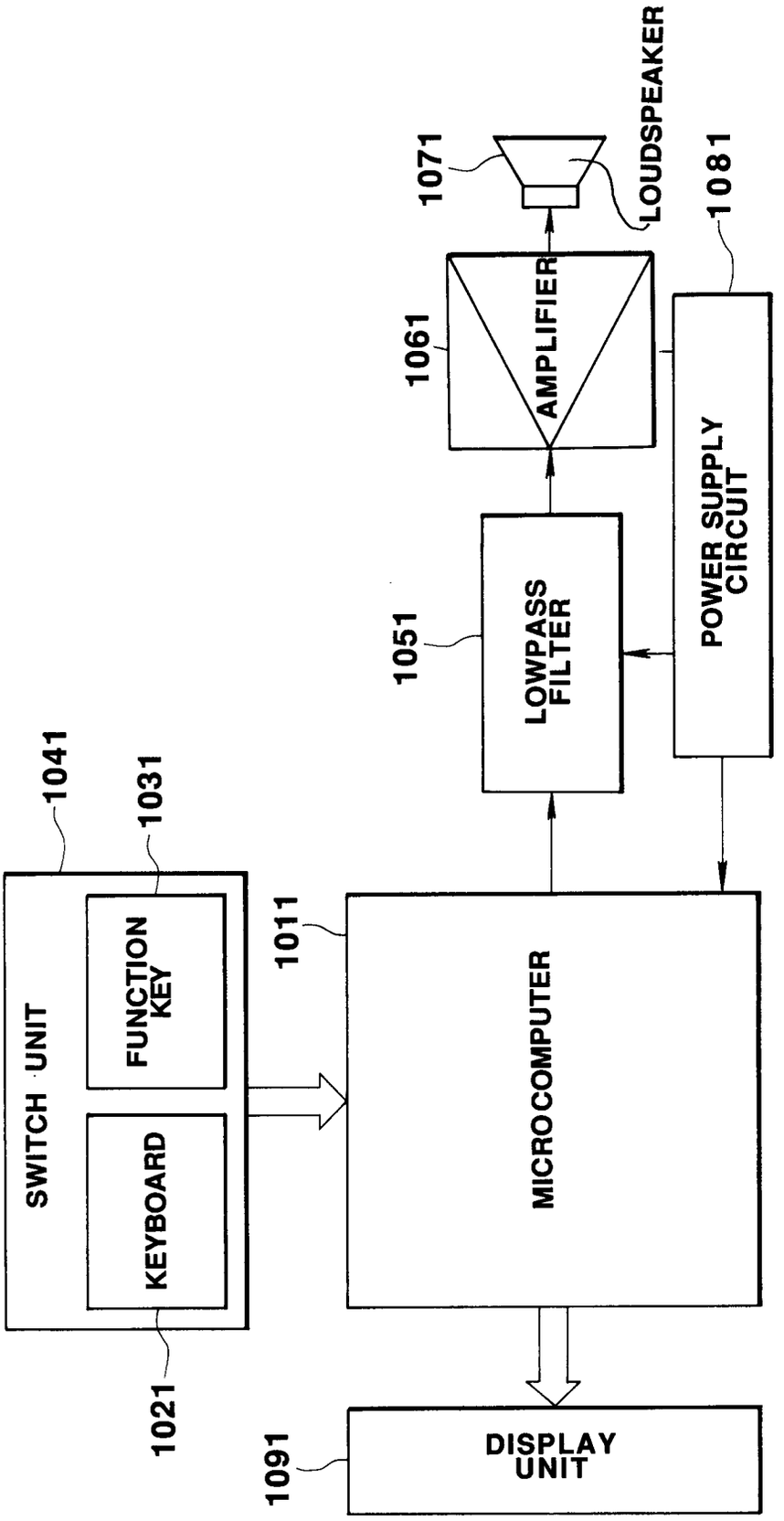
1071
LOUDSPEAKER
1081
1061
AMPLIFIER
POWER SUPPLY CIRCUIT
1051
LOWPASS FILTER
1041
1031
1011
SWITCH UNIT
FUNCTION KEY
KEYBOARD
MICROCOMPUTER
1021
1091
DISPLAY UNIT

FIG. 1

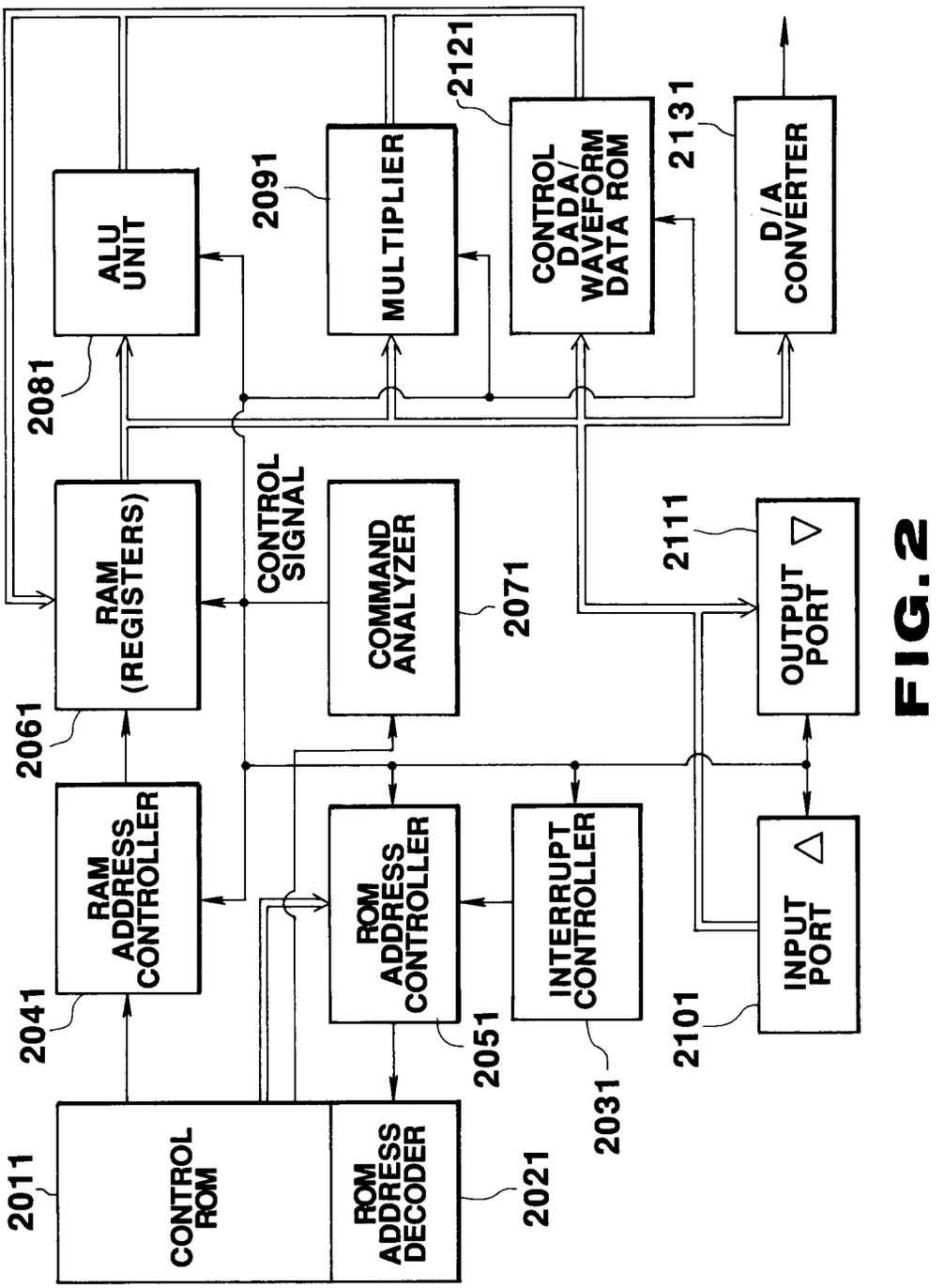
CONTROL ROM
ROM ADDRESS DECODER
RAM ADDRESS CONTROLLER
ROM ADDRESS CONTROLLER
INTERRUPT CONTROLLER
INPUT PORT
OUTPUT PORT
RAM (REGISTERS)
COMMAND ANALYZER
CONTROL SIGNAL
ALU UNIT
MULTIPLIER
CONTROL DADA/ WAVEFORM DATA ROM
D/A CONVERTER
2011
2021
2041
2051
2031
2101
2061
2071
2111
2081
2091
2121
2131

FIG. 2

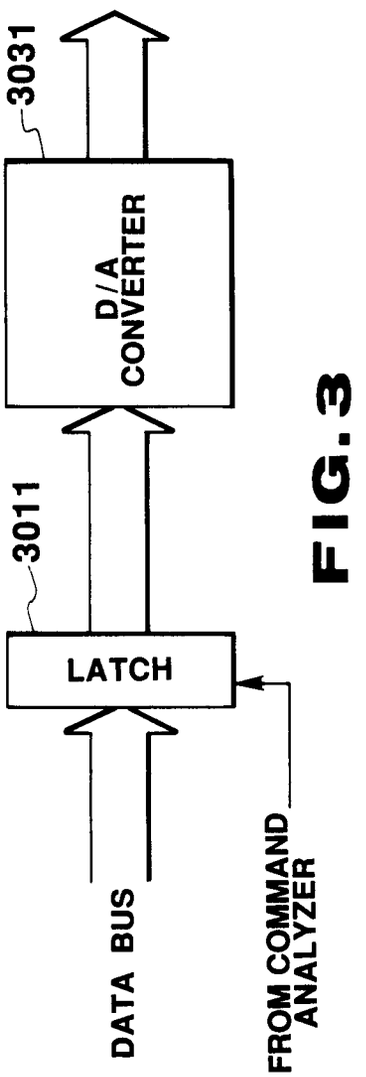
3031
D/A CONVERTER
3011
LATCH
DATA BUS
FROM COMMAND ANALYZER

FIG. 3

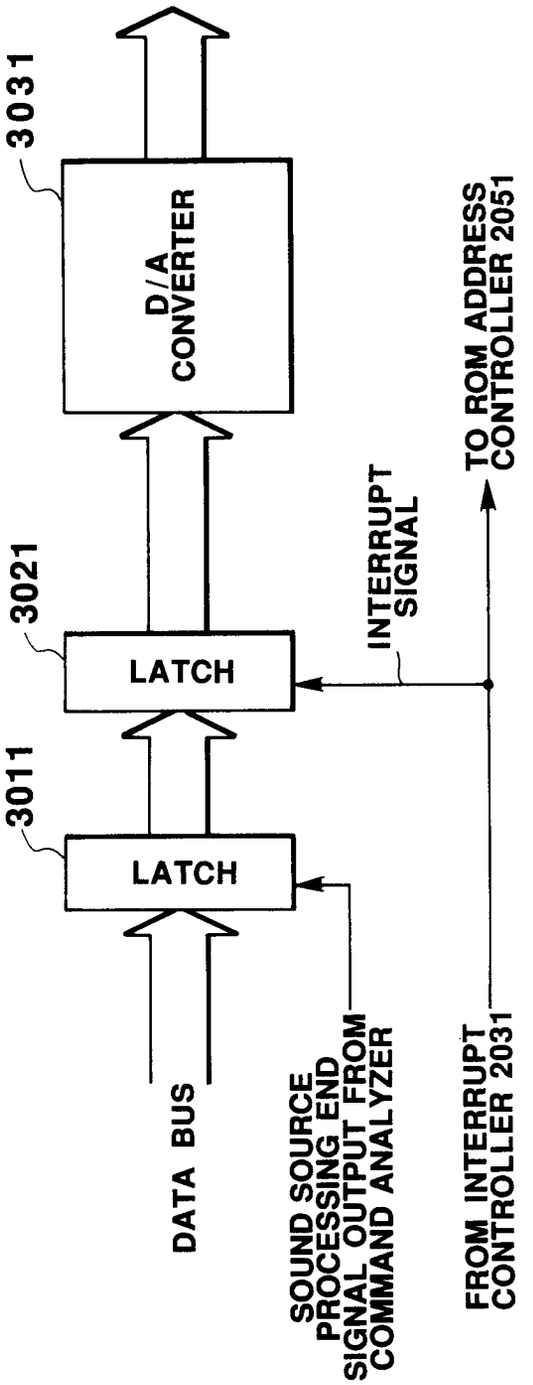
3031
D/A CONVERTER
3021
LATCH
INTERRUPT SIGNAL
TO ROM ADDRESS CONTROLLER 2051
3011
LATCH
DATA BUS
SOUND SOURCE PROCESSING END SIGNAL OUTPUT FROM COMMAND ANALYZER
FROM INTERRUPT CONTROLLER 2031

FIG. 4

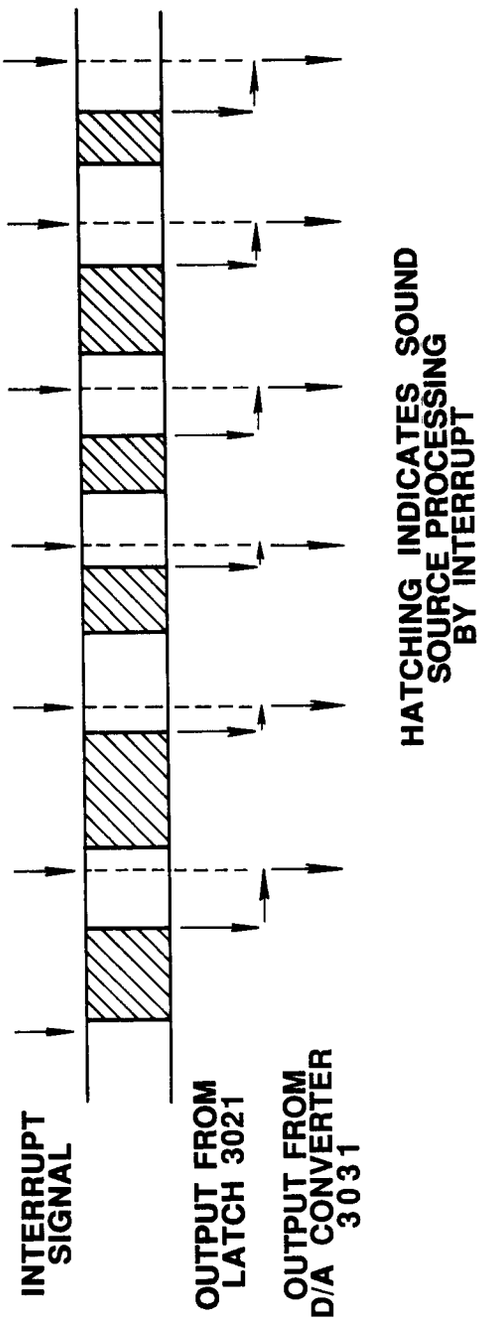
INTERRUPT SIGNAL
OUTPUT FROM LATCH 3021
OUTPUT FROM D/A CONVERTER 3031
HATCHING INDICATES SOUND SOURCE PROCESSING BY INTERRUPT

FIG.5

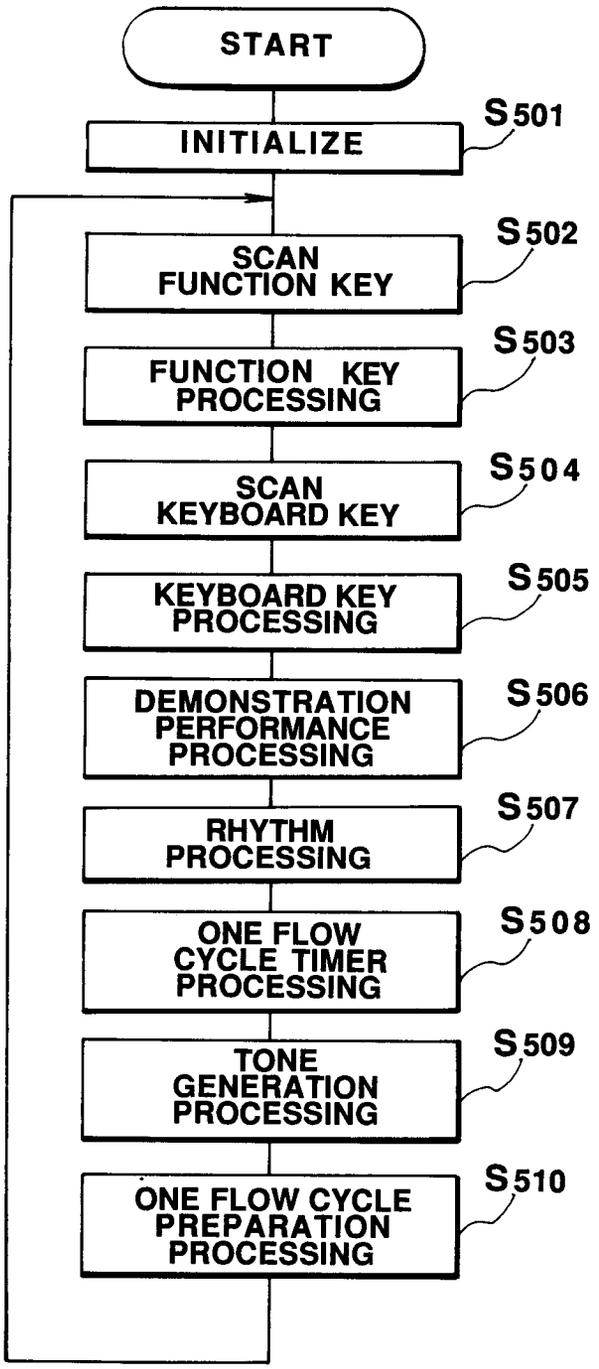
START
INITIALIZE
S501
SCAN FUNCTION KEY
S502
FUNCTION KEY PROCESSING
S503
SCAN KEYBOARD KEY
S504
KEYBOARD KEY PROCESSING
S505
DEMONSTRATION PERFORMANCE PROCESSING
S506
RHYTHM PROCESSING
S507
ONE FLOW CYCLE TIMER PROCESSING
S508
TONE GENERATION PROCESSING
S509
ONE FLOW CYCLE PREPARATION PROCESSING
S510

FIG.6

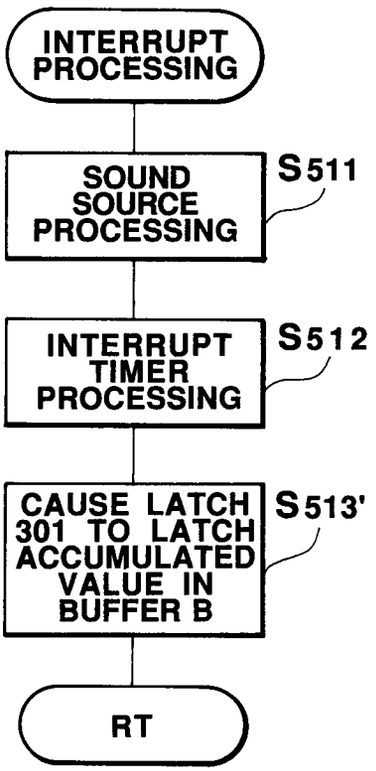
INTERRUPT PROCESSING
SOUND SOURCE PROCESSING
S511
INTERRUPT TIMER PROCESSING
S512
CAUSE LATCH 301 TO LATCH ACCUMULATED VALUE IN BUFFER B
S513'
RT

FIG.7

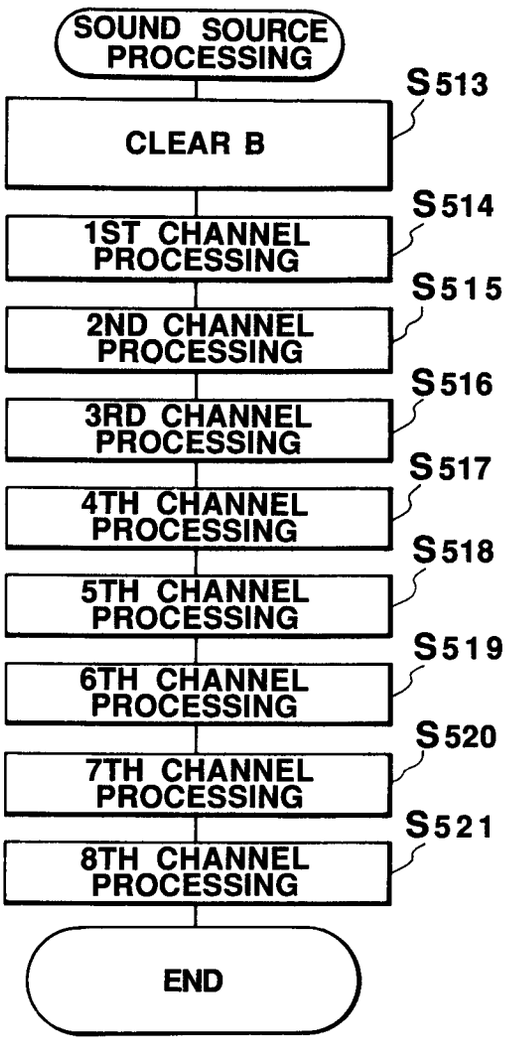
SOUND SOURCE PROCESSING
S513
CLEAR B
S514
1ST CHANNEL PROCESSING
S515
2ND CHANNEL PROCESSING
S516
3RD CHANNEL PROCESSING
S517
4TH CHANNEL PROCESSING
S518
5TH CHANNEL PROCESSING
S519
6TH CHANNEL PROCESSING
S520
7TH CHANNEL PROCESSING
S521
8TH CHANNEL PROCESSING
END

FIG.8

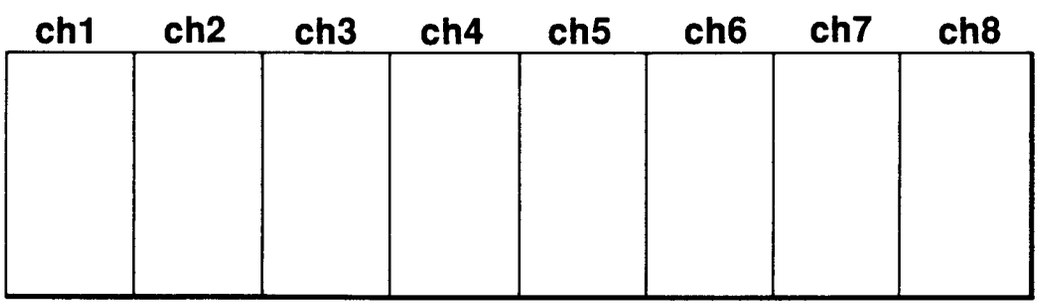
ch1
ch2
ch3
ch4
ch5
ch6
ch7
ch8

FIG.10

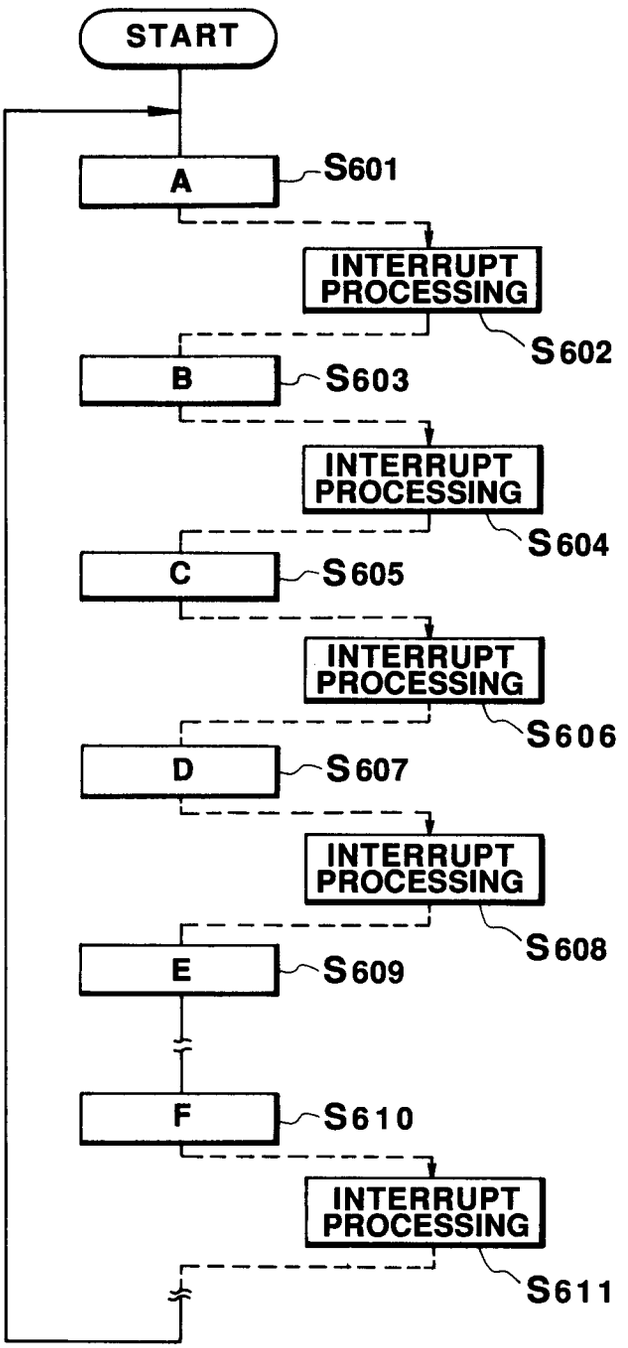
START
A
S601
INTERRUPT PROCESSING
S602
B
S603
INTERRUPT PROCESSING
S604
C
S605
INTERRUPT PROCESSING
S606
D
S607
INTERRUPT PROCESSING
S608
E
S609
F
S610
INTERRUPT PROCESSING
S611

