PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G10H 1/12 — A1

(11) International Publication Number: WO 96/31867

(43) International Publication Date: 10 October 1996 (10.10.96)

(21) International Application Number: PCT/IB96/00460

(22) International Filing Date: 5 April 1996 (05.04.96)

(30) Priority Data:
08/418,957 7 April 1995 (07.04.95) US

(71) Applicant: CREATIVE TECHNOLOGY LTD. [SG/SG]; Crescent #03-18, 67 Ayer Rajah, Singapore 139950 (SG).

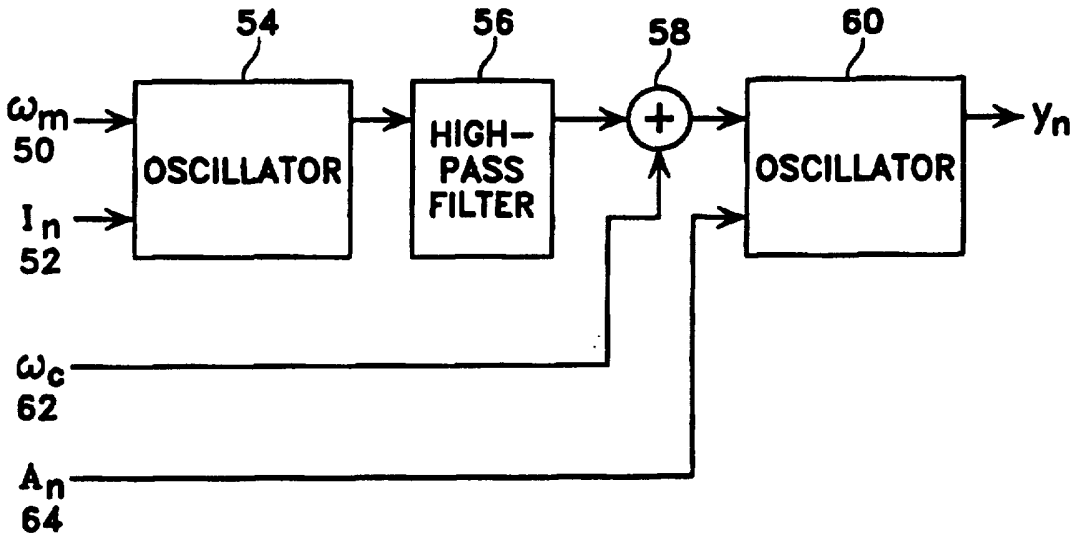
(72) Inventor: ROSSUM, David, P.; 102 Las Lomas Drive, Aptos, CA 95003-3221 (US).

(74) Agents: SALTZBERG, Robert, A. et al.; Morrison & Foerster L.L.P., 755 Page Mill Road, Palo Alto, CA 94304-1018 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD AND APPARATUS FOR SYNTHESIZING MUSICAL SOUNDS BY FREQUENCY MODULATION USING A FILTER



(57) Abstract

In one embodiment, the circuit includes a first order FIR highpass filter (56) that is placed between a modulation phase increment oscillator (54) and a carrier phase increment oscillator (60). The invention may also include waveshaping circuits, adders, multipliers, time division multiplexing, and other types of filters. Another embodiment of the invention provides a music synthesis method where a modulation phase increment is multiplied by a modulation index to produce a modulation signal. That modulation signal is then filtered and added to a carrier phase increment. Finally, that sum is multiplied by an amplitude envelope to produce a signal representing a musical sound. The method may include using a first order FIR highpass filter, waveshaping of both sinusoidal and non-sinusoidal waveforms, additional multiplying and time division multiplexing. The invention also includes self-modulation and cascading.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia
AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgystan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakhstan
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam

METHOD AND APPARATUS FOR SYNTHESIZING MUSICAL SOUNDS BY FREQUENCY MODULATION USING A FILTER

Field of the Invention

This invention relates to a method and apparatus for synthesizing musical sounds by frequency modulation using a filter, and in particular, a highpass filter.