FIG. 9

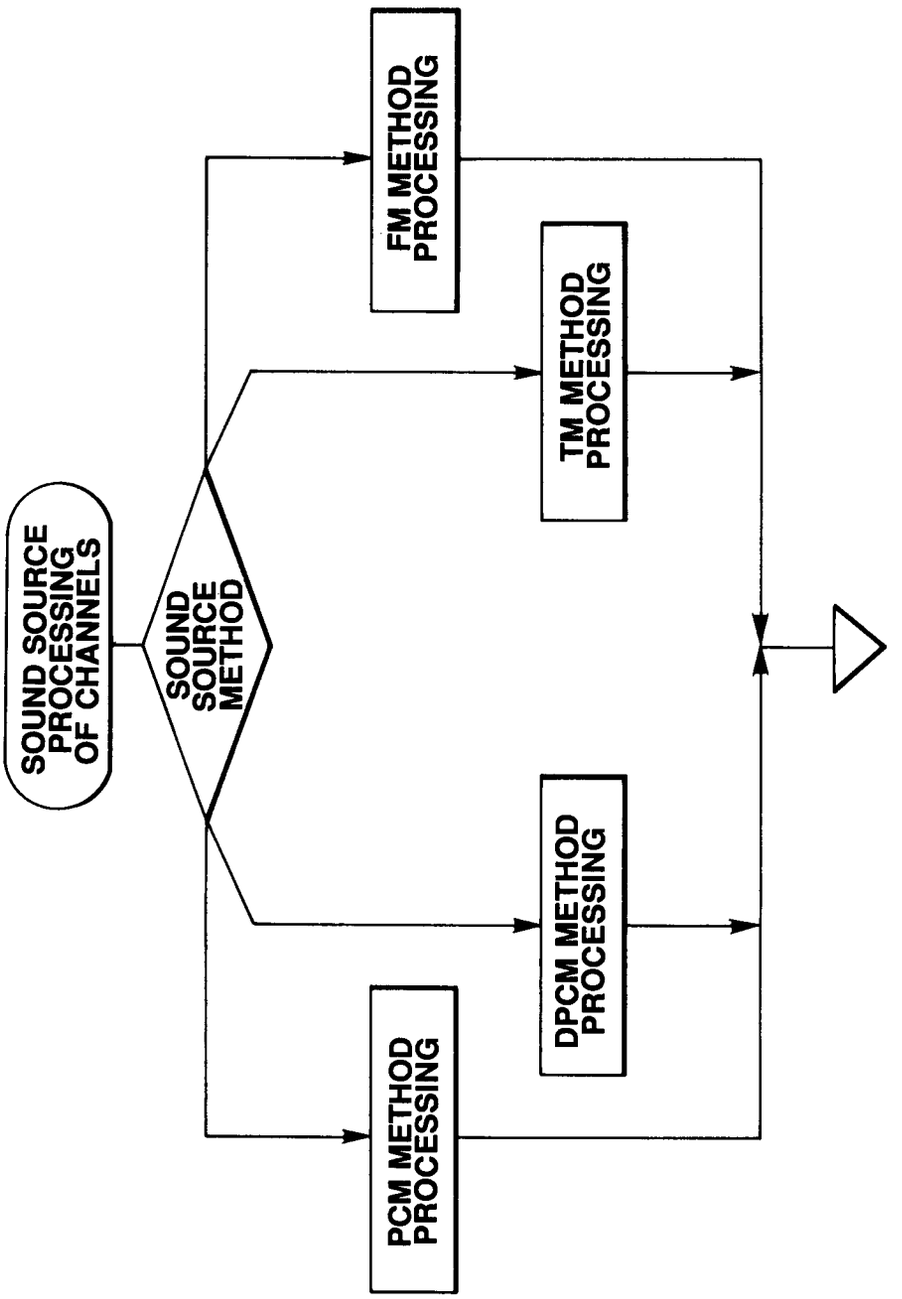
SOUND SOURCE PROCESSING OF CHANNELS
SOUND SOURCE METHOD
FM METHOD PROCESSING
TM METHOD PROCESSING
DPCM METHOD PROCESSING
PCM METHOD PROCESSING

FIG.11

PCM		DPCM		FM		TM	
S	SOUND SOURCE METHOD NO.	S	SOUND SOURCE METHOD NO.	S	SOUND SOURCE METHOD NO.	S	SOUND SOURCE METHOD NO.
AI	CURRENT ADDRESS INTEGRAL PART	AI	CURRENT ADDRESS INTEGRAL PART	A1	CURRENT ADDRESS (OP1)	A1	CURRENT ADDRESS (OP1)
AF	CURRENT ADDRESS DECIMAL PART	AF	CURRENT ADDRESS DECIMAL PART	A2	CURRENT ADDRESS (OP2)	A2	CURRENT ADDRESS (OP2)
AE	END ADDRESS INTEGRAL PART	AE	END ADDRESS INTEGRAL PART				
AL	LOOP ADDRESS INTEGRAL PART	AL	LOOP ADDRESS INTEGRAL PART				
PF	PITCH DATA DECIMAL PART	PF	PITCH DATA DECIMAL PART	P1	PITCH DATA (OP1)	P1	PITCH DATA (OP1)
PI	PITCH DATA INTEGRAL PART	PI	PITCH DATA INTEGRAL PART	P2	PITCH DATA (OP2)	P2	PITCH DATA (OP2)
XP	IMMEDIATELY PRECEDING SAMPLE DATA	XP	IMMEDIATELY PRECEDING SAMPLE DATA				
XN	NEXT SAMPLE DATA	OLD AI	CURRENT ADDRESS INTEGRAL PART BEFORE CHANGE				
D	DIFFERENCE BETWEEN ADJACENT SAMPLE DATA	D	DIFFERENCE BETWEEN ADJACENT SAMPLE DATA				
E	ENVELOPE VALUE	E	ENVELOPE VALUE	E1	ENVELOPE (OP1)	E1	ENVELOPE (OP1)
		XPL	SAMPLE DATA OF AL	E2	ENVELOPE (OP2)	E2	ENVELOPE (OP2)
				ML2	MODULATION LEVEL (OP2)	ML2	MODULATION LEVEL (OP2)
				MO2	MODULATION OUTPUT (OP2)	MO2	MODULATION OUTPUT (OP2)
				FL2	FEEDBACK LEVEL (OP2)	FL2	FEEDBACK LEVEL (OP2)
				FO2	FEEDBACK OUTPUT (OP2)	FO2	FEEDBACK OUTPUT (OP2)
O	OUTPUT	O	OUTPUT	O1	OP1 OUTPUT	O1	OP1 OUTPUT
				O2	OP2 OUTPUT	O2	OP2 OUTPUT

FIG.12

EP 0 463 411 A2

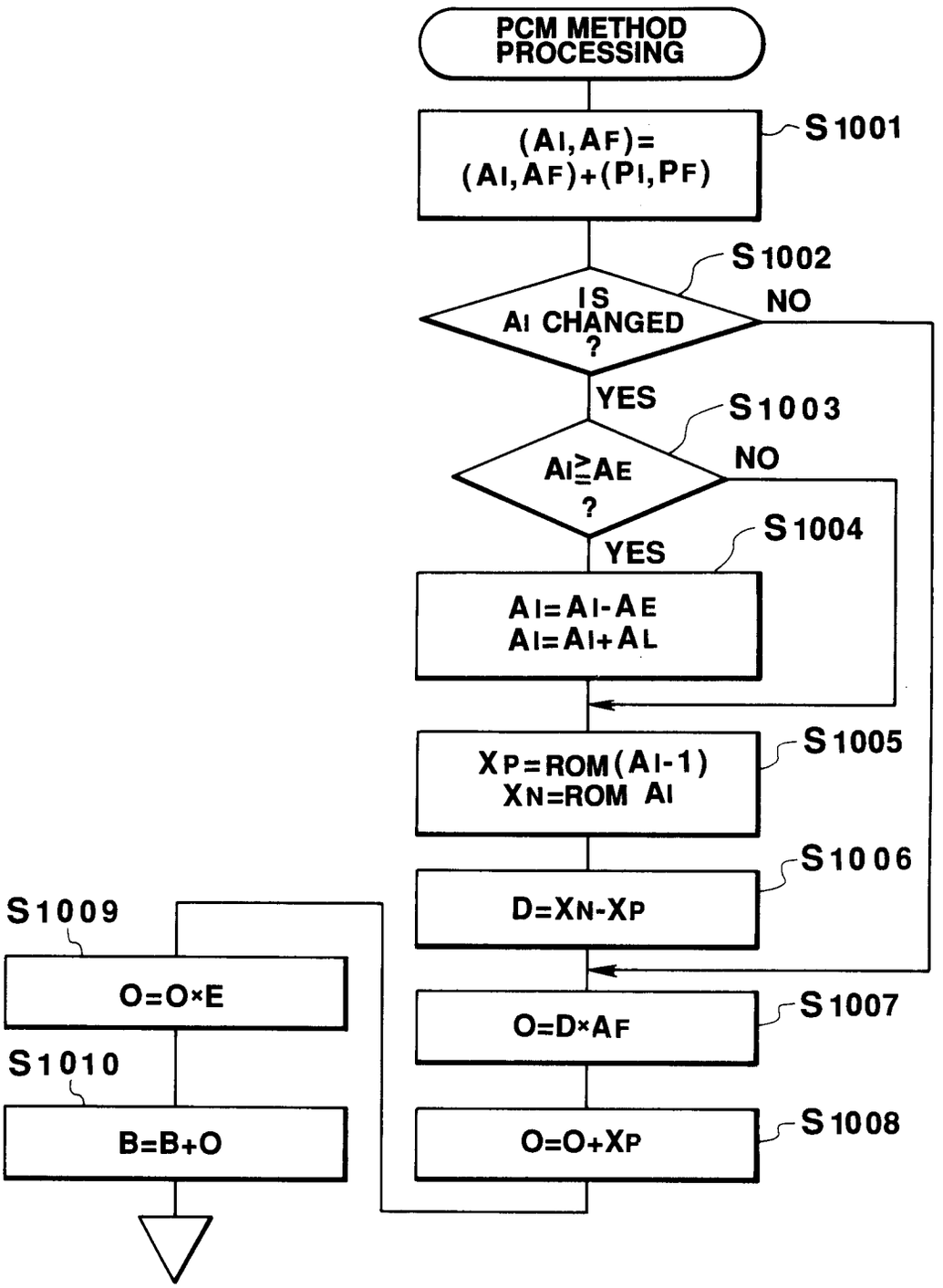
PCM METHOD PROCESSING
(AI,AF)= (AI,AF)+(PI,PF)
S1001
S1002
IS AI CHANGED ?
NO
YES
S1003
AI≧AE ?
NO
YES
S1004
AI=AI-AE
AI=AI+AL
XP=ROM(AI-1)
XN=ROM AI
S1005
D=XN-XP
S1006
O=D×AF
S1007
O=O+XP
S1008
S1009
O=O×E
S1010
B=B+O

FIG.13

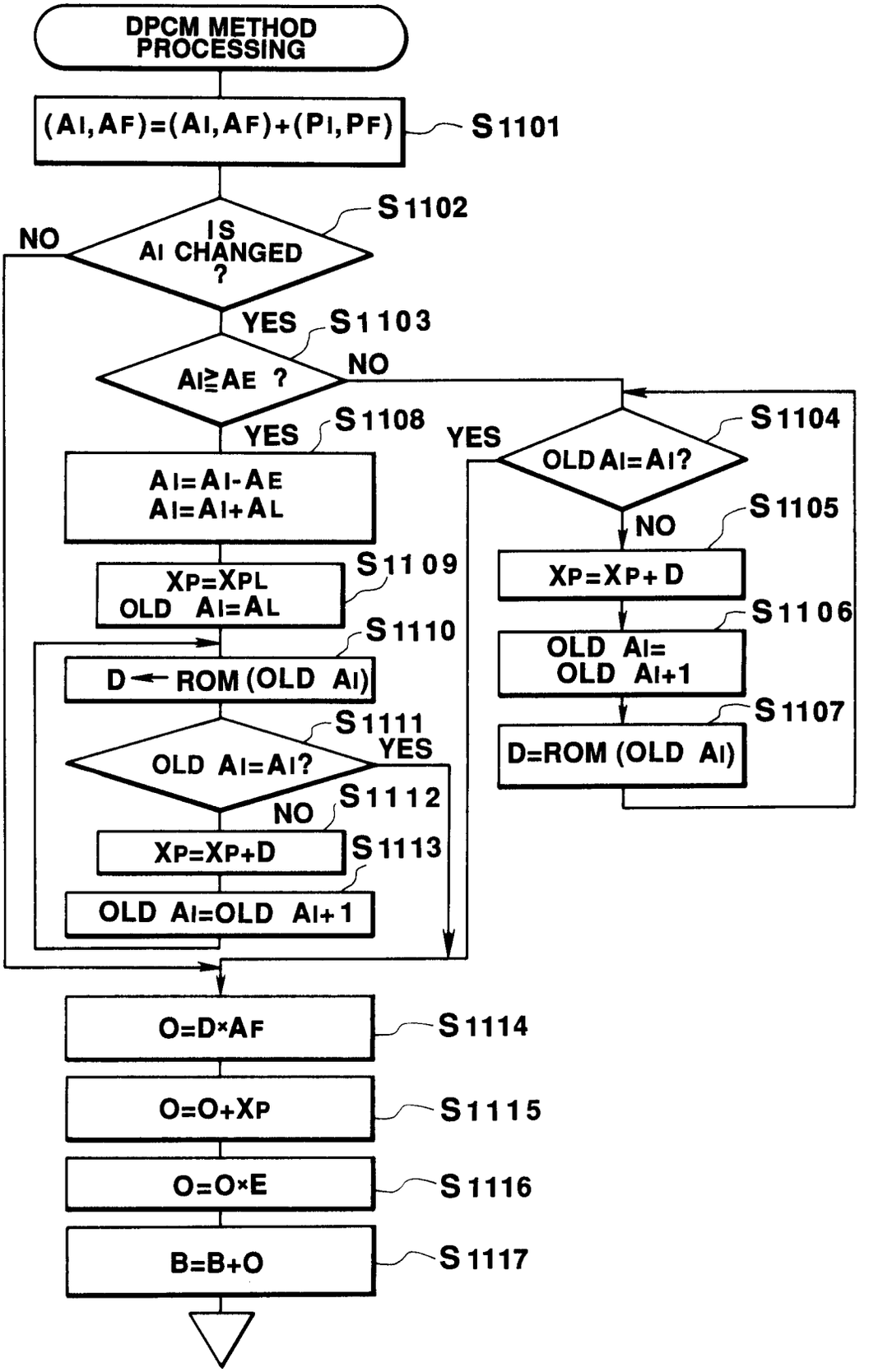
DPCM METHOD PROCESSING
(AI,AF)=(AI,AF)+(PI,PF)
S1101
IS AI CHANGED ?
S1102
NO
YES
AI≧AE ?
S1103
NO
YES
AI=AI-AE
AI=AI+AL
S1108
XP=XPL
OLD AI=AL
S1109
D ← ROM (OLD AI)
S1110
OLD AI=AI?
S1111
YES
NO
XP=XP+D
S1112
OLD AI=OLD AI+1
S1113
O=D×AF
S1114
O=O+XP
S1115
O=O×E
S1116
B=B+O
S1117
OLD AI=AI?
S1104
YES
NO
XP=XP+D
S1105
OLD AI=
OLD AI+1
S1106
D=ROM (OLD AI)
S1107

FIG.14

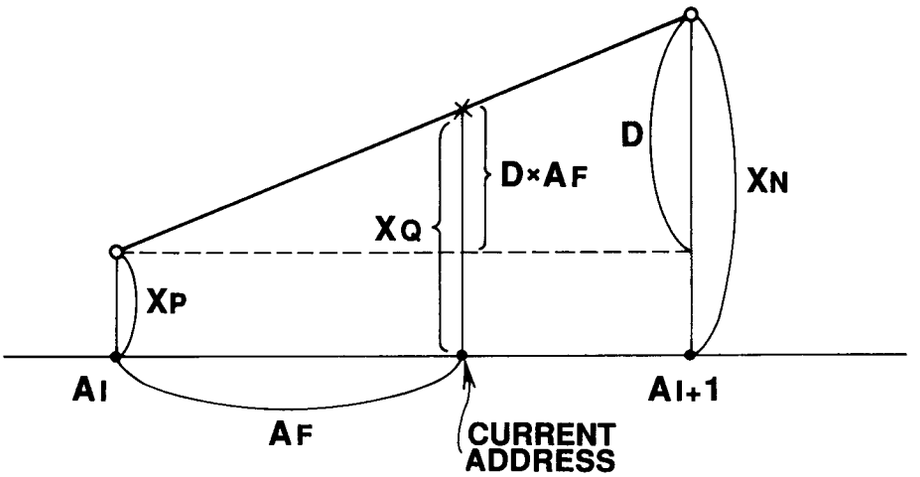
D
D×AF
XN
XQ
XP
AI
AI+1
AF
CURRENT
ADDRESS

FIG.15

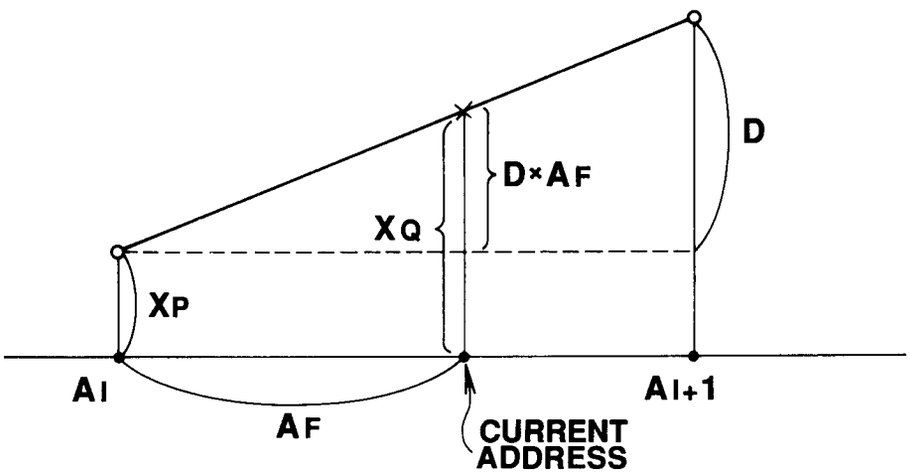
D
D×AF
XQ
XP
AI
AI+1
AF
CURRENT
ADDRESS

FIG.16

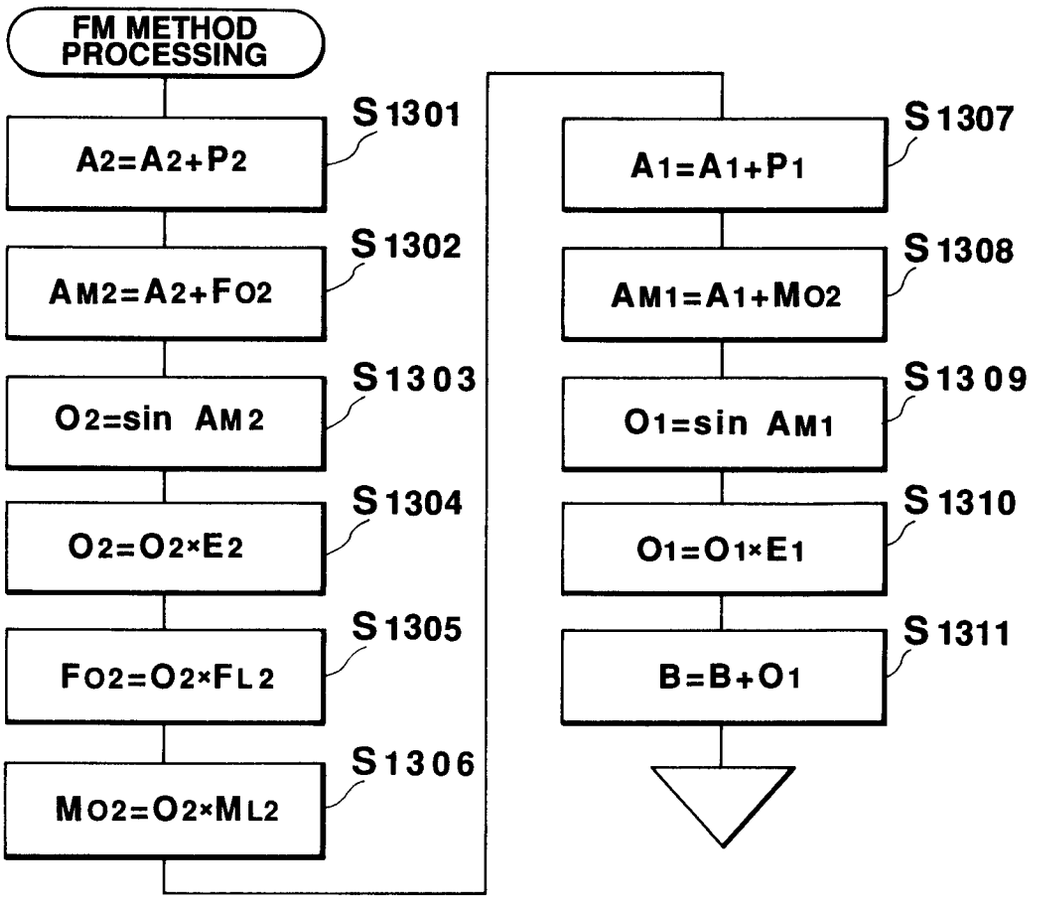
FM METHOD PROCESSING
S1301
A2=A2+P2
S1302
AM2=A2+FO2
S1303
O2=sin AM2
S1304
O2=O2×E2
S1305
FO2=O2×FL2
S1306
MO2=O2×ML2
S1307
A1=A1+P1
S1308
AM1=A1+MO2
S1309
O1=sin AM1
S1310
O1=O1×E1
S1311
B=B+O1

FIG.17

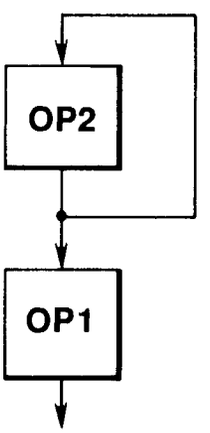
OP2
OP1

FIG.18

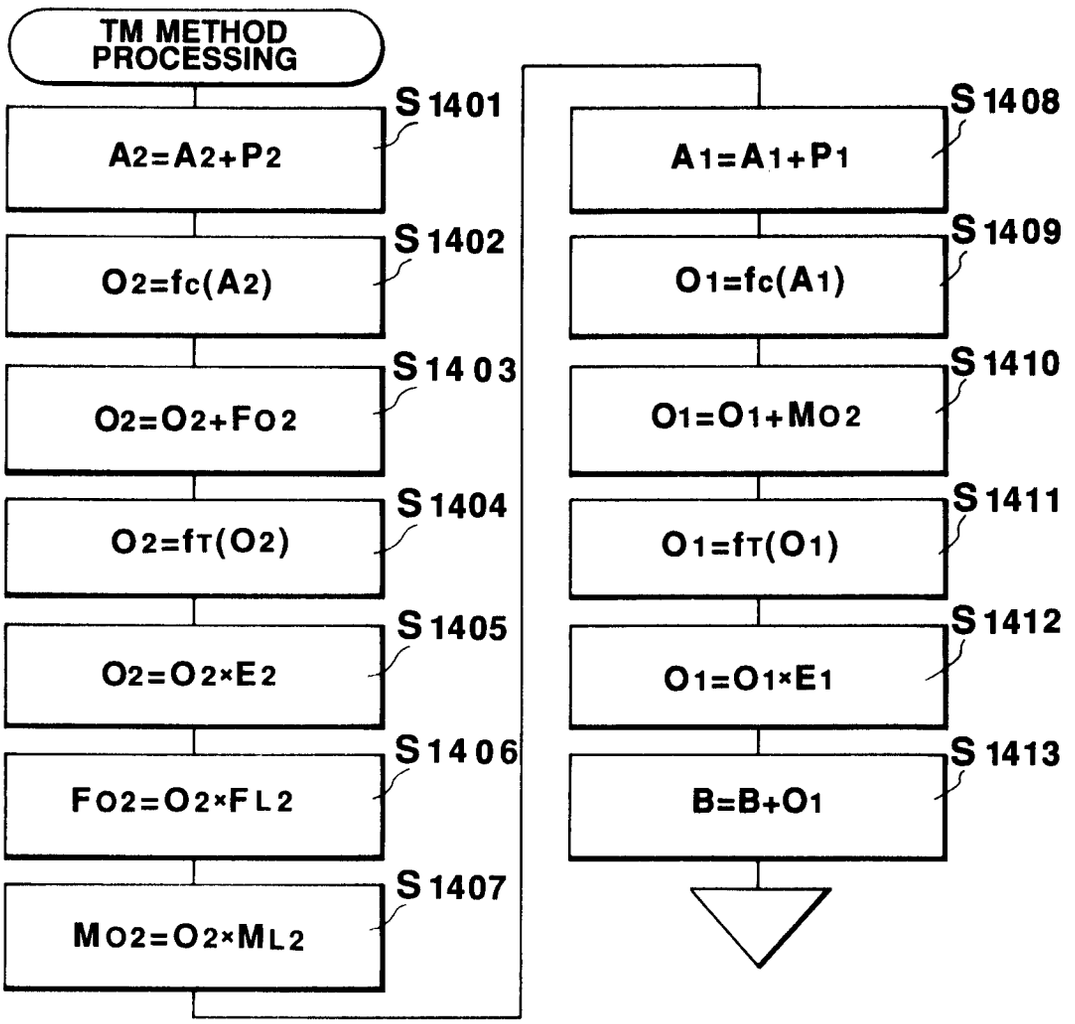
TM METHOD PROCESSING
S1401
A2=A2+P2
S1402
O2=fc(A2)
S1403
O2=O2+FO2
S1404
O2=fT(O2)
S1405
O2=O2×E2
S1406
FO2=O2×FL2
S1407
MO2=O2×ML2
S1408
A1=A1+P1
S1409
O1=fc(A1)
S1410
O1=O1+MO2
S1411
O1=fT(O1)
S1412
O1=O1×E1
S1413
B=B+O1

FIG.19

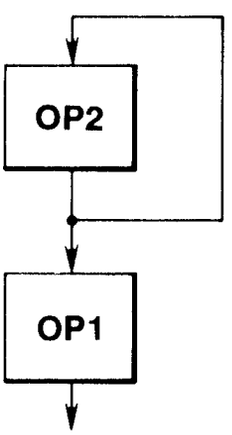
OP2
OP1

FIG.20

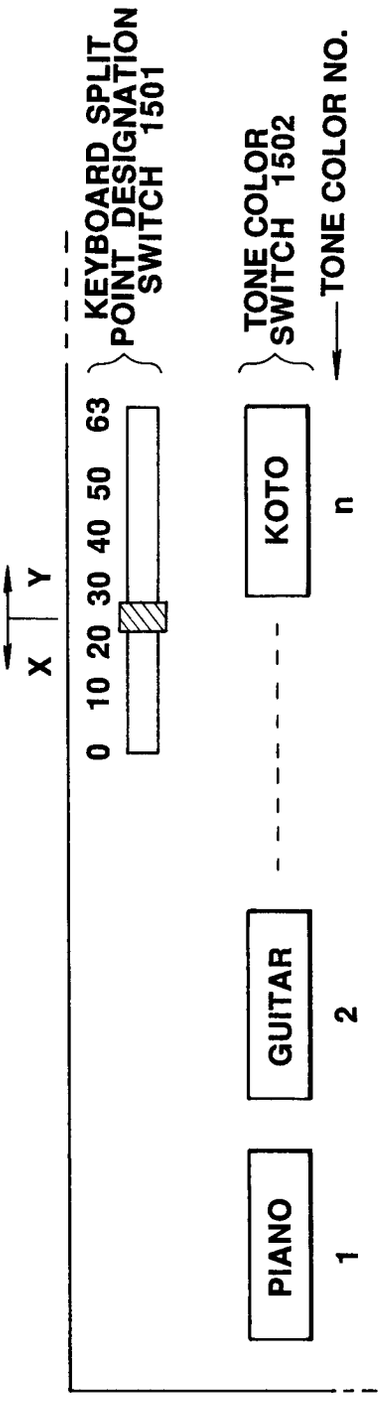
KEYBOARD SPLIT POINT DESIGNATION SWITCH 1501
TONE COLOR SWITCH 1502
TONE COLOR NO.
X
Y
0 10 20 30 40 50 63
PIANO
GUITAR
KOTO
1
2
n

FIG. 21

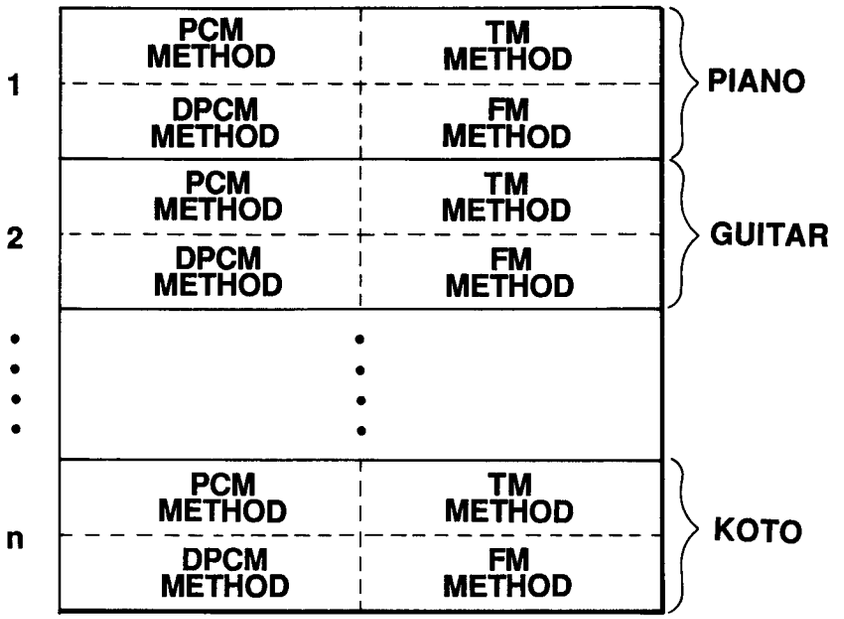

FIG. 22

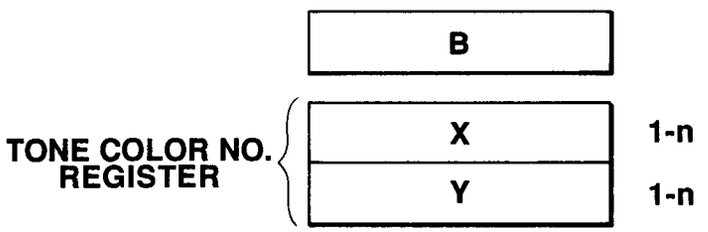

FIG. 23

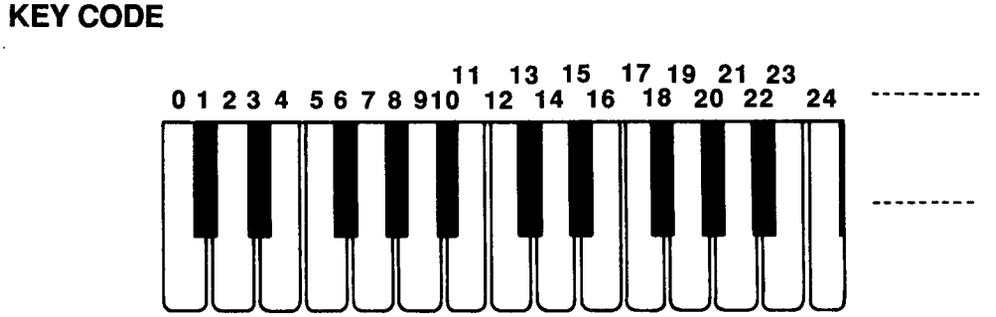

FIG. 24

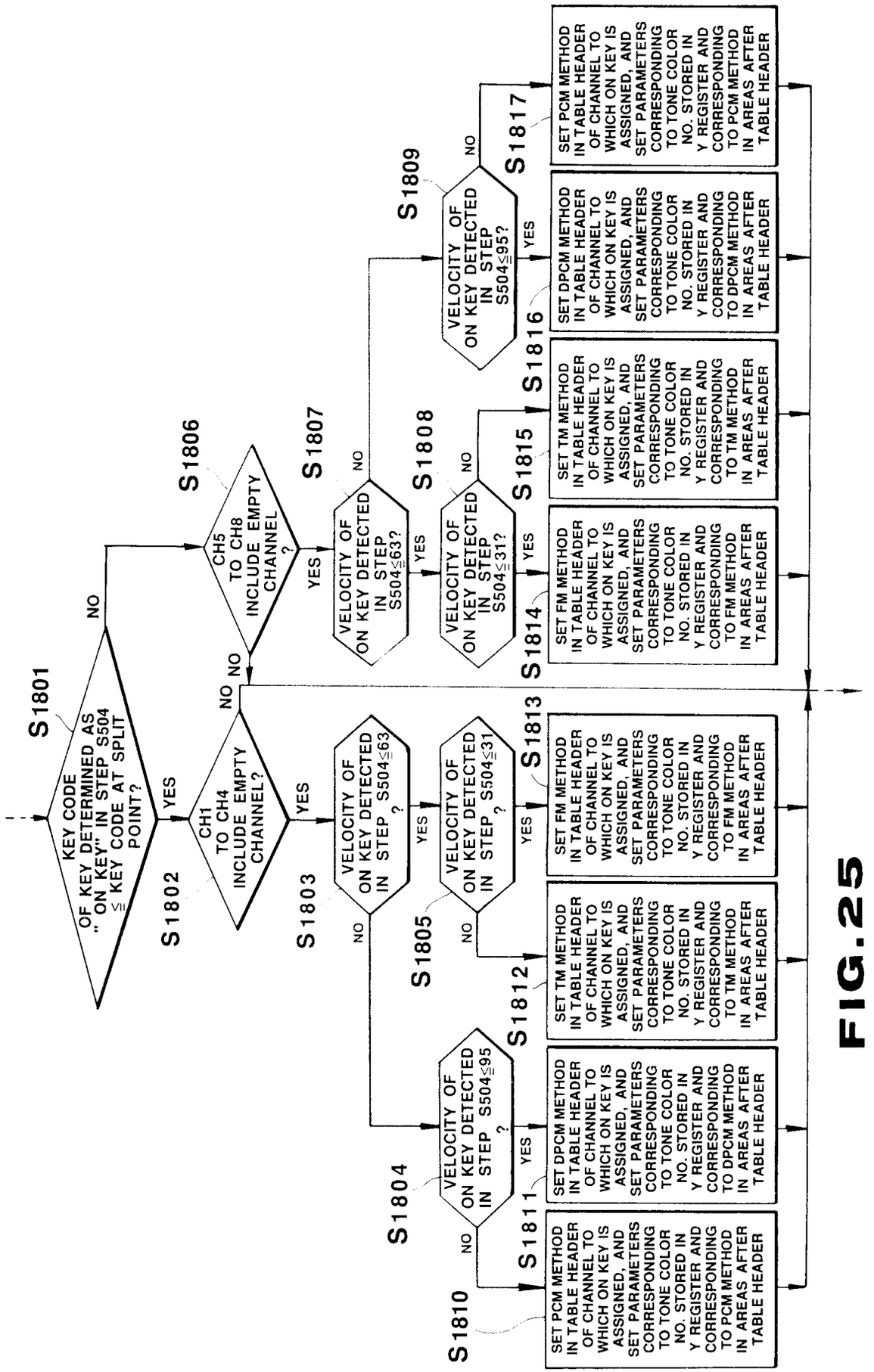
S1801
KEY CODE OF KEY DETERMINED AS "ON KEY" IN STEP S504 ≦ KEY CODE AT SPLIT POINT?
NO
YES
S1802
CH1 TO CH4 INCLUDE EMPTY CHANNEL?
NO
YES
S1803
VELOCITY OF ON KEY DETECTED IN STEP S504≦63?
NO
YES
S1804
VELOCITY OF ON KEY DETECTED IN STEP S504≦95?
NO
YES
S1805
VELOCITY OF ON KEY DETECTED IN STEP S504≦31?
NO
YES
S1806
CH5 TO CH8 INCLUDE EMPTY CHANNEL?
NO
YES
S1807
VELOCITY OF ON KEY DETECTED IN STEP S504≦63?
NO
YES
S1808
VELOCITY OF ON KEY DETECTED IN STEP S504≦31?
NO
YES
S1809
VELOCITY OF ON KEY DETECTED IN STEP S504≦95?
NO
YES
S1810
SET PCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO PCM METHOD IN AREAS AFTER TABLE HEADER
S1811
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER
S1812
SET TM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO TM METHOD IN AREAS AFTER TABLE HEADER
S1813
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
S1814
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
S1815
SET TM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO TM METHOD IN AREAS AFTER TABLE HEADER
S1816
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER
S1817
SET PCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO PCM METHOD IN AREAS AFTER TABLE HEADER

FIG.25

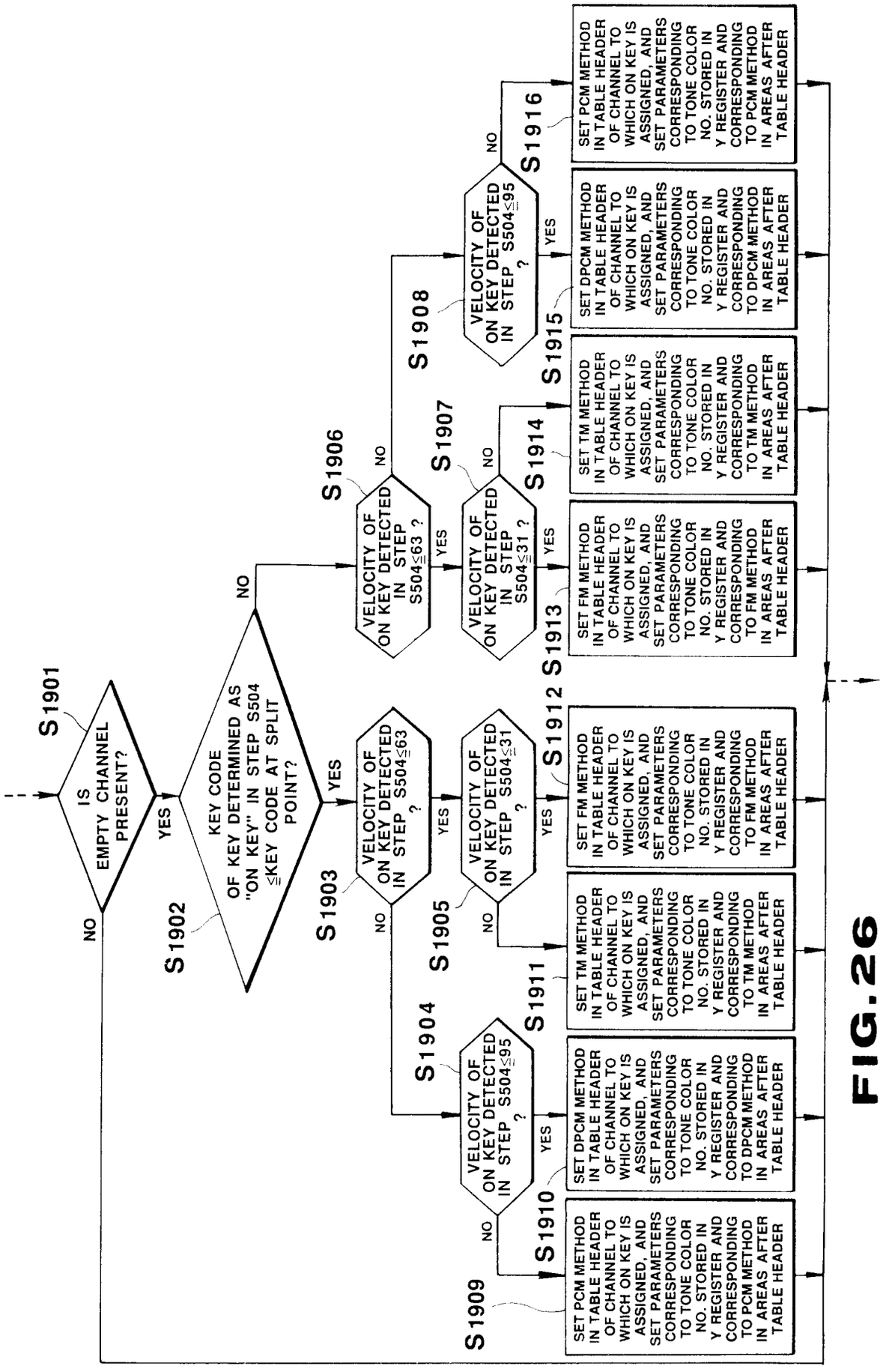
S1901
IS EMPTY CHANNEL PRESENT?
NO
YES
S1902
KEY CODE OF KEY DETERMINED AS "ON KEY" IN STEP S504 ≦KEY CODE AT SPLIT POINT?
NO
YES
S1903
VELOCITY OF ON KEY DETECTED IN STEP S504≦63 ?
NO
YES
S1904
VELOCITY OF ON KEY DETECTED IN STEP S504≦95 ?
NO
YES
S1905
VELOCITY OF ON KEY DETECTED IN STEP S504≦31 ?
NO
YES
S1906
VELOCITY OF ON KEY DETECTED IN STEP S504≦63 ?
NO
YES
S1907
VELOCITY OF ON KEY DETECTED IN STEP S504≦31 ?
NO
YES
S1908
VELOCITY OF ON KEY DETECTED IN STEP S504≦95 ?
NO
YES
S1909
SET PCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO PCM METHOD IN AREAS AFTER TABLE HEADER
S1910
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER
S1911
SET TM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO TM METHOD IN AREAS AFTER TABLE HEADER
S1912
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
S1913
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
S1914
SET TM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO TM METHOD IN AREAS AFTER TABLE HEADER
S1915
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER
S1916
SET PCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO PCM METHOD IN AREAS AFTER TABLE HEADER

FIG.26

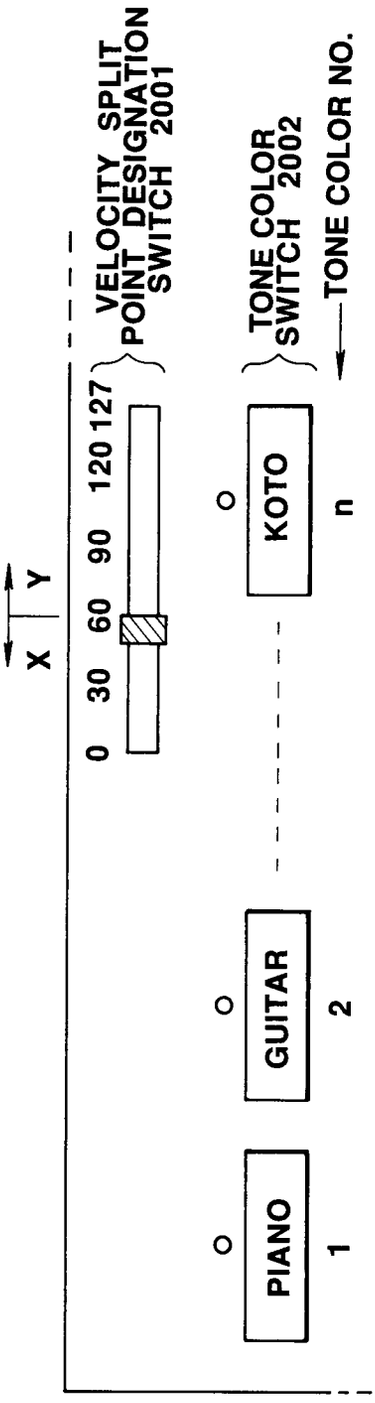
VELOCITY SPLIT POINT DESIGNATION SWITCH 2001
TONE COLOR SWITCH 2002
TONE COLOR NO.
X
Y
0
30
60
90
120
127
PIANO
1
GUITAR
2
KOTO
n