Background of the Invention

Frequency modulation has become a popular technique for synthesizing musical sounds in applications where higher fidelity can be traded off for lower cost. Using frequency modulation to synthesize musical sounds ("FM synthesis") was first described by John Chowning in his landmark 1973 article "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *Journal of the Audio Engineering Society,* vol. 21, no. 7, pp. 526-535. Chowning also described a particular frequency modulation technique in his article, which he pictured schematically as a MUSIC V "patch," or software code that can be patched into the MUSIC V sound synthesis program.

In his article, Chowning gave a mathematical formula for his frequency modulation technique, and he later obtained a patent, U.S. Patent No. 4,018,121, based on that mathematical formula. Interestingly, although Chowning did not mention it, the formula he gave for his frequency modulation technique does not correspond to his MUSIC V patch. This has caused some confusion, since implementations based on the formula will sometimes give incompatible results to implementations based on the MUSIC V patch, even though it was thought that the implementations were the same. This confusion has taken some years to sort out.

Frode Holm gives perhaps the best summary of the resulting confusion and its resolution in his article "Understanding FM Implementations: A Call for Common Standards," *Computer Music Journal,* vol. 16, no. 1, pp. 34-42 (1992). Holm concludes that there are actually two distinct methods of FM synthesis, one using Chowning's formula and one using his particular MUSIC V patch implementation. Holm's analysis shows, mathematically, that Chowning's formula actually implements phase modulation, while the MUSIC V patch is "true" frequency modulation.

The respective mathematical formulas best illustrate the difference between Chowning's FM synthesis formula and his actual MUSIC V patch. Chowning's FM synthesis formula is:

$$E(t) = A(t)\sin(\omega_c t + I(t)\sin(\omega_m t)) \qquad 1$$

where E(t) is the instantaneous amplitude of the synthesized musical note, ω_c is the carrier frequency, I(t) is the time varying modulation index, and ω_m is the modulation frequency. As is evident to those skilled in the art, I(t), ω_c and ω_m can be varied with time to create the desired "tone color" for the basic tone pitch provided by ω_c.

In contrast, the MUSIC V patch presented by Chowning, expressed as a formula, is:

$$E(t) = A(t)\sin([\omega_c + I(t)\omega_m \sin(\omega_m t)]t) \qquad 2$$

Note that the differences between the two equations are that the modulation term is now multiplied by the factor ω_m and that the entire argument of the outer sine function, rather than just ω_c, is now multiplied by the factor t. Because of these differences, the two approaches give incompatible results in some cases.

It should be noted that although these equations are helpful in understanding the theoretical basis for the two approaches, they only approximate the operation of the actual implementations of the Chowning formula and the MUSIC V patch. The operation of the actual implementations approaches the formulas above only as the sampling period used in the oscillators approaches zero. Moreover, the above formulas apply only when the waveform of the outputs of both the carrier and

modulating oscillators are simple sine waves. When they are not, as is often the case, the actual operation of the implementation deviates from the applicable formula.

Following Chowning's article, a number of people designed real-time synthesizers using the techniques described by Chowning. Many designed implementations based on the MUSIC V patch. Most of these implementations used phase increment oscillators, which used as their frequency or "phase increment" input the sum of a static (or slowly time varying) frequency parameter and one or more audio rate "frequency modulation" inputs. Others designed analog implementations using analog voltage-controlled oscillators to implement the MUSIC V patch. Both the analog and digital implementations expanded on the Chowning article, demonstrating such features as multiple modulators and carriers, cascaded FM oscillators, and self-modulation.

Implementations based on the MUSIC V patch work quite well for music synthesis under very limited circumstances. If ω_m is essentially fixed (time varying only at very slow rates), I(t) can simply be appropriately scaled by the controlling computer without substantial extra computation. Furthermore, as shown above, the common factor "t" in the argument of the carrier sine function allows the "t" to be factored out of both terms. This then allows the output of the modulation oscillator, $I(t)\omega_m \sin(\omega_m t)$, to be added directly to the carrier frequency ω_c before applying this sum to the carrier oscillator phase increment input. Hence, this is truly frequency modulation, and the carrier oscillator need only have a single frequency input.