FIG. 27

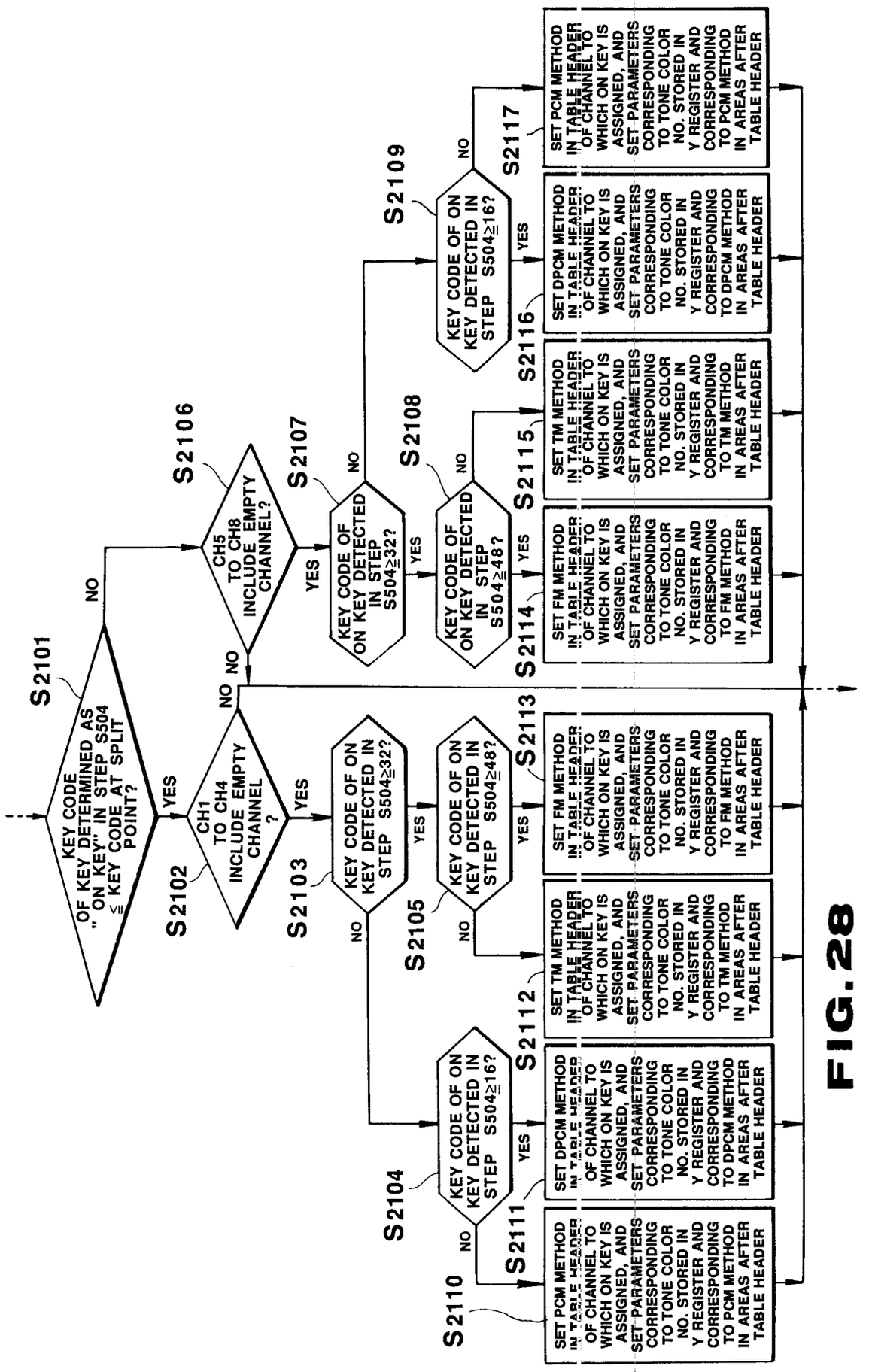
S2101
KEY CODE OF KEY DETERMINED AS " ON KEY" IN STEP S504 ≦ KEY CODE AT SPLIT POINT?
YES
NO
S2102
CH1 TO CH4 INCLUDE EMPTY CHANNEL ?
YES
NO
S2103
KEY CODE OF ON KEY DETECTED IN STEP S504≧32?
YES
NO
S2104
KEY CODE OF ON KEY DETECTED IN STEP S504≧16?
YES
NO
S2105
KEY CODE OF ON KEY DETECTED IN STEP S504≧48?
YES
NO
S2106
CH5 TO CH8 INCLUDE EMPTY CHANNEL?
YES
NO
S2107
KEY CODE OF ON KEY DETECTED IN STEP S504≧32?
YES
NO
S2108
KEY CODE OF ON KEY DETECTED IN STEP S504≧48?
YES
NO
S2109
KEY CODE OF ON KEY DETECTED IN STEP S504≧16?
YES
NO
S2110
SET PCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO PCM METHOD IN AREAS AFTER TABLE HEADER
S2111
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER
S2112
SET TM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO TM METHOD IN AREAS AFTER TABLE HEADER
S2113
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
S2114
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
S2115
SET TM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO TM METHOD IN AREAS AFTER TABLE HEADER
S2116
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER
S2117
SET PCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO PCM METHOD IN AREAS AFTER TABLE HEADER

FIG. 28

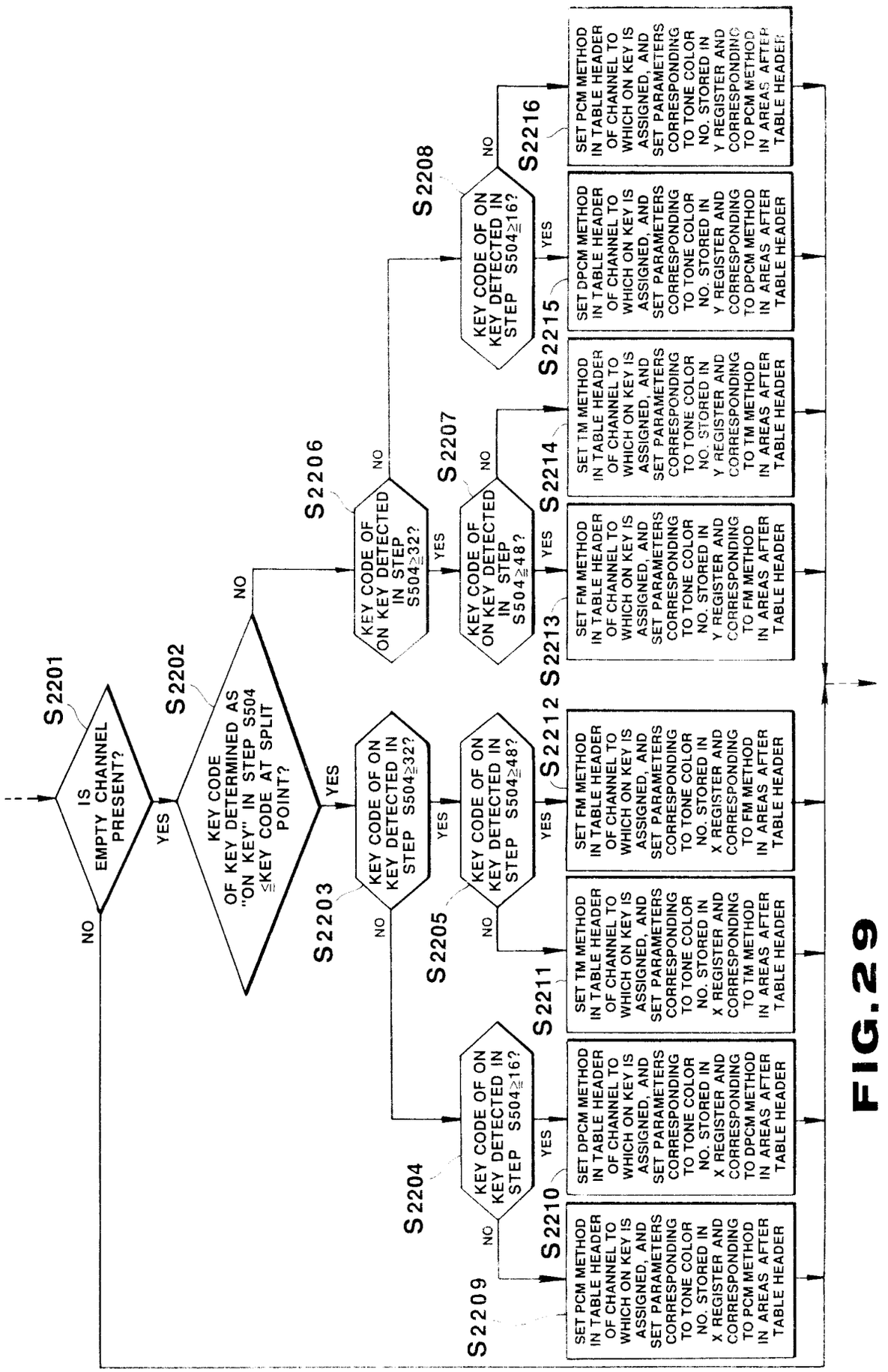
S2201
IS EMPTY CHANNEL PRESENT?
NO
YES
S2202
KEY CODE OF KEY DETERMINED AS "ON KEY" IN STEP S504 ≦KEY CODE AT SPLIT POINT?
NO
YES
S2203
KEY CODE OF ON KEY DETECTED IN STEP S504≧32?
NO
YES
S2204
KEY CODE OF ON KEY DETECTED IN STEP S504≧16?
NO
YES
S2205
KEY CODE OF ON KEY DETECTED IN STEP S504≧48?
NO
YES
S2206
KEY CODE OF ON KEY DETECTED IN STEP S504≧32?
NO
YES
S2207
KEY CODE OF ON KEY DETECTED IN STEP S504≧48?
NO
YES
S2208
KEY CODE OF ON KEY DETECTED IN STEP S504≧16?
NO
YES
S2209
SET PCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN X REGISTER AND CORRESPONDING TO PCM METHOD IN AREAS AFTER TABLE HEADER
S2210
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN X REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER
S2211
SET TM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN X REGISTER AND CORRESPONDING TO TM METHOD IN AREAS AFTER TABLE HEADER
S2212
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN X REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
S2213
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
S2214
SET TM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO TM METHOD IN AREAS AFTER TABLE HEADER
S2215
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER
S2216
SET PCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO PCM METHOD IN AREAS AFTER TABLE HEADER

FIG.29

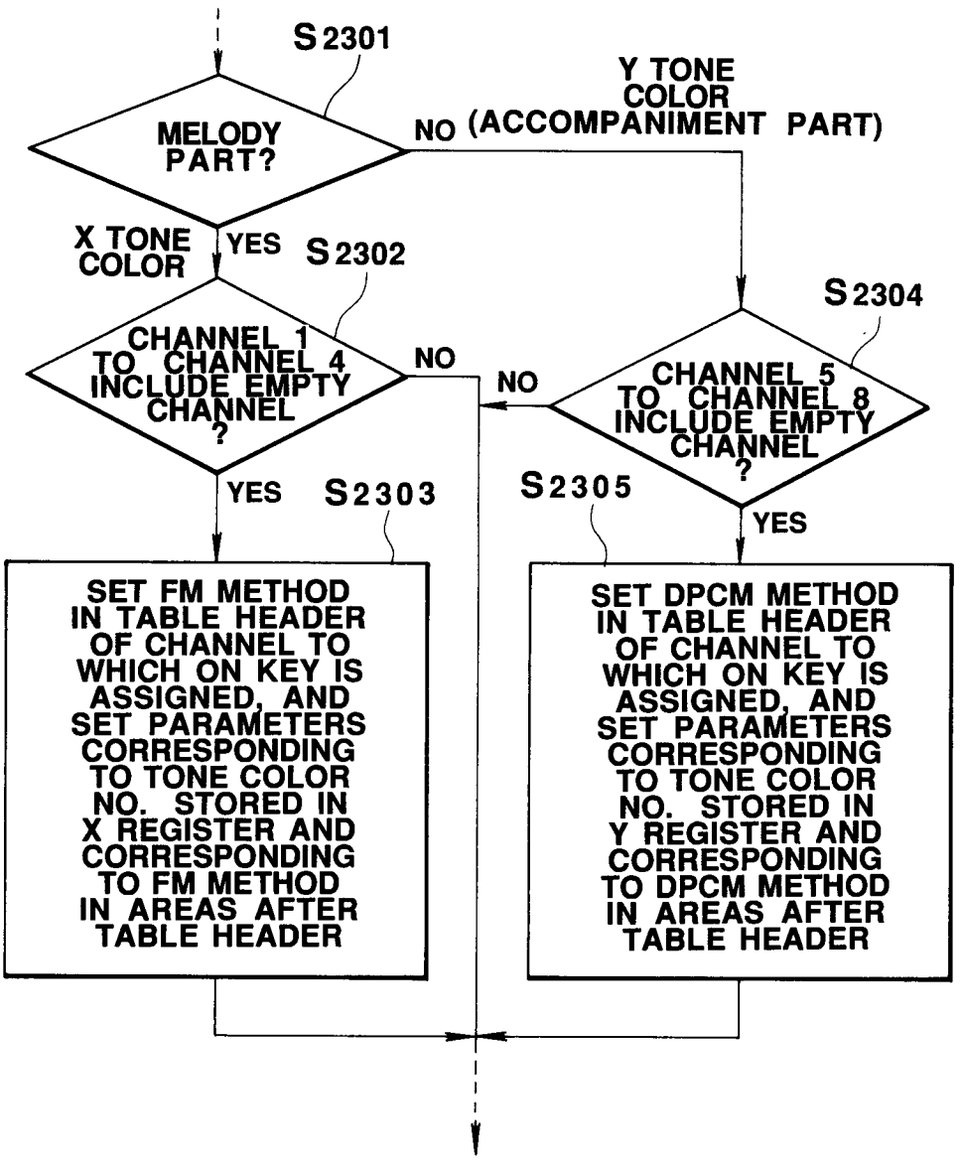
S2301
MELODY PART?
NO
Y TONE COLOR (ACCOMPANIMENT PART)
X TONE COLOR
YES
S2302
CHANNEL 1 TO CHANNEL 4 INCLUDE EMPTY CHANNEL ?
NO
S2304
CHANNEL 5 TO CHANNEL 8 INCLUDE EMPTY CHANNEL ?
NO
YES
S2303
S2305
YES
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN X REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER

FIG.30

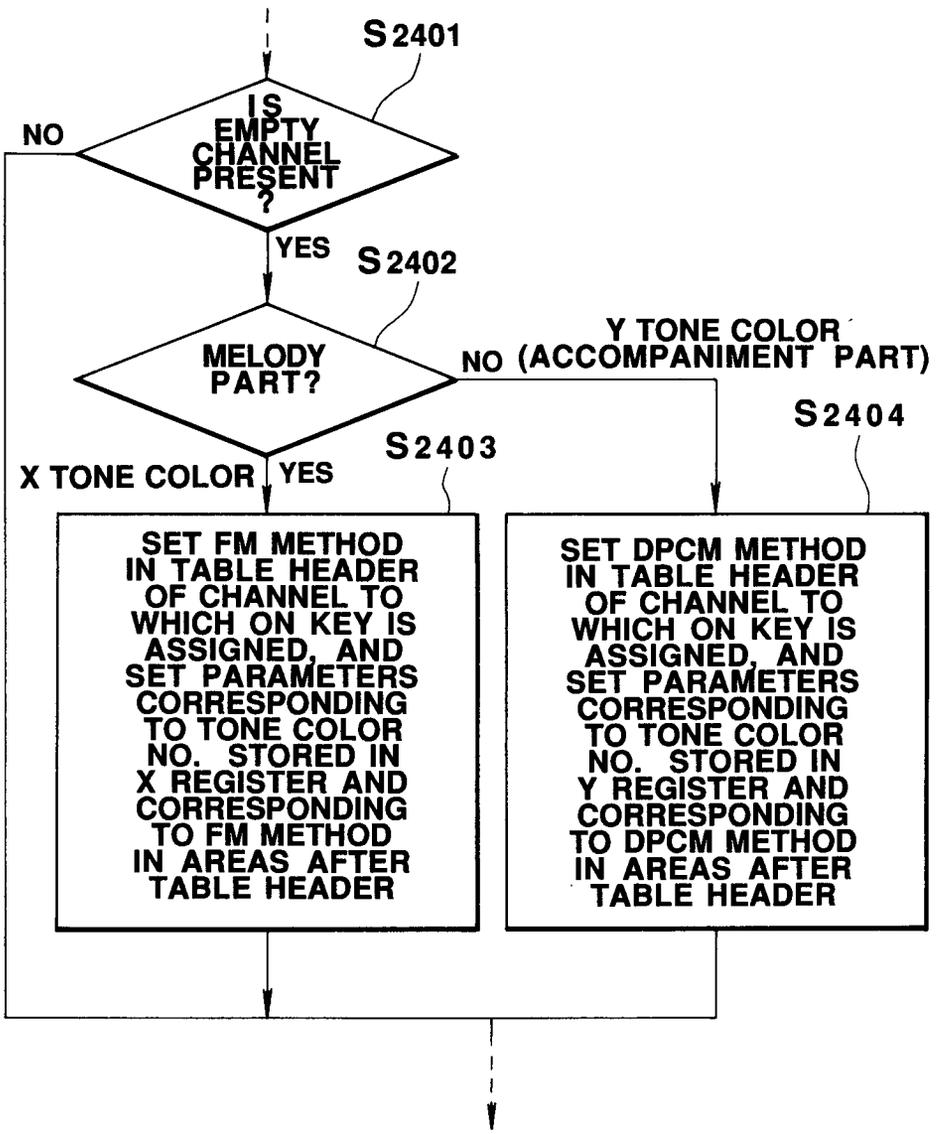
S2401
IS EMPTY CHANNEL PRESENT ?
NO
YES
S2402
MELODY PART?
Y TONE COLOR (ACCOMPANIMENT PART)
NO
S2404
S2403
X TONE COLOR
YES
SET FM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN X REGISTER AND CORRESPONDING TO FM METHOD IN AREAS AFTER TABLE HEADER
SET DPCM METHOD IN TABLE HEADER OF CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO TONE COLOR NO. STORED IN Y REGISTER AND CORRESPONDING TO DPCM METHOD IN AREAS AFTER TABLE HEADER

FIG. 31

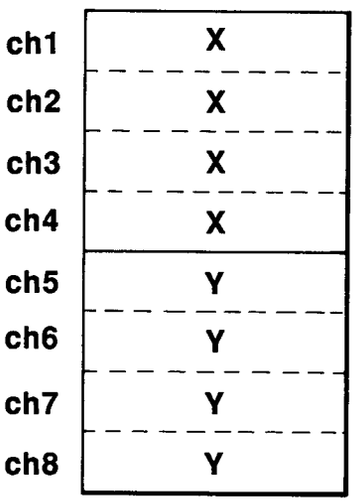
ch1
X
ch2
X
ch3
X
ch4
X
ch5
Y
ch6
Y
ch7
Y
ch8
Y

FIG. 32

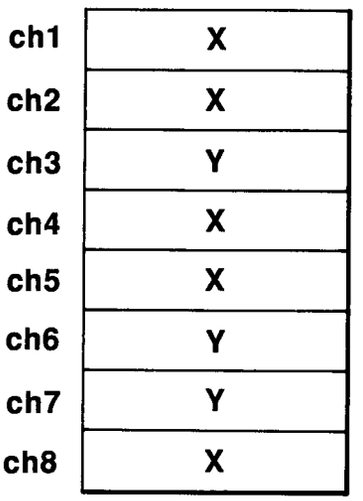
ch1
X
ch2
X
ch3
Y
ch4
X
ch5
X
ch6
Y
ch7
Y
ch8
X

FIG. 33

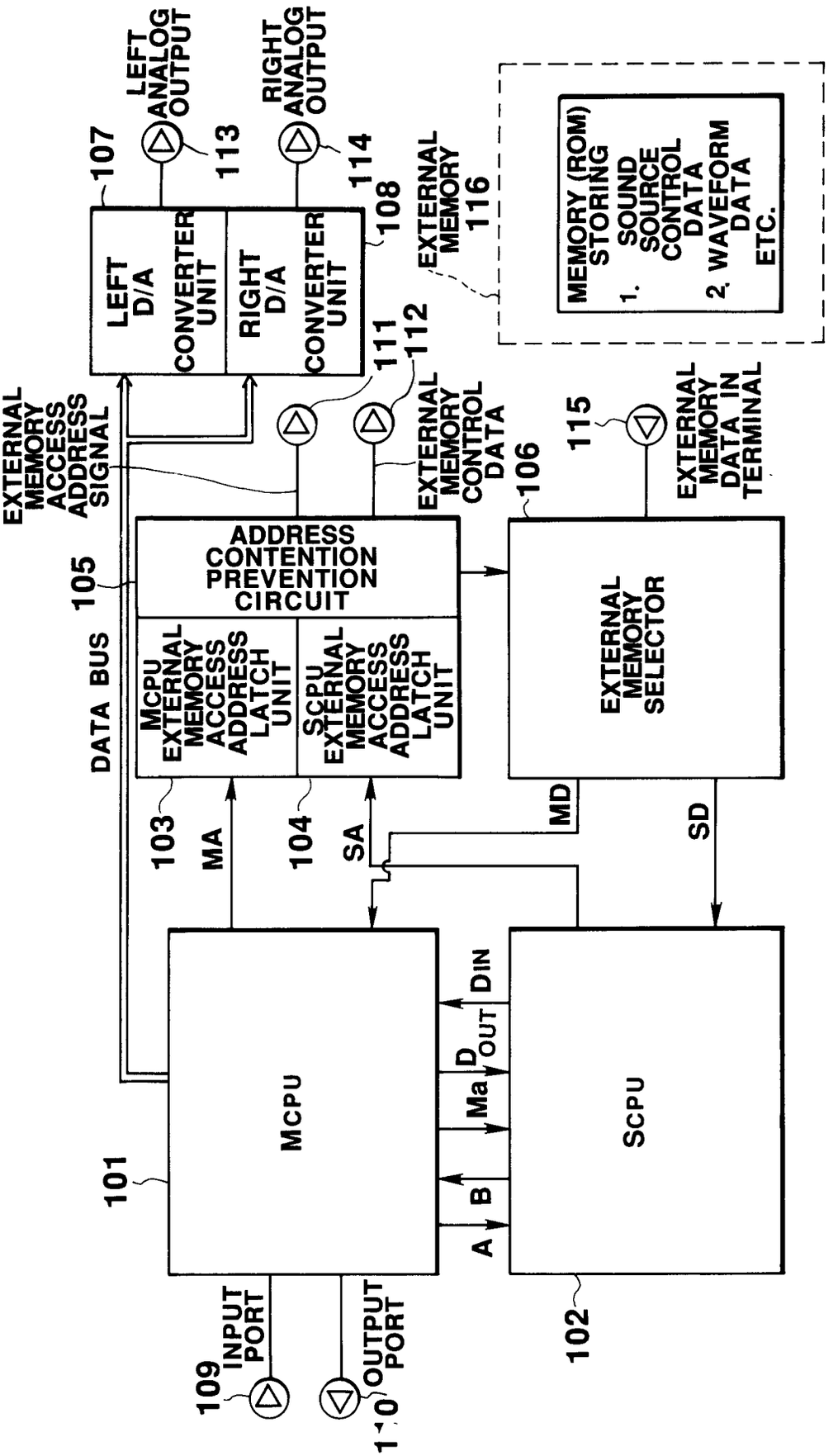
LEFT ANALOG OUTPUT
RIGHT ANALOG OUTPUT
107
113
114
108
EXTERNAL MEMORY 116
MEMORY (ROM) STORING
1. SOUND SOURCE CONTROL DATA
2. WAVEFORM DATA
ETC.
LEFT D/A CONVERTER UNIT
RIGHT D/A CONVERTER UNIT
111
112
EXTERNAL MEMORY ACCESS ADDRESS SIGNAL
EXTERNAL MEMORY CONTROL DATA
115
EXTERNAL MEMORY DATA IN TERMINAL
106
105
ADDRESS CONTENTION PREVENTION CIRCUIT
MCPU EXTERNAL MEMORY ACCESS ADDRESS LATCH UNIT
SCPU EXTERNAL MEMORY ACCESS ADDRESS LATCH UNIT
EXTERNAL MEMORY SELECTOR
DATA BUS
103
MA
104
SA
MD
SD
DIN
DOUT
Ma
B
A
MCPU
SCPU
101
102
109
INPUT PORT
OUTPUT PORT

FIG.34

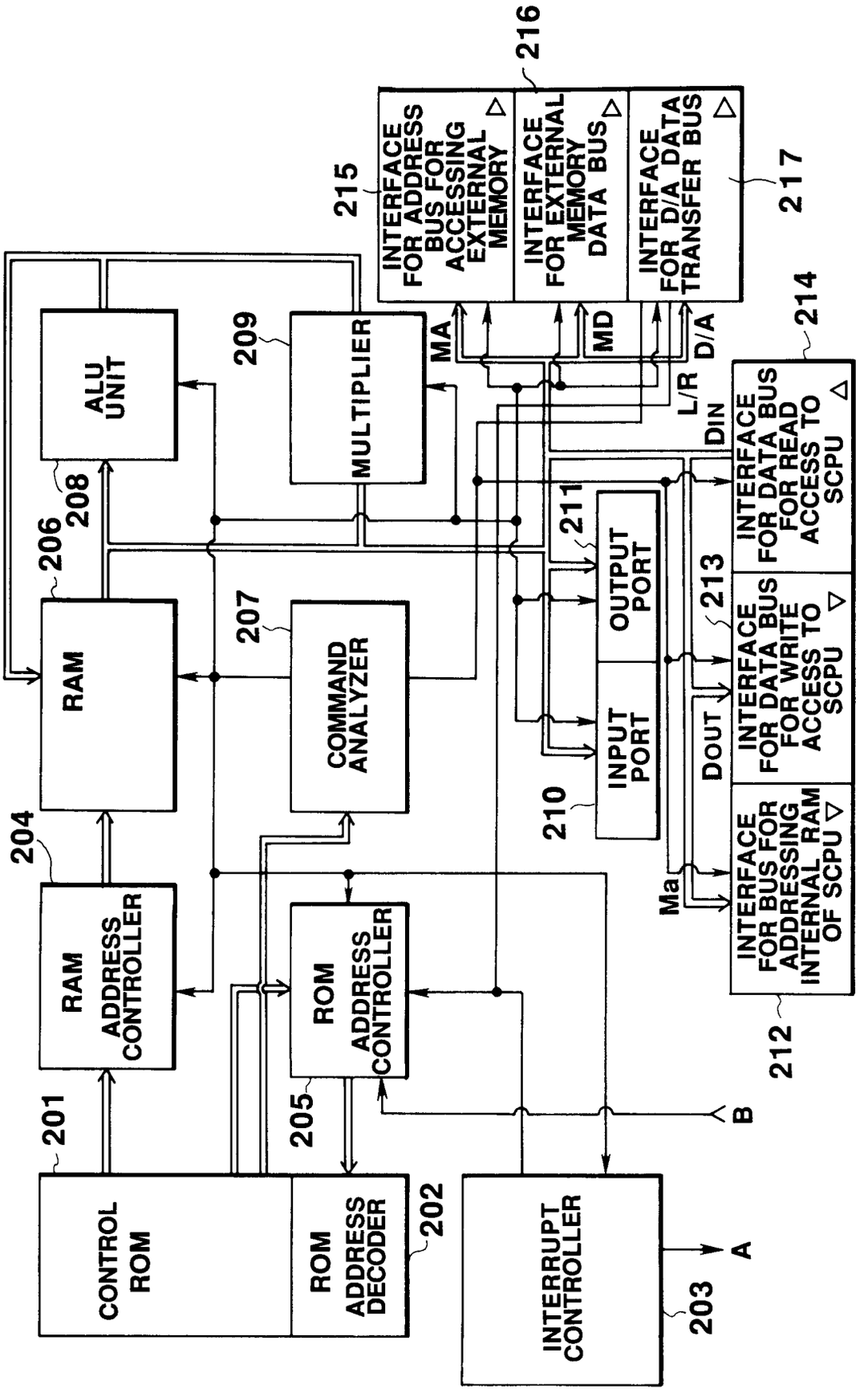
216
215
217
214
INTERFACE FOR ADDRESS BUS FOR ACCESSING EXTERNAL MEMORY
INTERFACE FOR EXTERNAL MEMORY DATA BUS
INTERFACE FOR D/A DATA TRANSFER BUS
ALU UNIT
MULTIPLIER
209
MA
MD
D/A
L/R
DIN
208
206
207
211
OUTPUT PORT
INTERFACE FOR DATA BUS FOR READ ACCESS TO SCPU
213
INTERFACE FOR DATA BUS FOR WRITE ACCESS TO SCPU
DOUT
RAM
COMMAND ANALYZER
INPUT PORT
210
204
RAM ADDRESS CONTROLLER
ROM ADDRESS CONTROLLER
Ma
INTERFACE FOR BUS FOR ADDRESSING INTERNAL RAM OF SCPU
212
201
205
B
CONTROL ROM
ROM ADDRESS DECODER
202
INTERRUPT CONTROLLER
A
203

FIG. 35

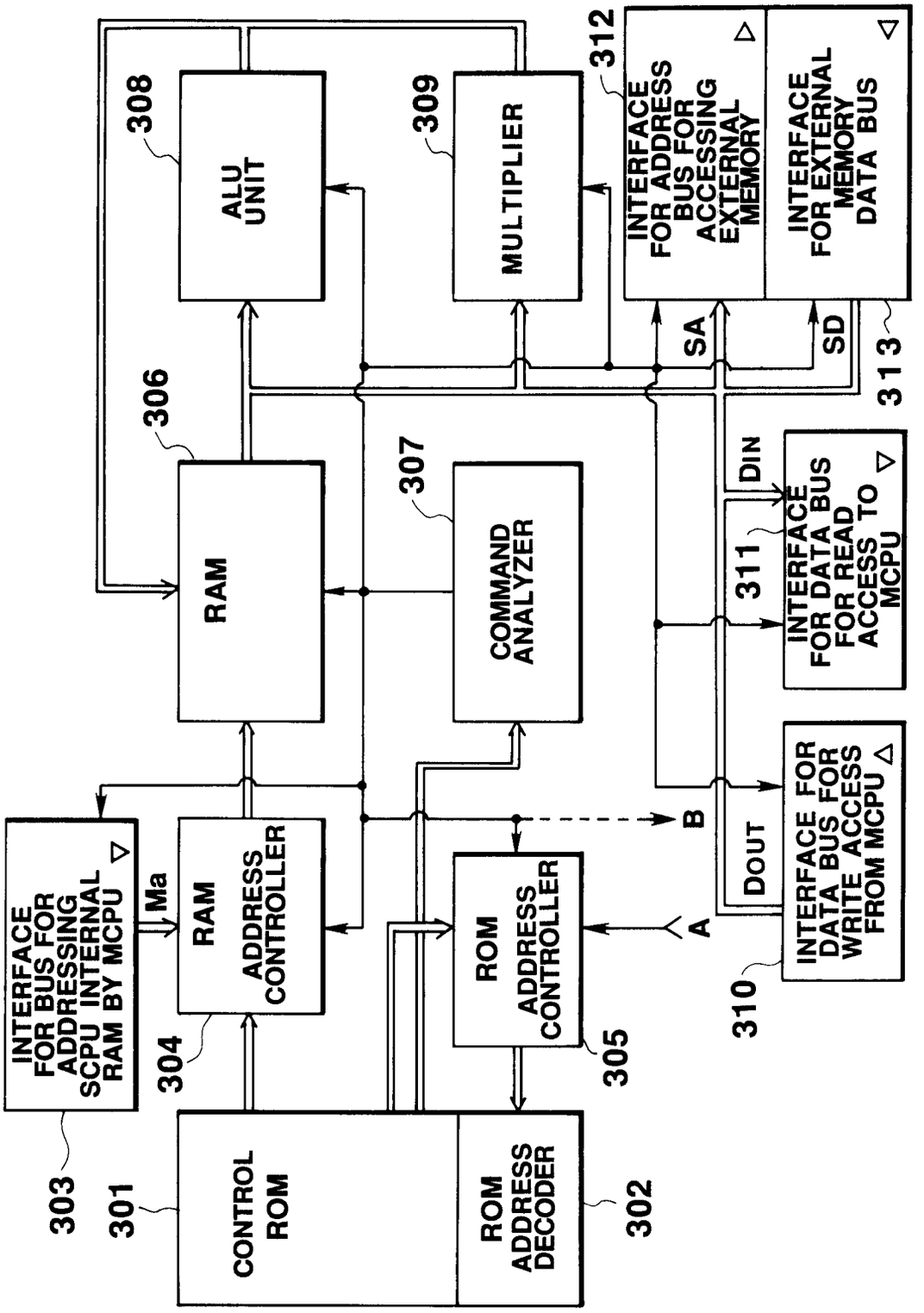
INTERFACE FOR BUS FOR ADDRESSING SCPU INTERNAL RAM BY MCPU
303
Ma
RAM ADDRESS CONTROLLER
304
CONTROL ROM
301
ROM ADDRESS DECODER
302
ROM ADDRESS CONTROLLER
305
RAM
306
COMMAND ANALYZER
307
ALU UNIT
308
MULTIPLIER
309
INTERFACE FOR DATA BUS FOR WRITE ACCESS FROM MCPU
310
INTERFACE FOR DATA BUS FOR READ ACCESS TO MCPU
311
INTERFACE FOR ADDRESS BUS FOR ACCESSING EXTERNAL MEMORY
312
INTERFACE FOR EXTERNAL MEMORY DATA BUS
313
SA
SD
DIN
DOUT
A
B

FIG. 36

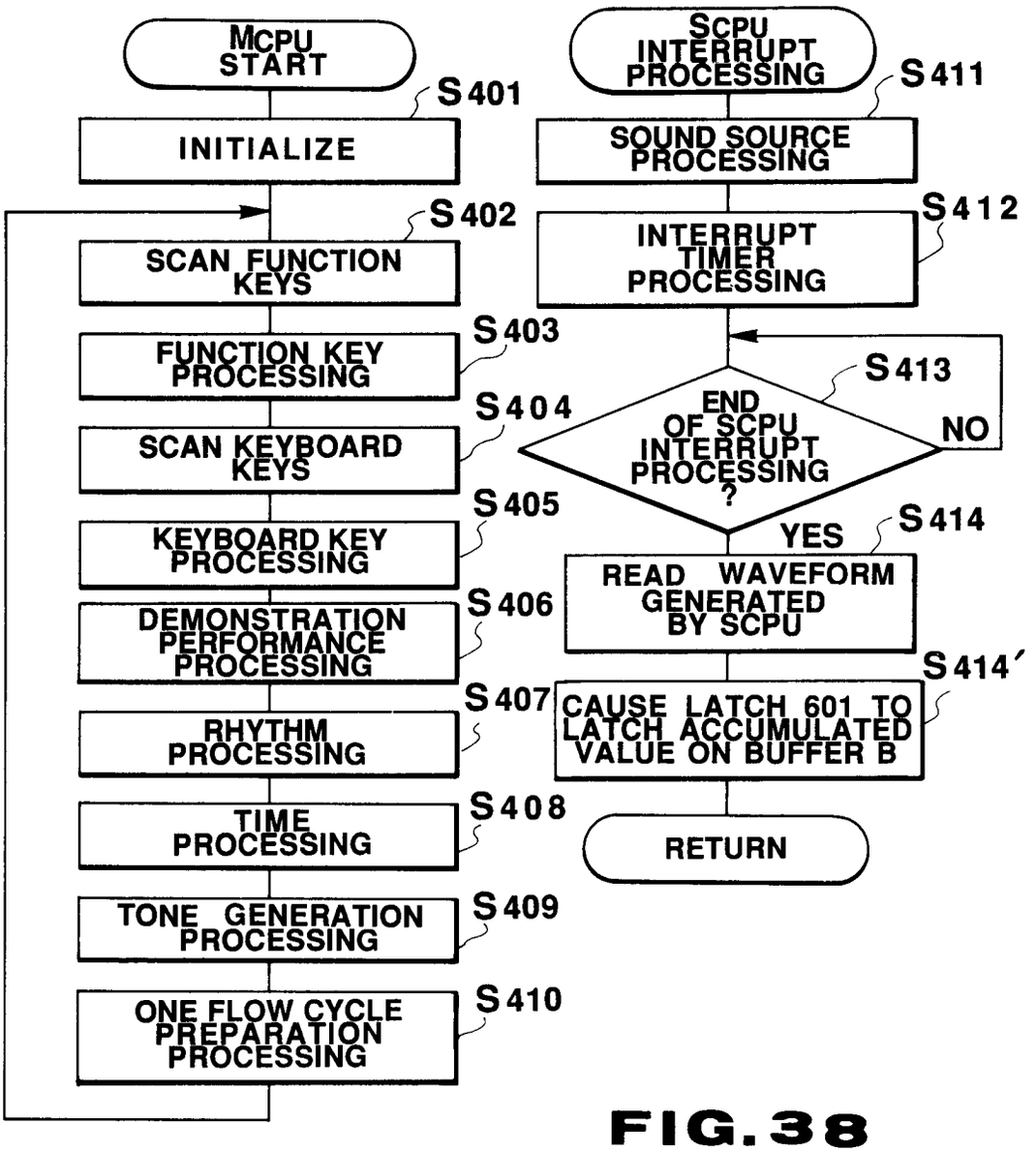
MCPU START
S401
INITIALIZE
S402
SCAN FUNCTION KEYS
S403
FUNCTION KEY PROCESSING
S404
SCAN KEYBOARD KEYS
S405
KEYBOARD KEY PROCESSING
S406
DEMONSTRATION PERFORMANCE PROCESSING
S407
RHYTHM PROCESSING
S408
TIME PROCESSING
S409
TONE GENERATION PROCESSING
S410
ONE FLOW CYCLE PREPARATION PROCESSING
SCPU INTERRUPT PROCESSING
S411
SOUND SOURCE PROCESSING
S412
INTERRUPT TIMER PROCESSING
S413
END OF SCPU INTERRUPT PROCESSING ?
NO
YES
S414
READ WAVEFORM GENERATED BY SCPU
S414'
CAUSE LATCH 601 TO LATCH ACCUMULATED VALUE ON BUFFER B
RETURN

FIG. 38

FIG. 37

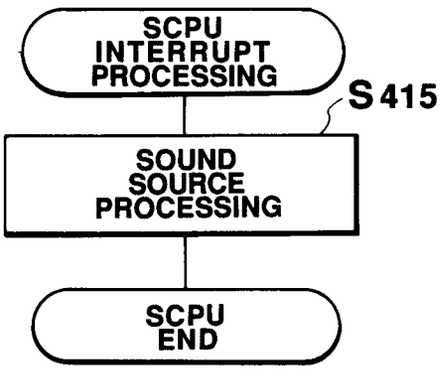
SCPU INTERRUPT PROCESSING
S415
SOUND SOURCE PROCESSING
SCPU END

FIG.39

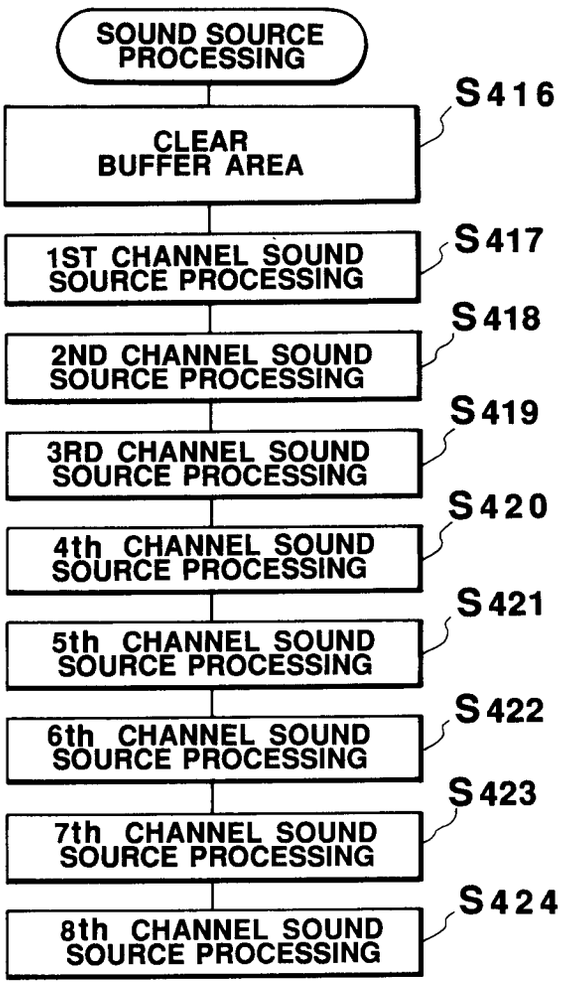
SOUND SOURCE PROCESSING
S416
CLEAR BUFFER AREA
S417
1ST CHANNEL SOUND SOURCE PROCESSING
S418
2ND CHANNEL SOUND SOURCE PROCESSING
S419
3RD CHANNEL SOUND SOURCE PROCESSING
S420
4th CHANNEL SOUND SOURCE PROCESSING
S421
5th CHANNEL SOUND SOURCE PROCESSING
S422
6th CHANNEL SOUND SOURCE PROCESSING
S423
7th CHANNEL SOUND SOURCE PROCESSING
S424
8th CHANNEL SOUND SOURCE PROCESSING

FIG.40

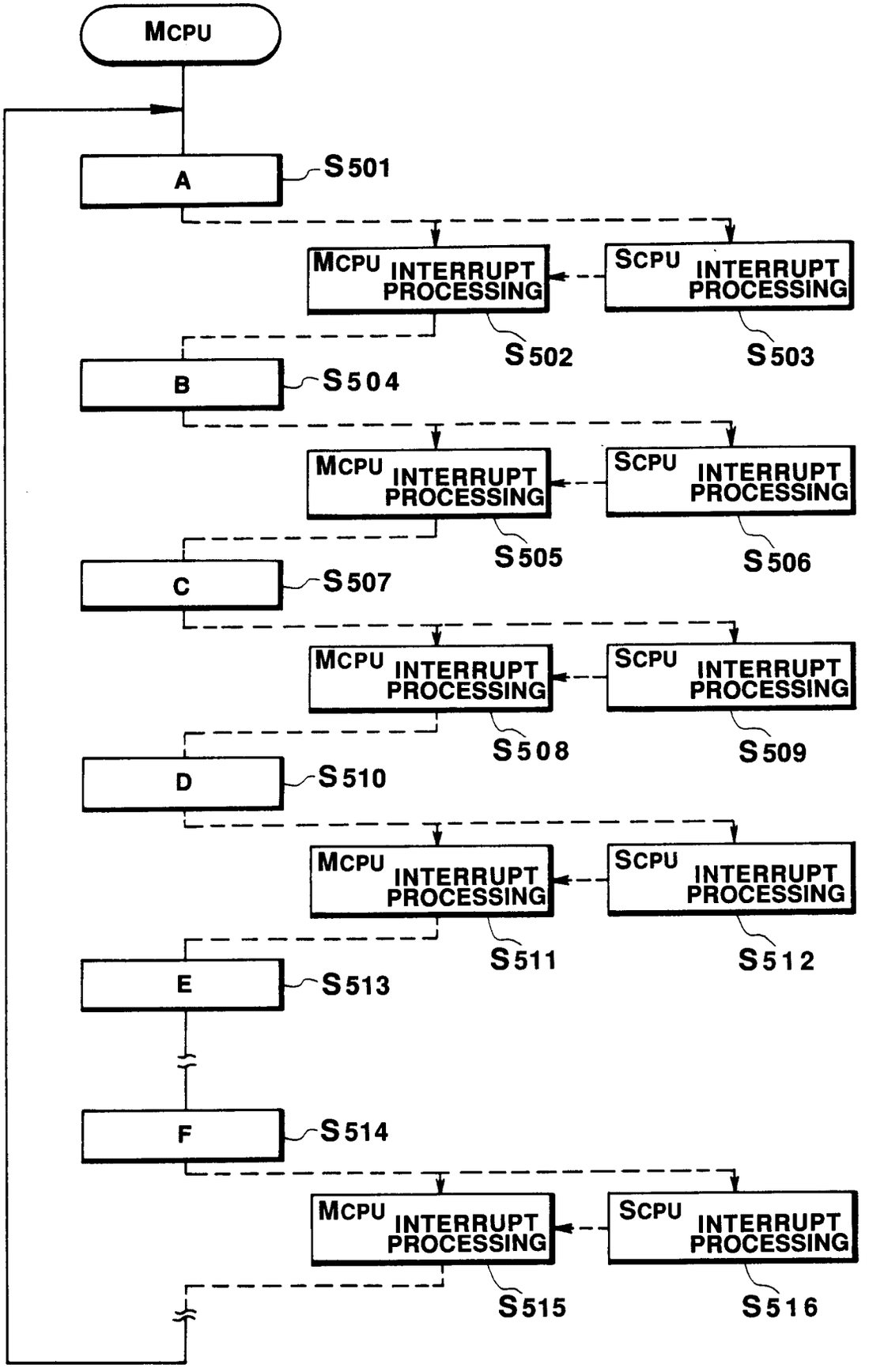
MCPU
A
S501
MCPU INTERRUPT PROCESSING
SCPU INTERRUPT PROCESSING
S502
S503
B
S504
MCPU INTERRUPT PROCESSING
SCPU INTERRUPT PROCESSING
S505
S506
C
S507
MCPU INTERRUPT PROCESSING
SCPU INTERRUPT PROCESSING
S508
S509
D
S510
MCPU INTERRUPT PROCESSING
SCPU INTERRUPT PROCESSING
S511
S512
E
S513
F
S514
MCPU INTERRUPT PROCESSING
SCPU INTERRUPT PROCESSING
S515
S516

FIG.41

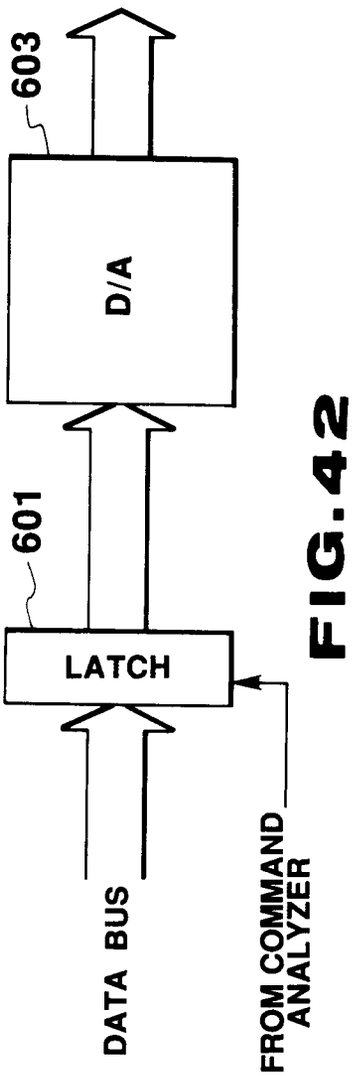
603
D/A
601
LATCH
DATA BUS
FROM COMMAND ANALYZER

FIG.42

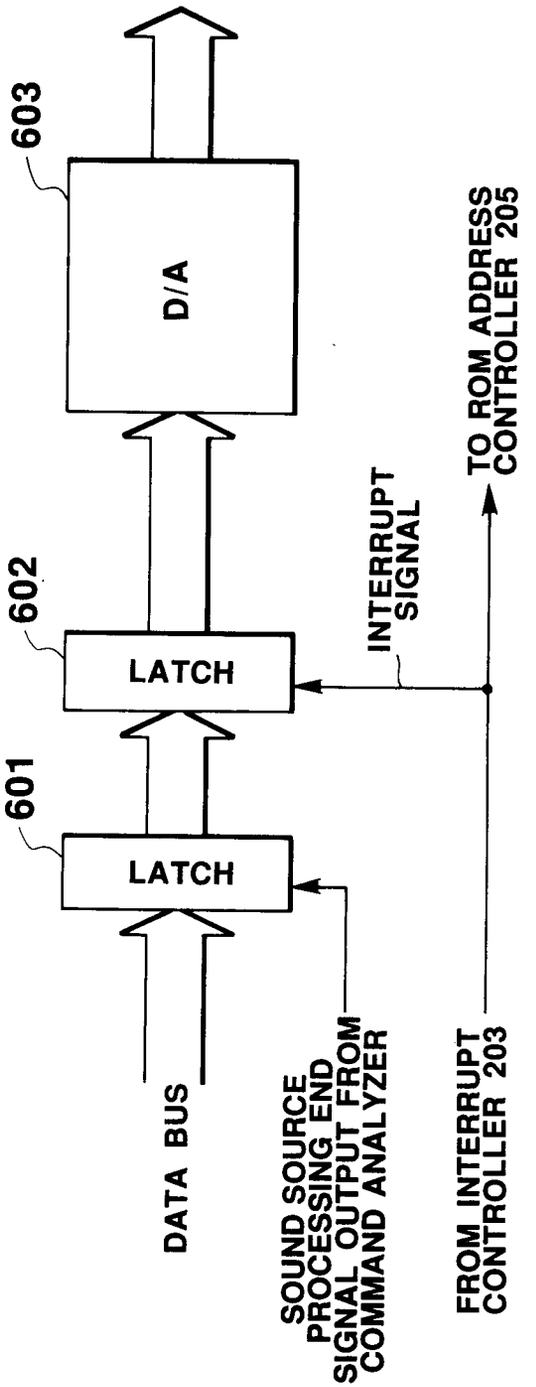
603
D/A
602
LATCH
601
LATCH
INTERRUPT SIGNAL
TO ROM ADDRESS CONTROLLER 205
DATA BUS
SOUND SOURCE PROCESSING END SIGNAL OUTPUT FROM COMMAND ANALYZER
FROM INTERRUPT CONTROLLER 203