Unfortunately, if ω_m is time varying at a substantial rate, as is common in certain applications, then I(t) must be scaled in a rapidly time varying manner in order to produce reasonably sounding results. Because the number of multiplications per second required to accomplish this is substantial, implementations based on the MUSIC V patch become costly to implement.

Also, the MUSIC V patch itself gives similar audible results to the Chowning formula only when the modulator output waveform is nominally a sinusoid. If waveforms with substantial harmonic content are included, such as the commonplace sawtooth or square waveforms, some deviation occurs between the two

formulas for each sinusoidal harmonic component of the waveform. Specifically, in such cases, the *i*th sinusoidal harmonic component in the MUSIC V patch must be multiplied by its own effective ω_{mi} to generate audibly similar results to the Chowning formula. Once the sinusoids are combined, as with a square or sawtooth waveform, this is impractical to do.

To avoid additional computations, most implementations based on the MUSIC V patch simply multiply by the ω_m for a nominal sinusoid, rather than multiplying each sinusoidal harmonic component by its own ω_m. These implementations, therefore, give results that are not compatible with implementations based directly on the Chowning formula in cases where the modulator output is not nominally a sinusoid.

Accordingly, existing FM synthesis implementations based on the MUSIC V patch have some disadvantages. They require numerous multiplications, which are costly. They can also give incompatible results, particularly in the case of sawtooth and square waveforms. However, because they do not require two frequency inputs to the carrier phase increment oscillator, they do have some advantages over direct implementations of the Chowning formula.

In contrast to the MUSIC V patch, when implemented directly the Chowning formula performs "phase modulation" instead of true frequency modulation. Because a frequency integrated over time is a phase, the term $I(t)\sin(\omega_m t)$ in Chowning's formula is actually being added to a phase, $\omega_c t$, rather than a frequency. Moreover, this addition is done within the carrier phase increment oscillator, rather than as a distinct addition operation performed prior to the frequency being input to that oscillator, as is the case in true frequency modulation. Accordingly, as Holm points out, a direct implementation requires two distinct inputs into the carrier phase increment oscillator, one input representing the "static" frequency and the other a "phase modulation" increment value.

Like those based on the MUSIC V patch, existing phase modulation implementations based directly on the Chowning formula also have some disadvantages. They require a phase increment oscillator with both a frequency and a

phase input, which adds complexity, and thus expense, to the circuitry required to implement the oscillator.

None of the existing implementations of either Chowning approach combines the advantages of the two approaches in a way that achieves reasonably sounding results that are audibly compatible. In particular, none combines the simplicity of the MUSIC V phase increment oscillator, with its single frequency input, with the minimal number of multiplications that can be achieved by the mathematically simpler phase modulation based directly on Chowning's formula. Moreover, unless a number of multiplications are done, none of the implementations based on the MUSIC V patch give audibly similar results for all types of waveforms, both sinusoidal and not, to the results of implementations based directly on Chowning's formula.

Summary of the Invention

This invention provides a new circuit and method for synthesizing musical sounds. In one embodiment, the invention includes a first order FIR highpass filter that is placed between a modulation phase increment oscillator and a carrier phase increment oscillator. The invention may also include waveshaping circuits, multipliers, time division multiplexing, and other types of filters.

Another embodiment of the invention provides a music synthesis method where a modulation phase increment is multiplied by a modulation index to produce a modulation signal. That modulation signal is then filtered and added to a carrier phase increment. Finally, that sum is multiplied by an amplitude envelope to produce a signal representing a musical sound. The method may include using a first order FIR highpass filter, waveshaping of both sinusoidal and non-sinusoidal waveforms, additional multiplying and time division multiplexing.

The invention also includes self-modulation and cascading.

Brief Description of the Drawings

The objects, features and advantages of the present invention will be apparent to one skilled in the art in light of the following detailed description in which:

Figure 1 shows a signal flow diagram of a known phase increment oscillator.

Figure 2 shows graphically the basis for creating the eight standard "OPL3" waveforms using a waveshaping circuit based on quadratic splines.

Figure 3 shows graphically the creation of the eight standard "OPL3" waveforms using a waveshaping circuit based on quadratic splines.

Figure 4 shows a block diagram of a hardware implementation of a waveshaping circuit based on quadratic splines.

Figure 5 shows the relationship of signals in a waveshaping circuit based on quadratic splines for the eight "OPL3" waveforms.

Figure 6 shows a signal flow diagram of a known first order FIR highpass filter.

Figure 7 shows a signal flow diagram of an embodiment of the present invention.

Figure 8 shows a signal flow diagram of the present invention in self-modulated form.

Figure 9 shows a signal flow diagram of the present invention in cascaded form.

Figure 10 shows a block diagram of the present invention implemented in multichannel real time hardware.

Detailed Description of the Invention

Before the present methods and apparatuses are described, it is to be understood that this invention is not limited to the particular apparatuses or methods described as such, which those of skill in the art can, of course, vary. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to be limiting as to the scope of the present invention, which will be limited only by the appended claims.

It should be noted that, as used in this specification and the appended claims, the singular forms "a", "an" and "the" include the plural referents unless the context clearly dictates otherwise.

Unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this invention belongs. Although any methods and materials similar or equivalent to those described herein can be useful in the practice or testing of the present invention, preferred methods and materials are described below. All publications and patents mentioned herein are incorporated herein by reference.

This invention provides a method and apparatus for replacing the multiplication by ω_m of the modulator output in a true frequency modulation implementation with a highpass filter. Rather than explicitly multiply the modulator output by ω_m as suggested by Chowning in his article, the modulator output instead passes through a first order highpass filter with a cutoff slope of 6 dB per octave. Such a cutoff slope means that for each octave of increase in modulator frequency, the amplitude is doubled.

As will be evident to those skilled in the art, this highpass filtering operation is in principle the same as multiplying each sinusoidal harmonic component of the oscillator output by its own frequency. Thus, it accomplishes the desired multiplication of each harmonic component of a spectrally rich waveform by the appropriate factor.

Accordingly, this invention can be used to implement the Chowning MUSIC V patch formula while eliminating the troublesome ω_m multiplication. It also

allows for rapidly time varying modulator frequencies without the computational burden of frequent scaling of I(t). Finally, the present invention produces results audibly compatible with the Chowning formula, yet still requires only a single frequency input into the carrier phase increment oscillator. This combines the simplicity of true "frequency modulation," using a simple MUSIC V-type phase increment oscillator having a single frequency input, with the computational benefits of the "phase modulation" technique.

Various embodiments of the present invention are illustrated in Figures 1 through 10. These embodiments will first be described in terms of conventional signal flow diagrams.

Figure 1 shows a flow diagram for a simple phase increment oscillator. The actual MUSIC V oscillators, as well as those of most subsequent real-time synthesizers, were implemented as "phase increment" oscillators. While many variations of this oscillator exist, including in particular numerous connection topologies for implementing various FM patches, the fundamental core of the oscillator remains unchanged.

In a phase increment oscillator, a phase value is stored in a register or memory, and at each successive sample period it is incremented by a phase increment, which represents the instantaneous frequency of the oscillator. As will be easily seen by those skilled in the art, if the phase increment (ω_n) input 10 is a constant much less than 2π, the signal at the output of the modulo operator 14 will be a "sawtooth" waveform increasing slowly with constant slope from zero to 2π, then jumping suddenly back to zero to begin rising again. Hence this signal is commonly referred to as a "phase sawtooth."