FIG.43

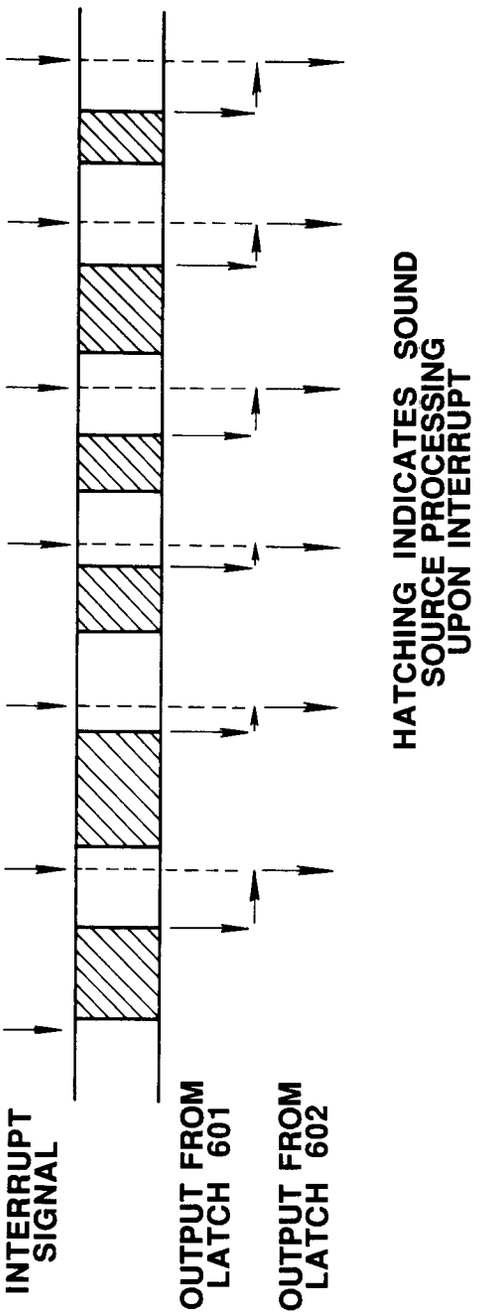
INTERRUPT SIGNAL
OUTPUT FROM LATCH 601
OUTPUT FROM LATCH 602
HATCHING INDICATES SOUND SOURCE PROCESSING UPON INTERRUPT

FIG.44

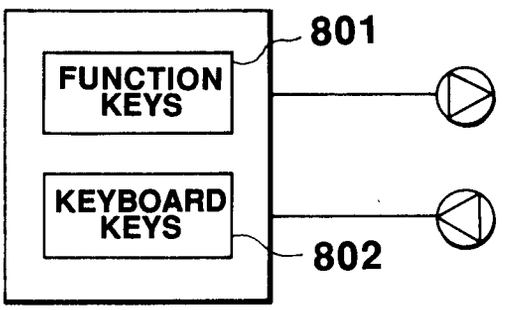
801
FUNCTION KEYS
KEYBOARD KEYS
802

FIG.45

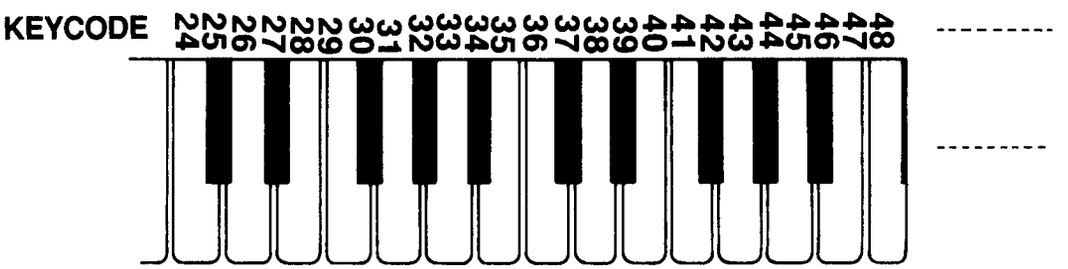
KEYCODE
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48

FIG.46

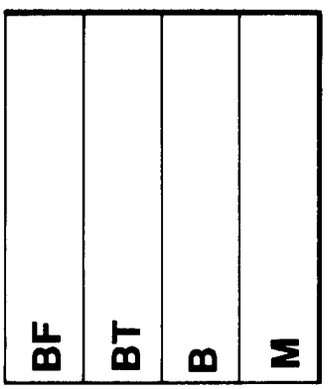

FIG.50

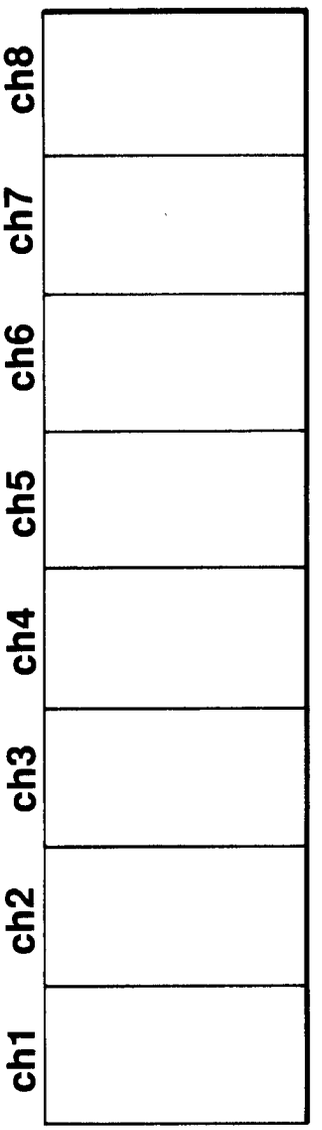

FIG.47

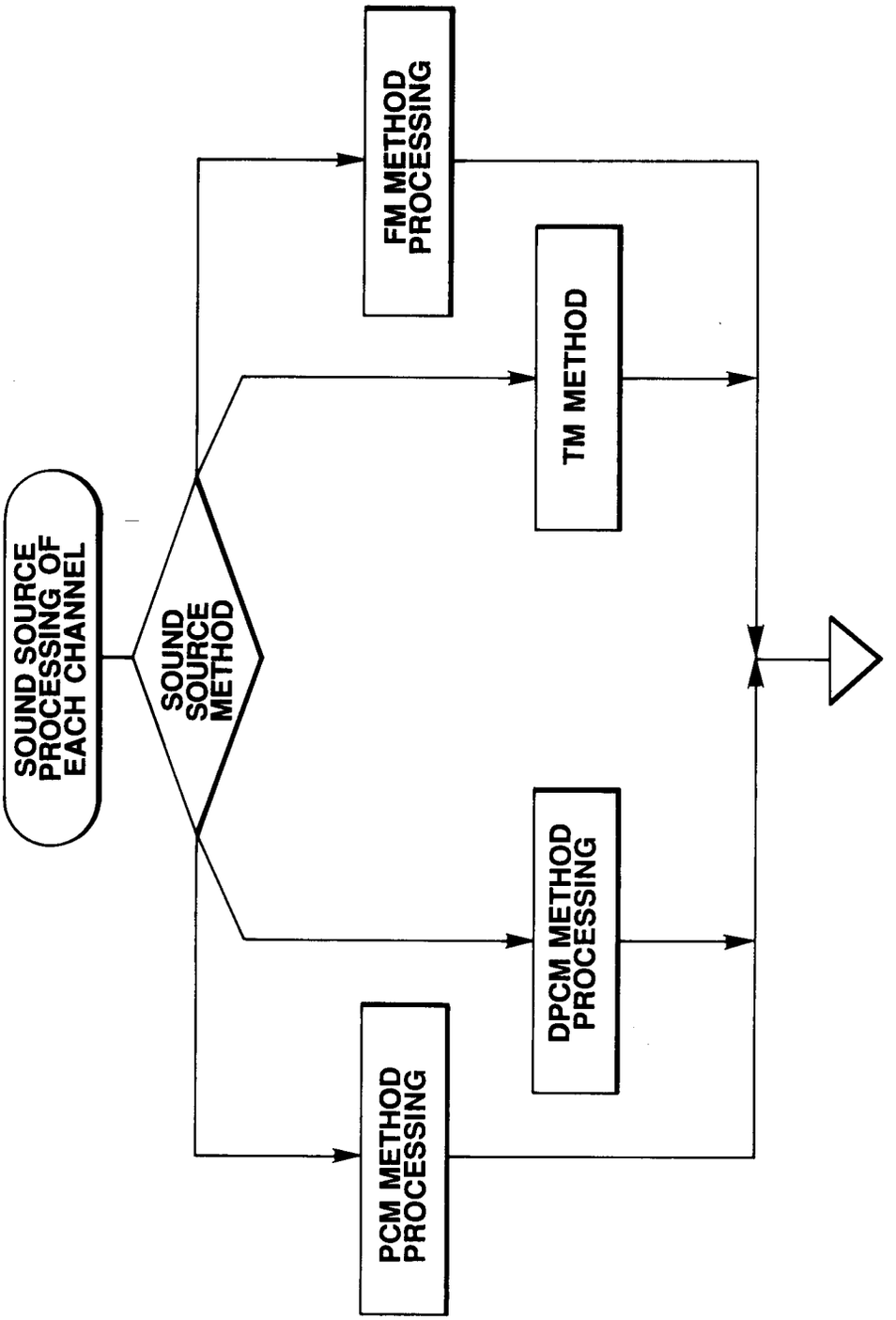
SOUND SOURCE PROCESSING OF EACH CHANNEL
SOUND SOURCE METHOD
FM METHOD PROCESSING
TM METHOD
DPCM METHOD PROCESSING
PCM METHOD PROCESSING

FIG.48

PCM		DPCM		FM		TM	
G	SOUND SOURCE METHOD NO.	G	SOUND SOURCE METHOD NO.	G	SOUND SOURCE METHOD NO.	G	SOUND SOURCE METHOD NO.
A_I	CURRENT ADDRESS INTEGRAL PART	A_I	CURRENT ADDRESS INTEGRAL PART	A_1	CURRENT ADDRESS (OP1)	A_1	CURRENT ADDRESS (OP1)
A_F	CURRENT ADDRESS DECIMAL PART	A_F	CURRENT ADDRESS DECIMAL PART	A_2	CURRENT ADDRESS (OP2)	A_2	CURRENT ADDRESS (OP2)
A_E	END ADDRESS INTEGRAL PART	A_E	END ADDRESS INTEGRAL PART				
A_L	LOOP ADDRESS INTEGRAL PART	A_L	LOOP ADDRESS INTEGRAL PART				
P_F	PITCH DATA DECIMAL PART	P_F	PITCH DATA DECIMAL PART	P_1	PITCH DATA (OP1)	P_1	PITCH DATA (OP1)
P_I	PITCH DATA INTEGRAL PART	P_I	PITCH DATA INTEGRAL PART	P_2	PITCH DATA (OP2)	P_2	PITCH DATA (OP2)
X_P	IMMEDIATELY PRECEDING SAMPLE DATA	X_P	IMMEDIATELY PRECEDING SAMPLE DATA				
X_N	NEXT SAMPLE DATA	OLD A_I	CURRENT ADDRESS INTEGRAL PART BEFORE CHANGE				
D	DIFFERENCE BETWEEN ADJACENT SAMPLE DATA	D	DIFFERENCE BETWEEN ADJACENT SAMPLE DATA				
E	ENVELOPE VALUE	E	ENVELOPE VALUE	E_1	ENVELOPE (OP1)	E_1	ENVELOPE (OP1)
		X_{PL}	SAMPLE DATA OF A_L	E_2	ENVELOPE (OP2)	E_2	ENVELOPE (OP2)
				M_{L2}	MODULATION LEVEL (OP2)	M_{L2}	MODULATION LEVEL (OP2)
				M_{O2}	MODULATION OUTPUT (OP2)	M_{O2}	MODULATION OUTPUT (OP2)
				F_{L2}	FEEDBACK LEVEL (OP2)	F_{L2}	FEEDBACK LEVEL (OP2)
				F_{O2}	FEEDBACK OUTPUT (OP2)	F_{O2}	FEEDBACK OUTPUT (OP2)
O	OUTPUT	O	OUTPUT	O_1	OP1 OUTPUT	O_1	OP1 OUTPUT
				O_2	OP2 OUTPUT	O_2	OP2 OUTPUT
C		C		C		C	

FIG.49

EP 0 463 411 A2

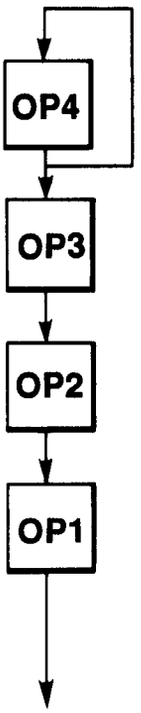

FIG. 51

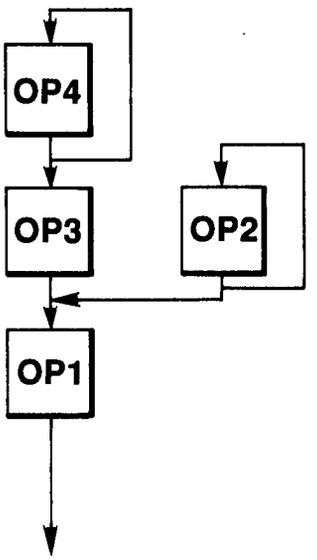

FIG. 52

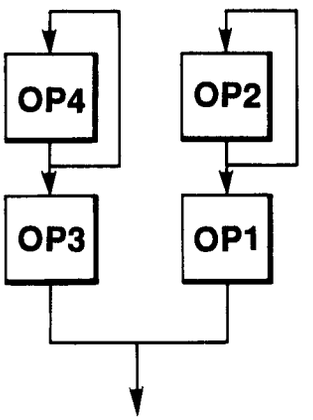

FIG. 53

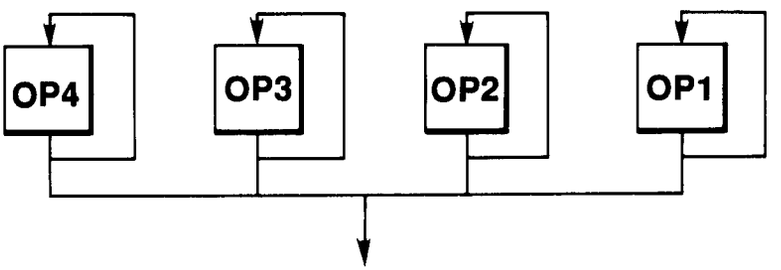

FIG. 54

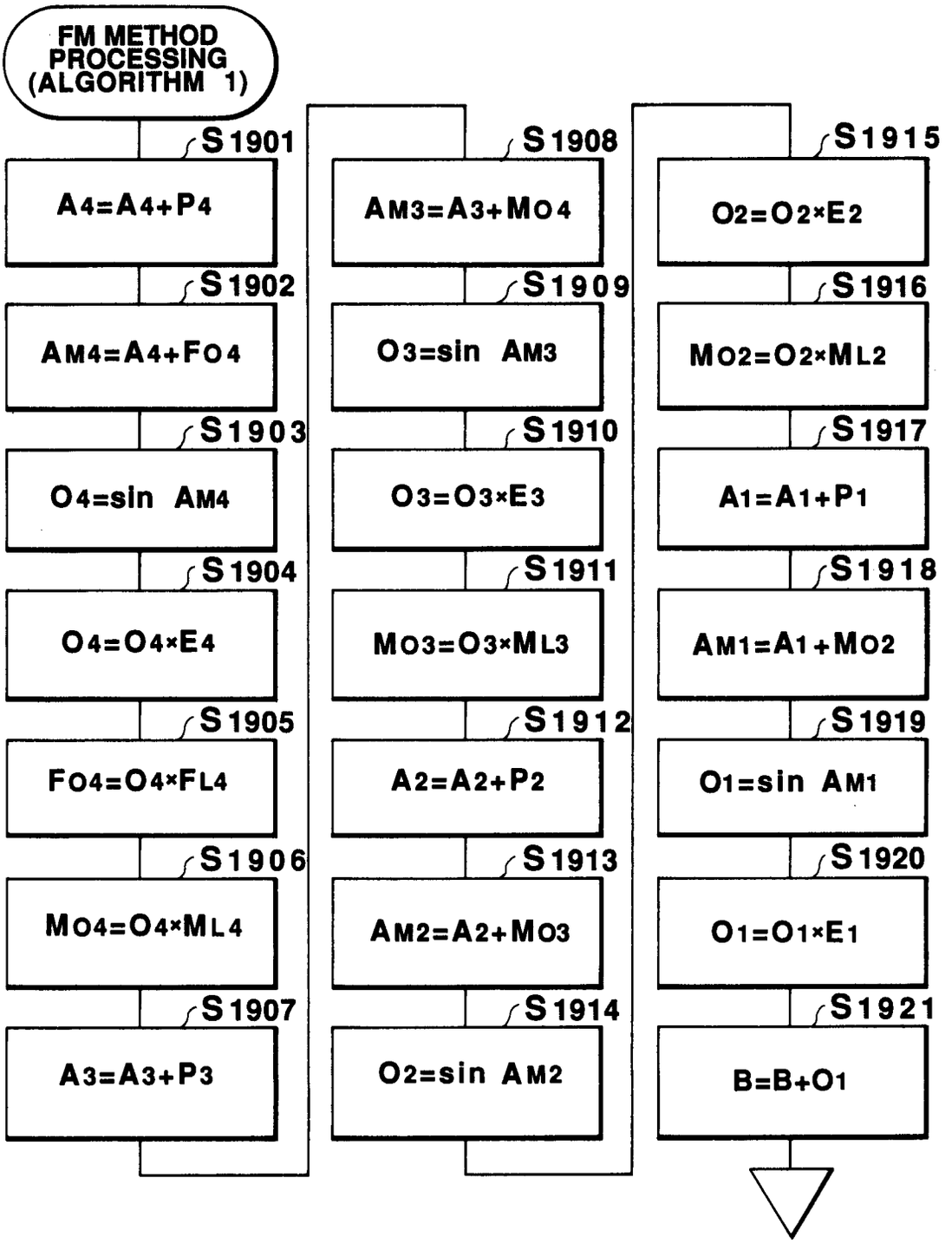
FM METHOD PROCESSING (ALGORITHM 1)
S1901
A4=A4+P4
S1902
AM4=A4+FO4
S1903
O4=sin AM4
S1904
O4=O4×E4
S1905
FO4=O4×FL4
S1906
MO4=O4×ML4
S1907
A3=A3+P3
S1908
AM3=A3+MO4
S1909
O3=sin AM3
S1910
O3=O3×E3
S1911
MO3=O3×ML3
S1912
A2=A2+P2
S1913
AM2=A2+MO3
S1914
O2=sin AM2
S1915
O2=O2×E2
S1916
MO2=O2×ML2
S1917
A1=A1+P1
S1918
AM1=A1+MO2
S1919
O1=sin AM1
S1920
O1=O1×E1
S1921
B=B+O1

FIG. 55

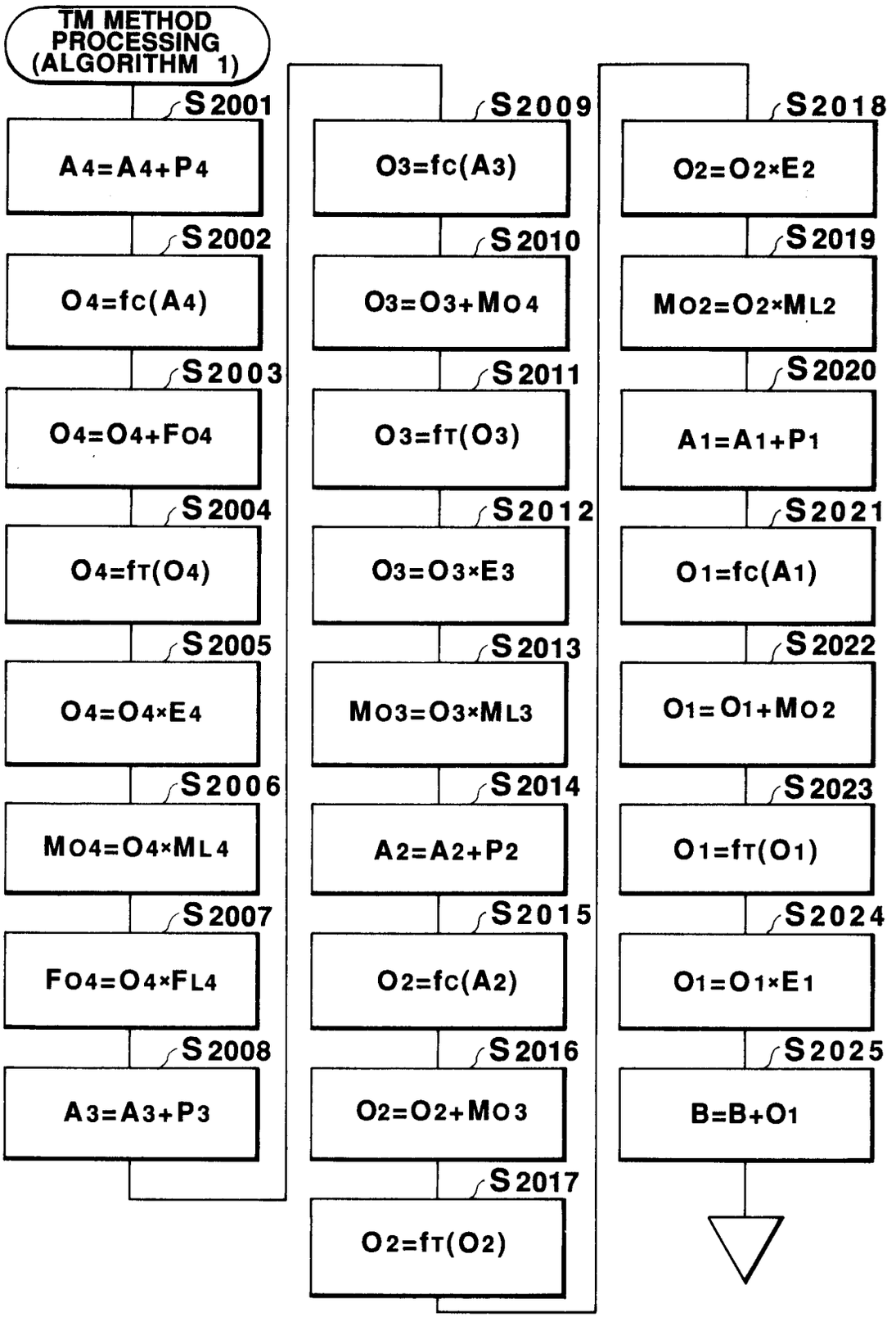
TM METHOD PROCESSING (ALGORITHM 1)
S2001
A4=A4+P4
S2002
O4=fc(A4)
S2003
O4=O4+FO4
S2004
O4=fT(O4)
S2005
O4=O4×E4
S2006
MO4=O4×ML4
S2007
FO4=O4×FL4
S2008
A3=A3+P3
S2009
O3=fc(A3)
S2010
O3=O3+MO4
S2011
O3=fT(O3)
S2012
O3=O3×E3
S2013
MO3=O3×ML3
S2014
A2=A2+P2
S2015
O2=fc(A2)
S2016
O2=O2+MO3
S2017
O2=fT(O2)
S2018
O2=O2×E2
S2019
MO2=O2×ML2
S2020
A1=A1+P1
S2021
O1=fc(A1)
S2022
O1=O1+MO2
S2023
O1=fT(O1)
S2024
O1=O1×E1
S2025
B=B+O1

FIG.56

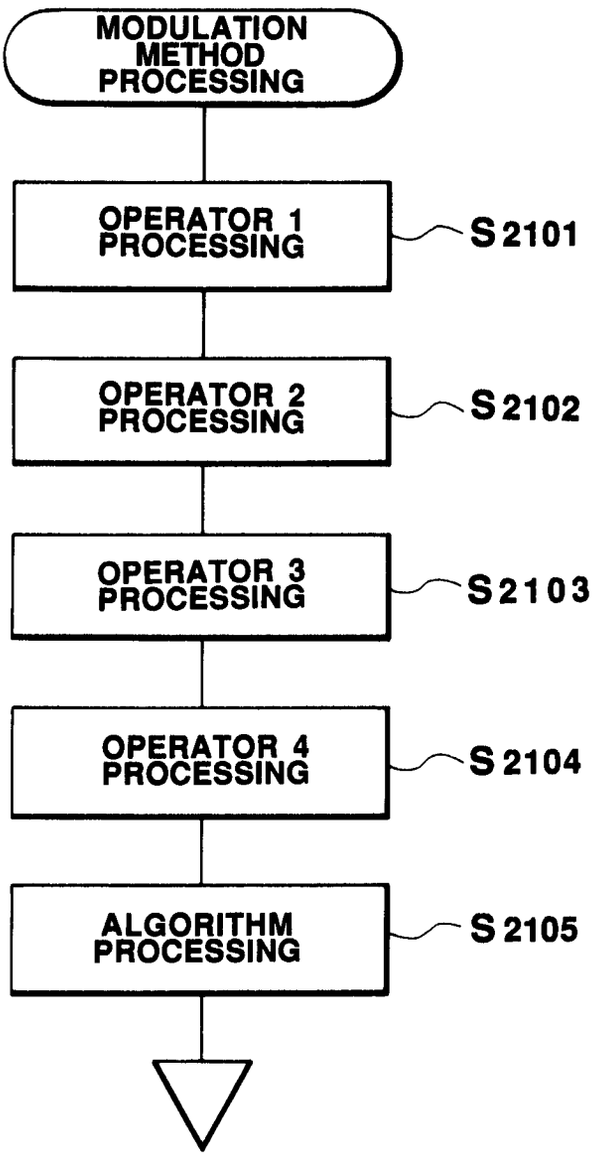
MODULATION METHOD PROCESSING
OPERATOR 1 PROCESSING
S2101
OPERATOR 2 PROCESSING
S2102
OPERATOR 3 PROCESSING
S2103
OPERATOR 4 PROCESSING
S2104
ALGORITHM PROCESSING
S2105

FIG.57

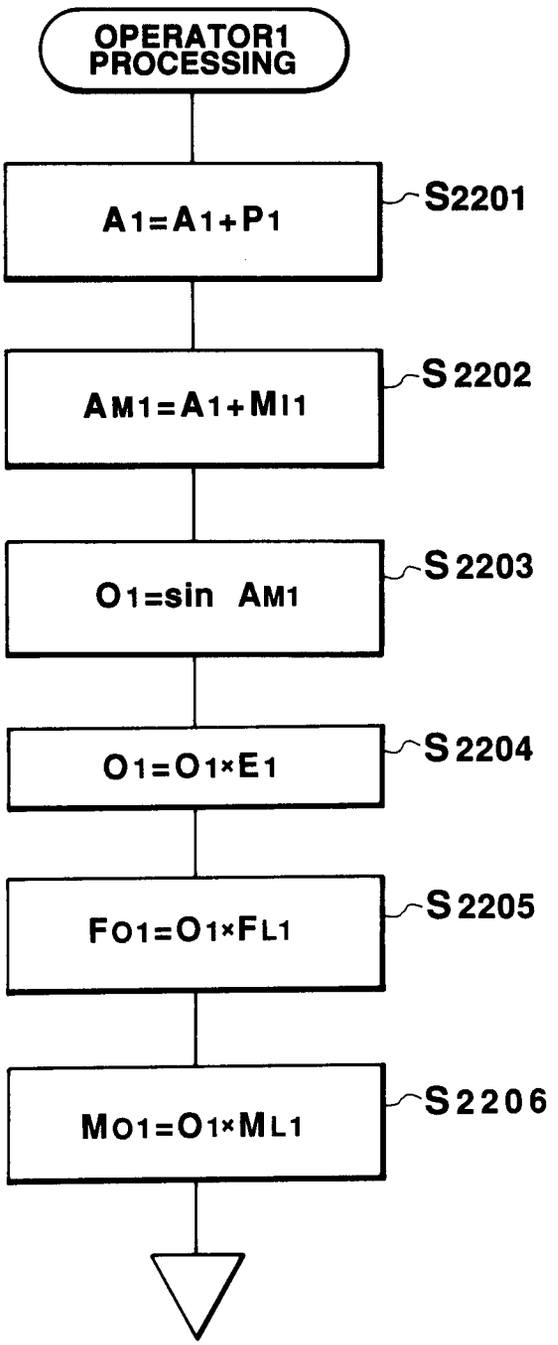
OPERATOR1 PROCESSING
A1=A1+P1
S2201
AM1=A1+MI1
S2202
O1=sin AM1
S2203
O1=O1×E1
S2204
FO1=O1×FL1
S2205
MO1=O1×ML1
S2206

FIG.58

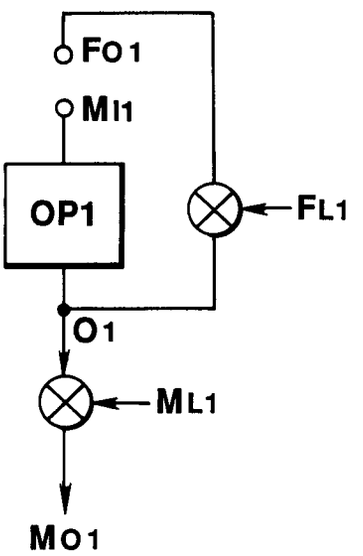
FO1
MI1
OP1
FL1
O1
ML1
MO1

FIG.59

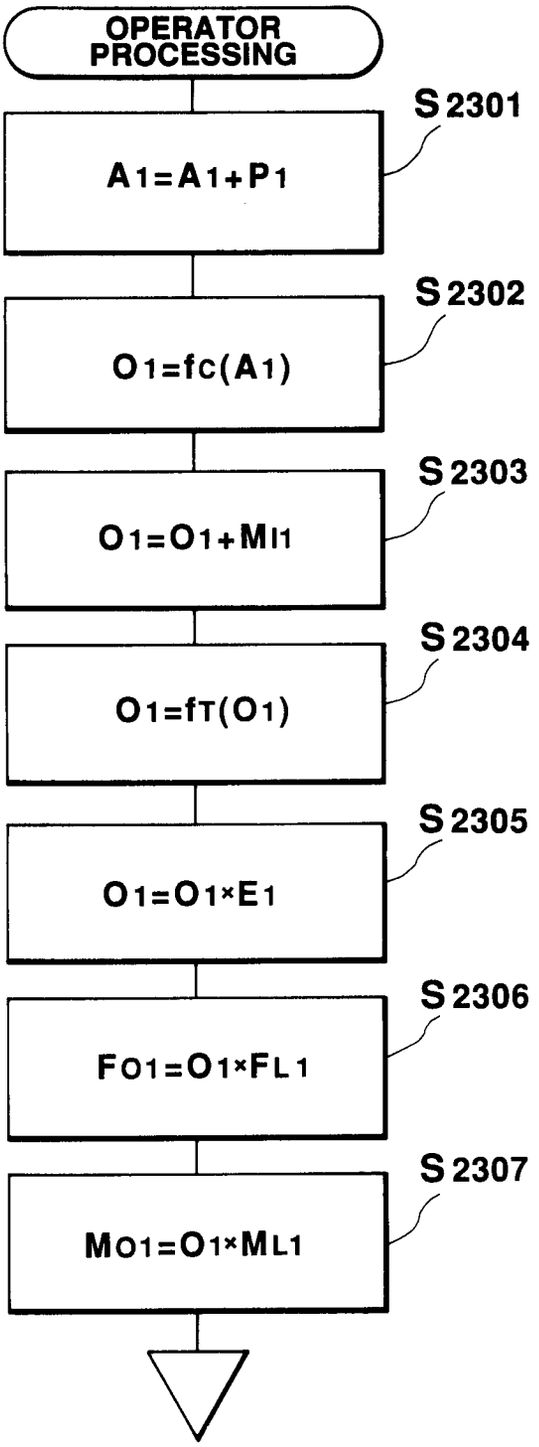
OPERATOR PROCESSING
S2301
A1=A1+P1
S2302
O1=fc(A1)
S2303
O1=O1+MI1
S2304
O1=fT(O1)
S2305
O1=O1×E1
S2306
FO1=O1×FL1
S2307
MO1=O1×ML1

FIG.60

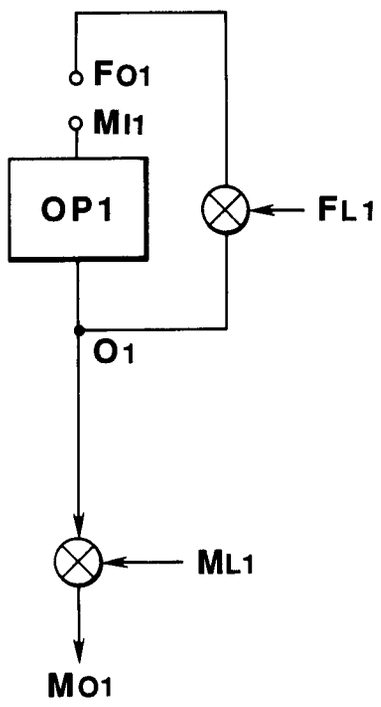
FO1
MI1
OP1
FL1
O1
ML1
MO1

FIG.61

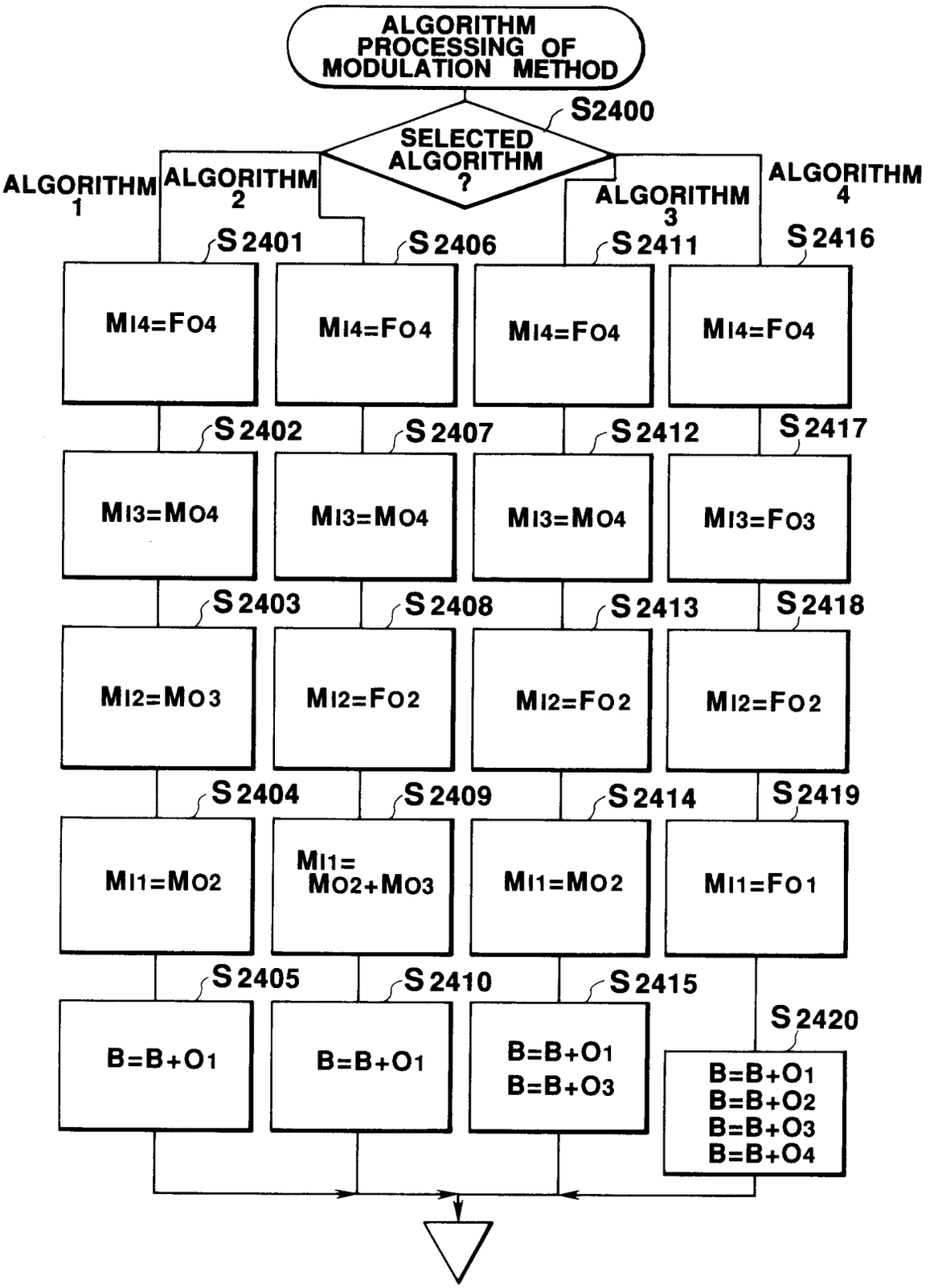
ALGORITHM PROCESSING OF MODULATION METHOD
S2400
SELECTED ALGORITHM ?
ALGORITHM 1
ALGORITHM 2
ALGORITHM 3
ALGORITHM 4
S2401
S2406
S2411
S2416
MI4=FO4
MI4=FO4
MI4=FO4
MI4=FO4
S2402
S2407
S2412
S2417
MI3=MO4
MI3=MO4
MI3=MO4
MI3=FO3
S2403
S2408
S2413
S2418
MI2=MO3
MI2=FO2
MI2=FO2
MI2=FO2
S2404
S2409
S2414
S2419
MI1=MO2
MI1= MO2+MO3
MI1=MO2
MI1=FO1
S2405
S2410
S2415
S2420
B=B+O1
B=B+O1
B=B+O1
B=B+O3
B=B+O1
B=B+O2
B=B+O3
B=B+O4

FIG.62

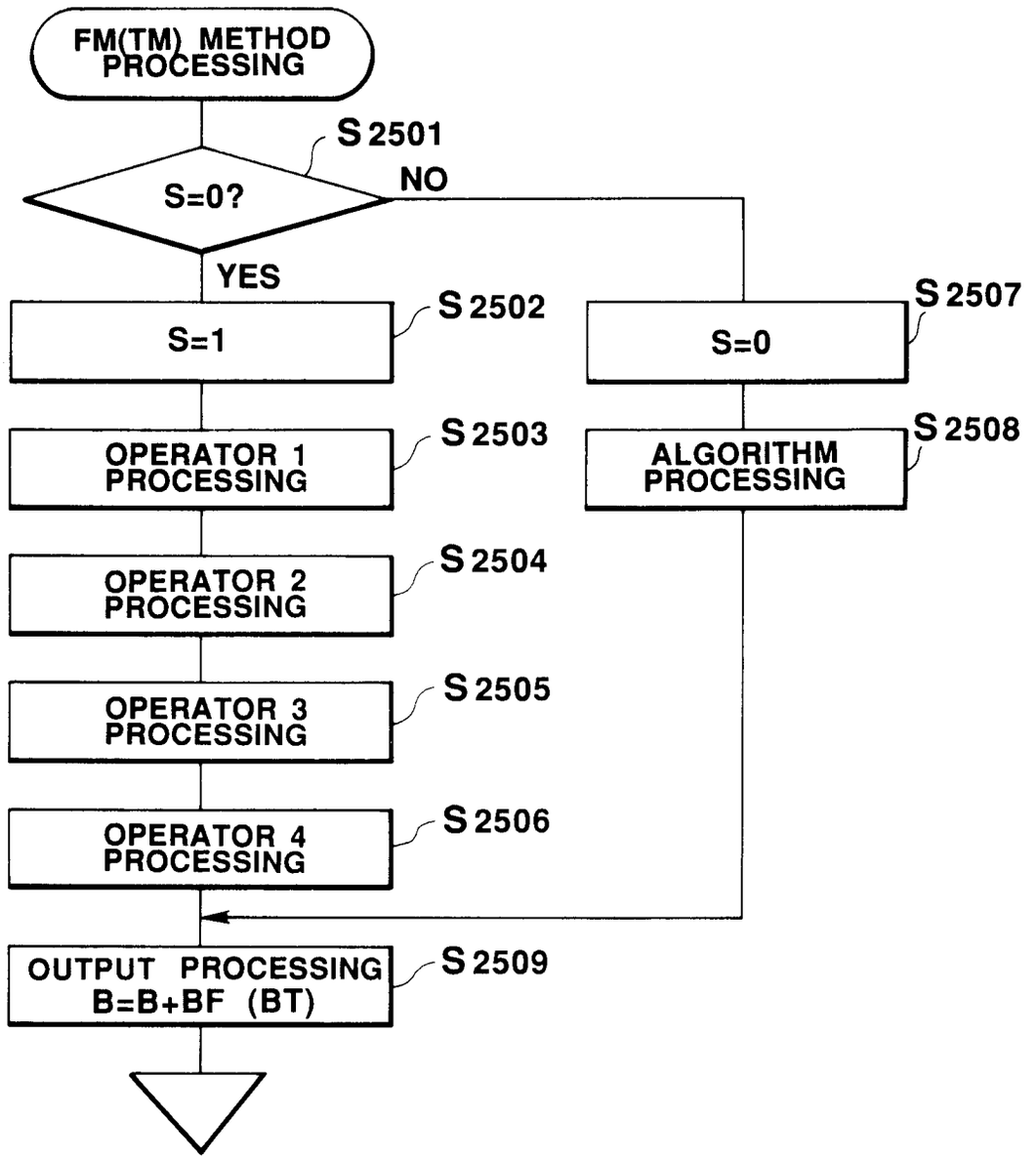
FM(TM) METHOD PROCESSING
S 2501
S=0?
NO
YES
S=1
S 2502
S=0
S 2507
OPERATOR 1 PROCESSING
S 2503
ALGORITHM PROCESSING
S 2508
OPERATOR 2 PROCESSING
S 2504
OPERATOR 3 PROCESSING
S 2505
OPERATOR 4 PROCESSING
S 2506
OUTPUT PROCESSING B=B+BF (BT)
S 2509

FIG. 63

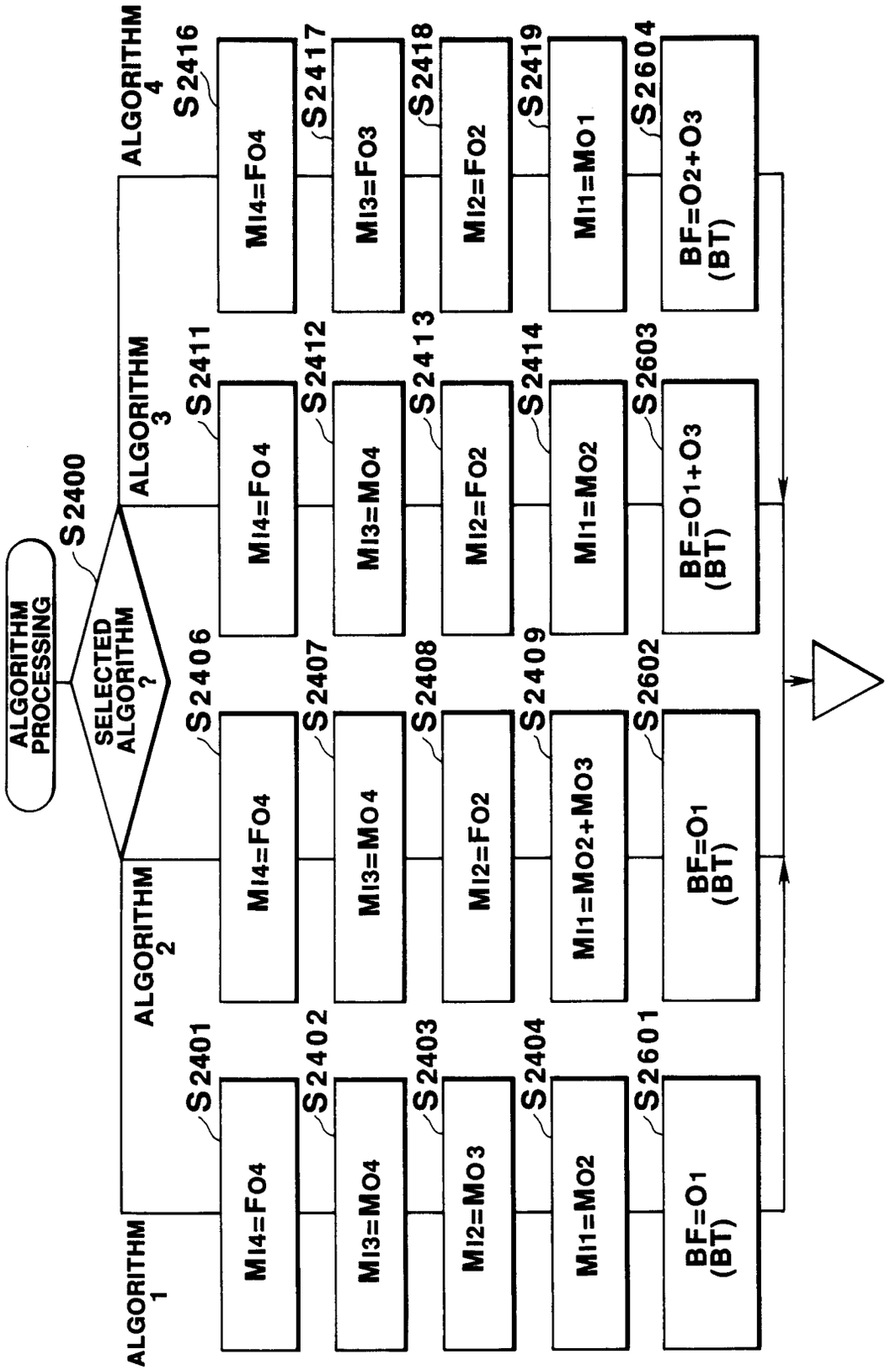
ALGORITHM PROCESSING
S2400
SELECTED ALGORITHM ?
ALGORITHM 1
ALGORITHM 2
ALGORITHM 3
ALGORITHM 4
S2401
S2402
S2403
S2404
S2406
S2407
S2408
S2409
S2411
S2412
S2413
S2414
S2416
S2417
S2418
S2419
S2601
S2602
S2603
S2604
MI4=FO4
MI3=MO4
MI2=MO3
MI1=MO2
MI4=FO4
MI3=MO4
MI2=FO2
MI1=MO2+MO3
MI4=FO4
MI3=MO4
MI2=FO2
MI1=MO2
MI4=FO4
MI3=FO3
MI2=FO2
MI1=MO1
BF=O1 (BT)
BF=O1 (BT)
BF=O1+O3 (BT)
BF=O2+O3 (BT)

FIG.64

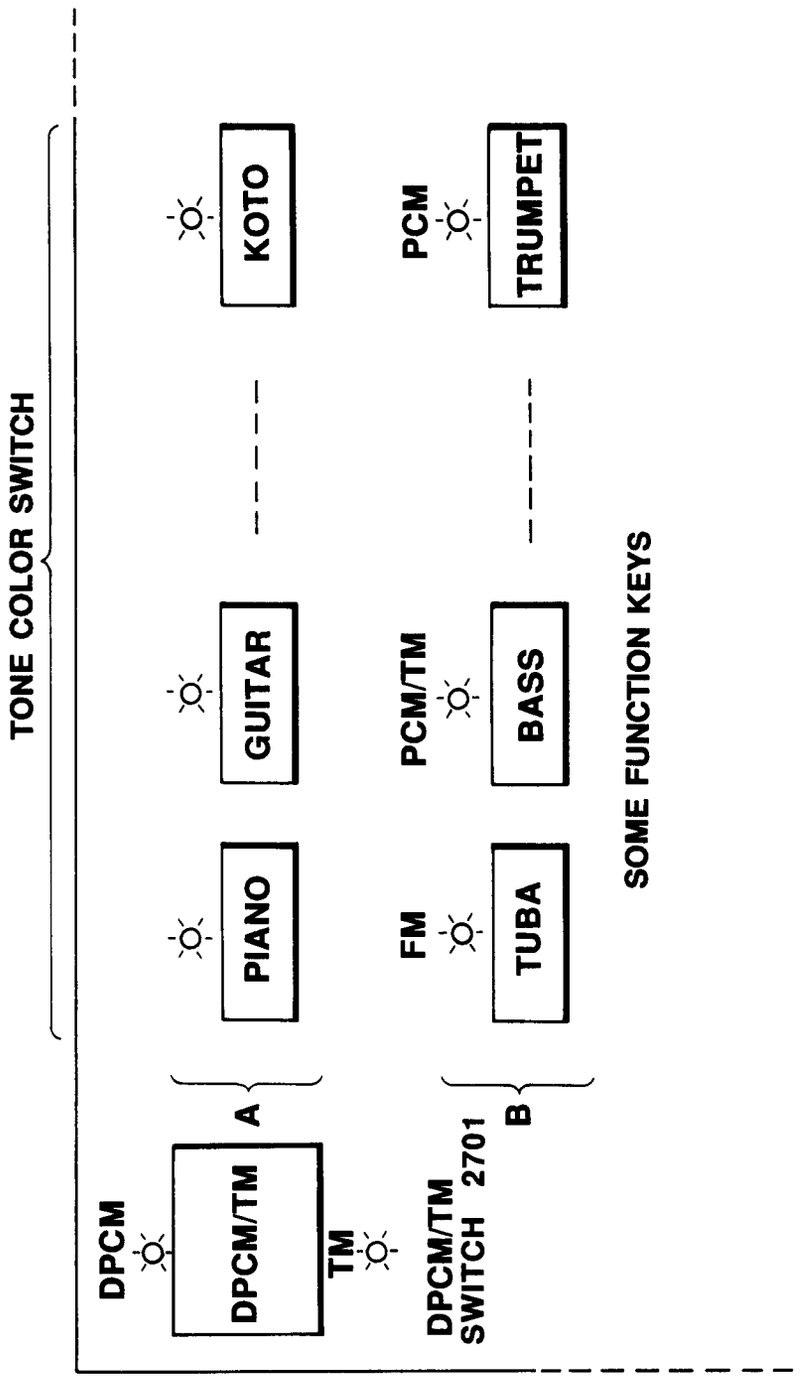
TONE COLOR SWITCH
DPCM
DPCM/TM
TM
DPCM/TM SWITCH 2701
A
B
PIANO
GUITAR
FM
PCM/TM
TUBA
BASS
SOME FUNCTION KEYS
PCM
KOTO
TRUMPET

FIG.65

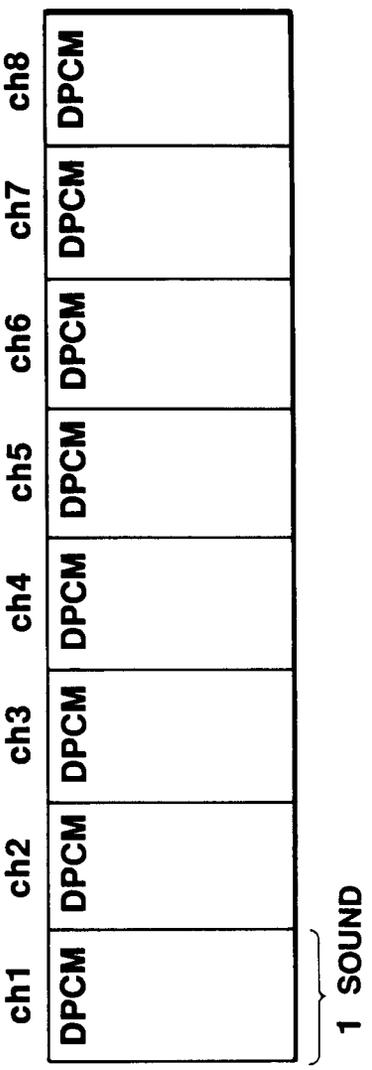
ch1
ch2
ch3
ch4
ch5
ch6
ch7
ch8
DPCM
DPCM
DPCM
DPCM
DPCM
DPCM
DPCM
DPCM
1 SOUND

FIG.66

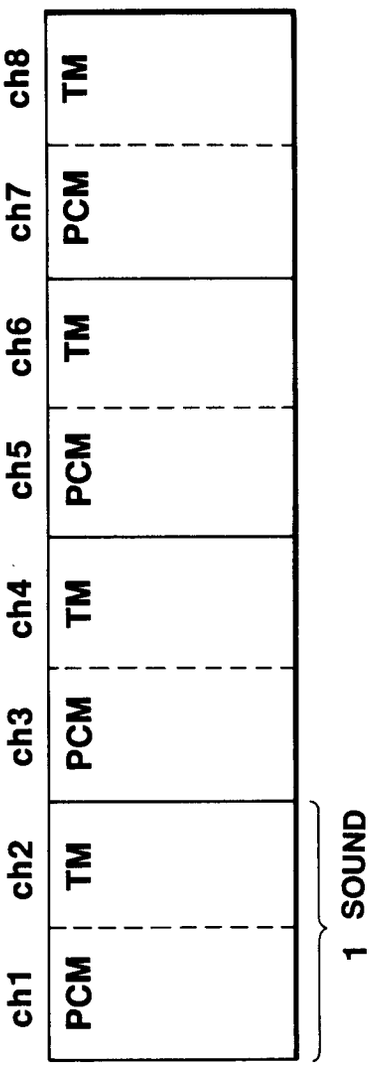
ch1
ch2
ch3
ch4
ch5
ch6
ch7
ch8
PCM
TM
PCM
TM
PCM
TM
PCM
TM
1 SOUND

FIG.67

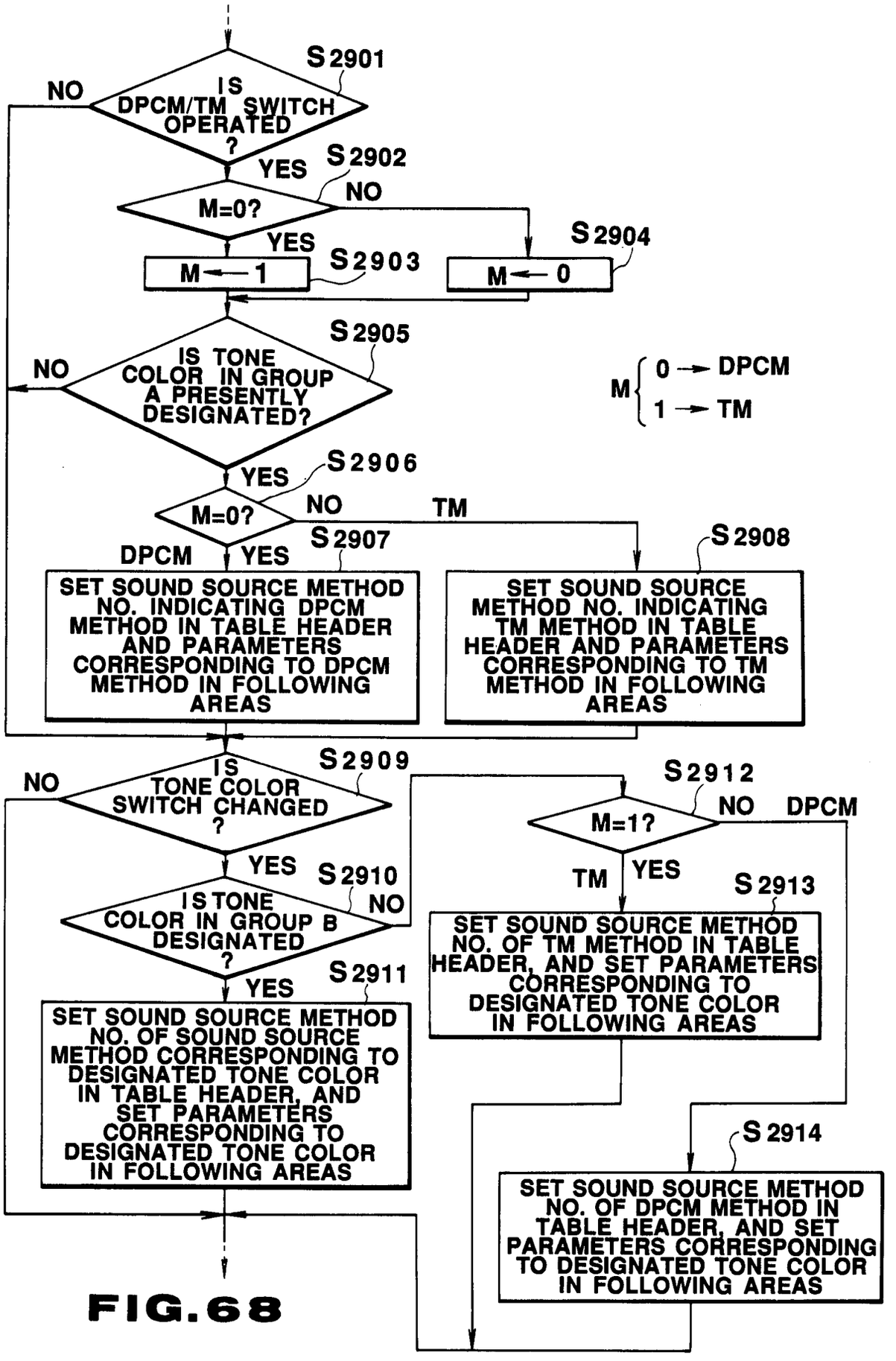
S2901
IS DPCM/TM SWITCH OPERATED ?
NO
YES
S2902
M=0?
NO
YES
S2903
M ← 1
S2904
M ← 0
S2905
IS TONE COLOR IN GROUP A PRESENTLY DESIGNATED?
NO
M { 0 → DPCM
1 → TM
YES
S2906
M=0?
NO
TM
DPCM
YES
S2907
SET SOUND SOURCE METHOD NO. INDICATING DPCM METHOD IN TABLE HEADER AND PARAMETERS CORRESPONDING TO DPCM METHOD IN FOLLOWING AREAS
S2908
SET SOUND SOURCE METHOD NO. INDICATING TM METHOD IN TABLE HEADER AND PARAMETERS CORRESPONDING TO TM METHOD IN FOLLOWING AREAS
S2909
IS TONE COLOR SWITCH CHANGED ?
NO
YES
S2910
IS TONE COLOR IN GROUP B DESIGNATED ?
NO
YES
S2911
SET SOUND SOURCE METHOD NO. OF SOUND SOURCE METHOD CORRESPONDING TO DESIGNATED TONE COLOR IN TABLE HEADER, AND SET PARAMETERS CORRESPONDING TO DESIGNATED TONE COLOR IN FOLLOWING AREAS
S2912
M=1?
NO
DPCM
TM
YES
S2913
SET SOUND SOURCE METHOD NO. OF TM METHOD IN TABLE HEADER, AND SET PARAMETERS CORRESPONDING TO DESIGNATED TONE COLOR IN FOLLOWING AREAS
S2914
SET SOUND SOURCE METHOD NO. OF DPCM METHOD IN TABLE HEADER, AND SET PARAMETERS CORRESPONDING TO DESIGNATED TONE COLOR IN FOLLOWING AREAS

FIG.68

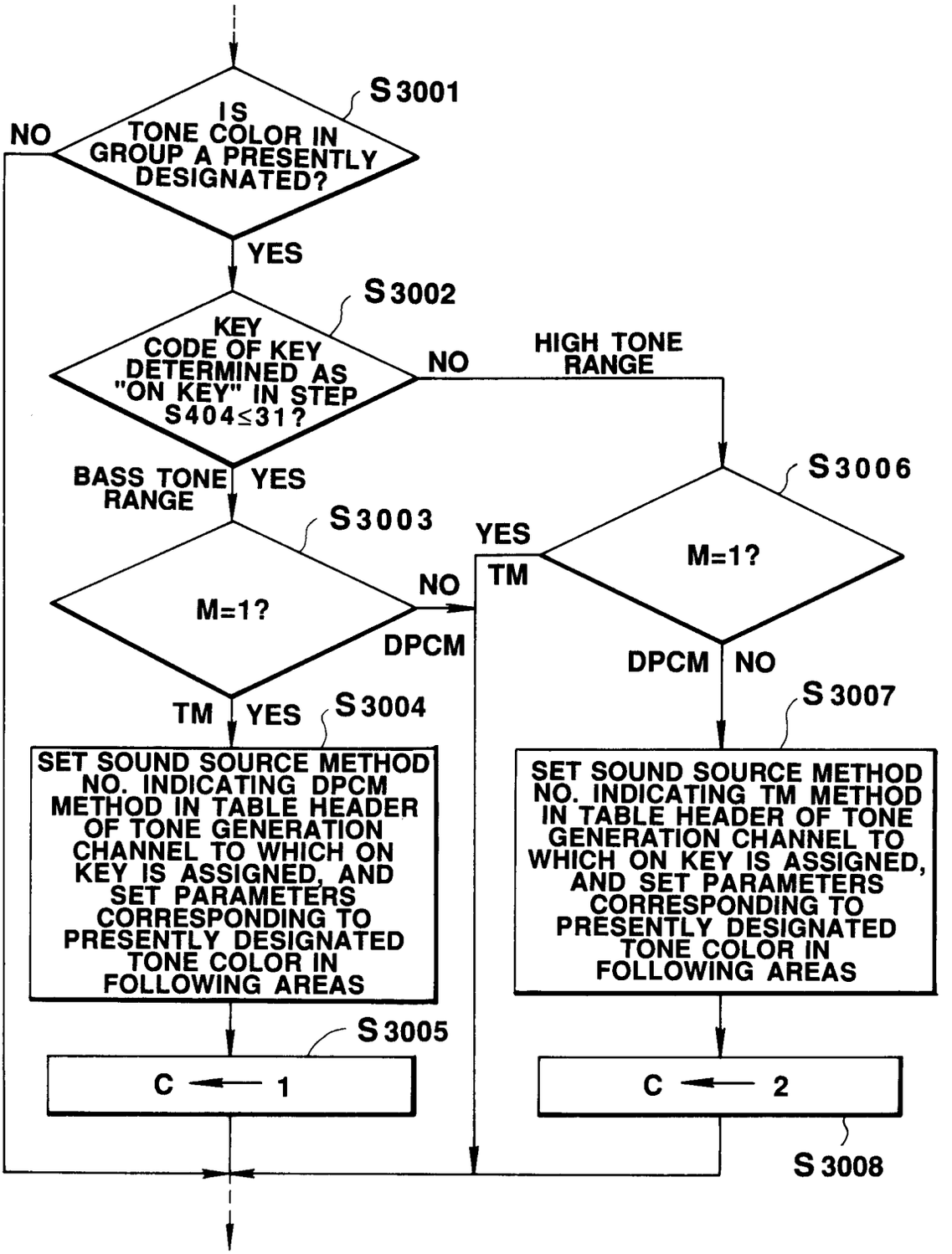
S3001
IS TONE COLOR IN GROUP A PRESENTLY DESIGNATED?
NO
YES
S3002
KEY CODE OF KEY DETERMINED AS "ON KEY" IN STEP S404≤31?
NO
HIGH TONE RANGE
BASS TONE RANGE
YES
S3003
M=1?
NO
DPCM
TM
YES
S3004
SET SOUND SOURCE METHOD NO. INDICATING DPCM METHOD IN TABLE HEADER OF TONE GENERATION CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO PRESENTLY DESIGNATED TONE COLOR IN FOLLOWING AREAS
S3005
C ← 1
S3006
M=1?
YES
TM
DPCM
NO
S3007
SET SOUND SOURCE METHOD NO. INDICATING TM METHOD IN TABLE HEADER OF TONE GENERATION CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO PRESENTLY DESIGNATED TONE COLOR IN FOLLOWING AREAS
C ← 2
S3008

FIG.69

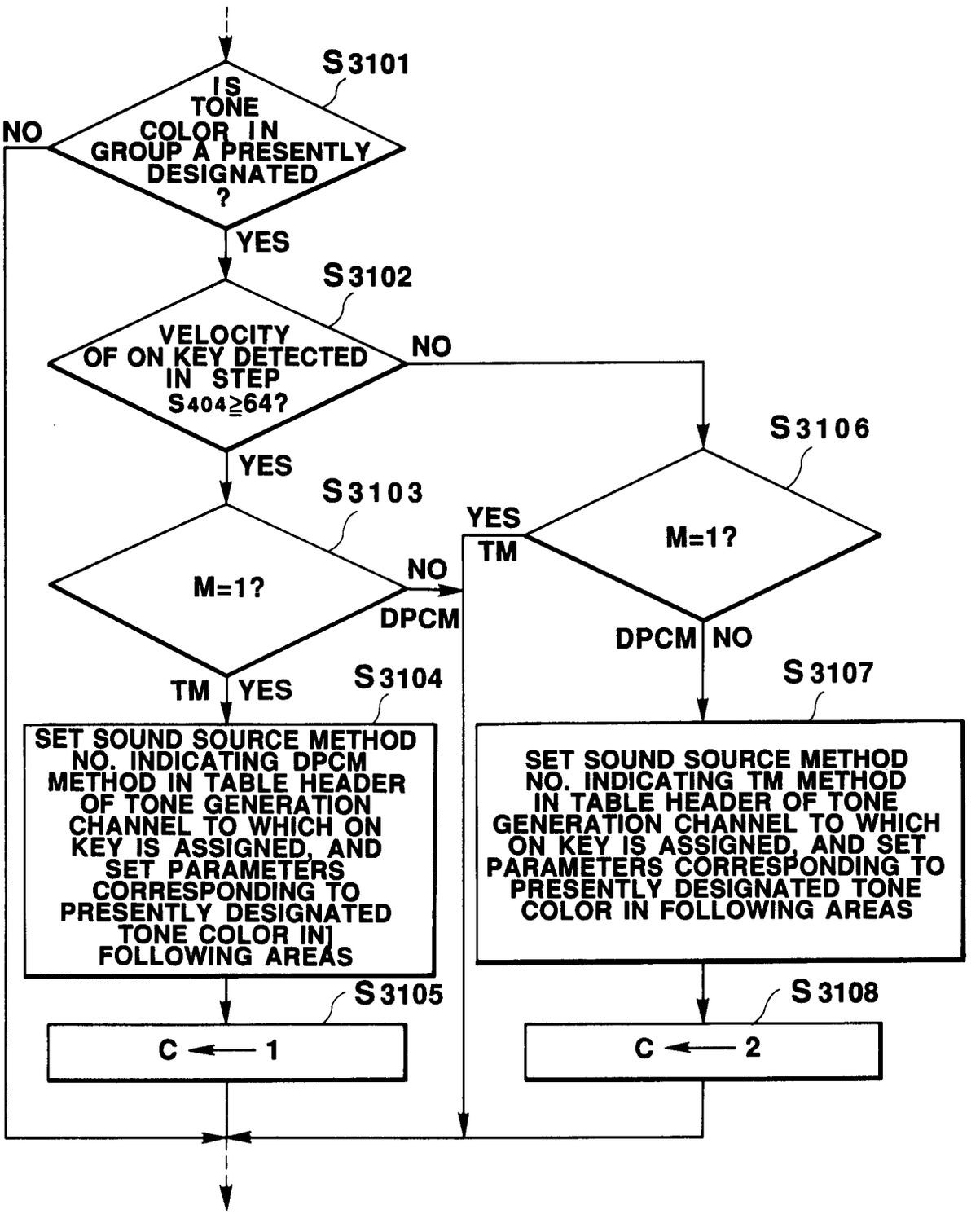
S3101
IS TONE COLOR IN GROUP A PRESENTLY DESIGNATED ?
NO
YES
S3102
VELOCITY OF ON KEY DETECTED IN STEP S404≧64?
NO
YES
S3103
M=1?
NO
DPCM
TM
YES
S3104
SET SOUND SOURCE METHOD NO. INDICATING DPCM METHOD IN TABLE HEADER OF TONE GENERATION CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO PRESENTLY DESIGNATED TONE COLOR IN] FOLLOWING AREAS
S3105
C ← 1
S3106
M=1?
YES
TM
DPCM
NO
S3107
SET SOUND SOURCE METHOD NO. INDICATING TM METHOD IN TABLE HEADER OF TONE GENERATION CHANNEL TO WHICH ON KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO PRESENTLY DESIGNATED TONE COLOR IN FOLLOWING AREAS
S3108
C ← 2

FIG.70

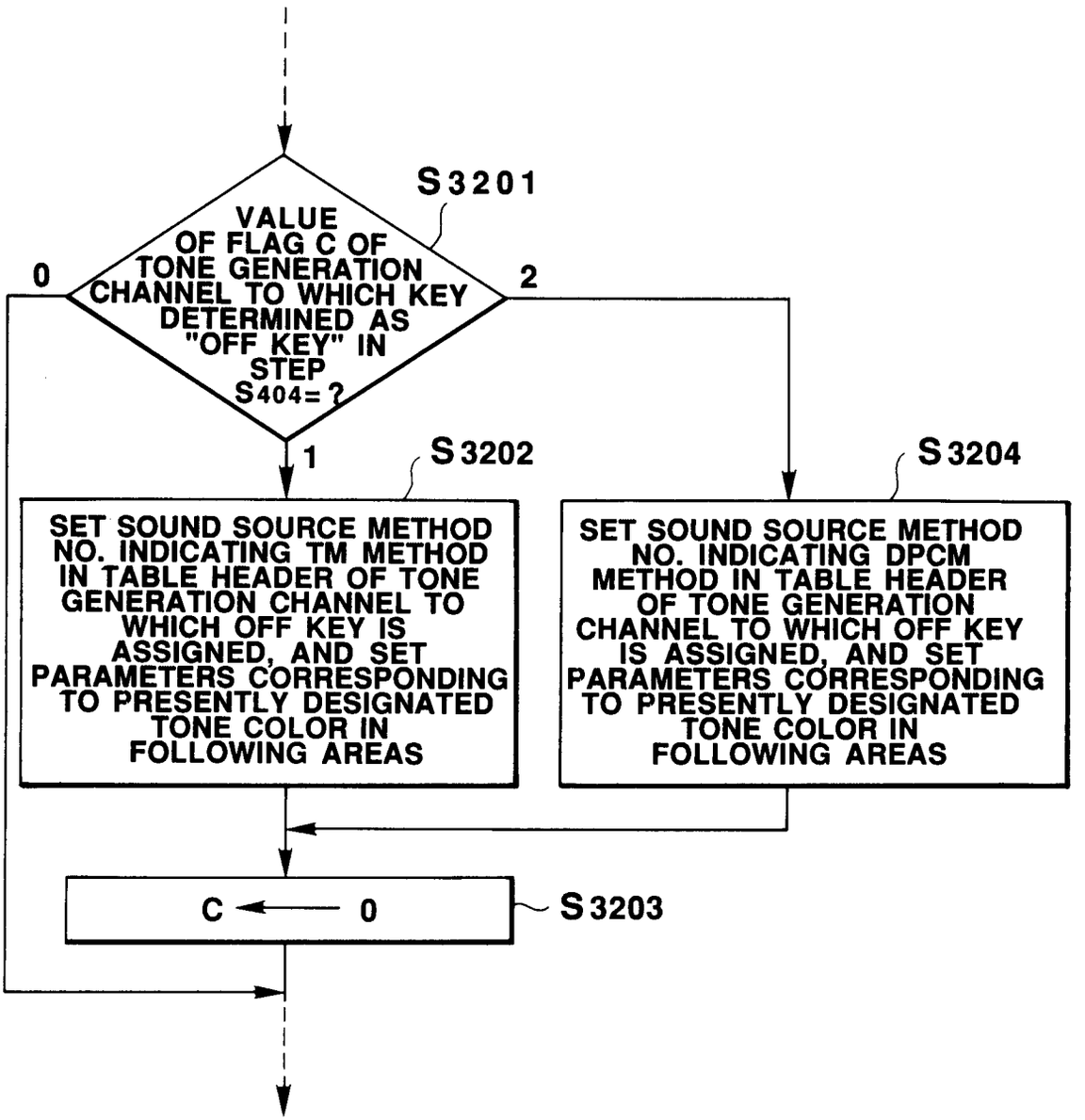
S3201
VALUE OF FLAG C OF TONE GENERATION CHANNEL TO WHICH KEY DETERMINED AS "OFF KEY" IN STEP S404=?
0
1
2
S3202
SET SOUND SOURCE METHOD NO. INDICATING TM METHOD IN TABLE HEADER OF TONE GENERATION CHANNEL TO WHICH OFF KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO PRESENTLY DESIGNATED TONE COLOR IN FOLLOWING AREAS
S3204
SET SOUND SOURCE METHOD NO. INDICATING DPCM METHOD IN TABLE HEADER OF TONE GENERATION CHANNEL TO WHICH OFF KEY IS ASSIGNED, AND SET PARAMETERS CORRESPONDING TO PRESENTLY DESIGNATED TONE COLOR IN FOLLOWING AREAS
C ← 0
S3203

FIG.71