Although phase increment oscillators are often used to generate standard phase sawtooth signals by using a constant input, they can also accept time varying phase angle inputs. In either case, the resulting phase output can then be transformed into a sine waveform (or any other waveform) by a variety of techniques. In MUSIC V, for example, a lookup table was used to convert the phase sawtooth into a sine wave.

As is well known to those skilled in the art, phase increment oscillators can be implemented in a variety of ways. In MUSIC V, they were implemented as software running on a computer. In this specification, they are described as implemented in time division multiplexed circuitry. The scope of the present invention should not be limited to any particular implementation of a phase increment oscillator.

In the phase increment oscillator shown in Figure 1, an adder 12 adds a phase increment (ω_n) input 10 to the previous phase during each sample period. A modulo operator 14 takes the result modulo 2π. A delay operator 22 then stores this new phase until the next sample period, in which the above steps are repeated. A waveshaping circuit 16 also receives this same phase signal, which it uses to produce the desired waveform.

As mentioned above, the waveshaping circuit 16 can be implemented using a ROM lookup table, and a variety of other techniques will also be evident to those skilled in the art. The waveshaping circuit 16 can be designed to produce whatever waveform is desired, sinusoidal or non-sinusoidal, from the output of the modulo operator 14.

One method of generating waveforms based on quadratic splines disclosed in co-pending Patent Application No. 08/418,518, filed on April 7, 1995, can also be used to implement the waveshaping circuit 16 to generate the standard eight waveforms of the Yamaha Corporation "OPL3" synthesizer chip. That quadratic spline method is shown in Figures 2 through 5.

The quadratic spline method performs the function of the waveshaping circuit 16 in Figure 1. Figure 2 shows pictorially the generation, according to the quadratic spline method, of an inverted sine waveform. Although for clarity the example of a standard phase sawtooth being input to the waveshaping circuit 16 is sometimes used in the detailed description of the quadratic spline method, it will be evident to those skilled in the art that the phase angle input need not be limited to a standard phase sawtooth; any phase angle input may be used.

Row 2a of Figure 2 shows several cycles of the standard phase sawtooth, with time varying over the horizontal axis and amplitude varying from -1

to +1 on the vertical axis. Note that the vertical axis has been scaled and a fixed offset added to the standard view of the phase sawtooth varying from 0 to 2π; this is of course of no audible consequence.

Row 2b shows the standard phase sawtooth with a phase offset of π added to it. In other words, the phase sawtooth has been shifted 180 degrees along the horizontal axis. Row 2c shows the absolute value of the signal in Figure 2b. Row 2d shows the signal, in this case simply $y(t)=1$, which according to the quadratic spline method is to be ANDed with Row 2c. Row 2e shows the results of that ANDing. Finally, Row 2f shows the end results of the quadratic spline method, obtained by multiplying the signal in Row 2a with that in Row 2e, which can be seen to approximate an inverted sine waveform of amplitude ranging from -¼ to ¼. This is an inverted form of the first standard "OPL3" waveform, obtained according to the quadratic spline method.

Figure 3 shows pictorially the method, according to the quadratic spline method, for forming each of the eight standard "OPL3" waveforms (Waveforms #0 to #7). Each of the steps of the method shown pictorially in Figure 3 will be discussed in greater detail below when describing the operation of the hardware embodiment of the quadratic spline method.

Column 3a of Figure 3 shows one cycle, from 0 to 2π, of each of the eight waveforms. Column 3b shows the first modification, if any, to the input phase sawtooth required by the quadratic spline method. This first modification results in either the original phase sawtooth (for Waveforms #0 to #3), the original phase sawtooth doubled in frequency (for Waveforms #4 and #5), and in one case, also halved in amplitude (for Waveform #7), or the signum of the original phase sawtooth, also halved in amplitude (for Waveform #6).

Column 3c shows the modified phase sawtooth shifted in phase, if required, and its absolute value taken, if required, in both cases according to the quadratic spline method. Column 3d shows the function which, according to the quadratic spline method, is ANDed with the modified phase sawtooth of column 3c, and column 3e shows the results of the ANDing of columns 3c and 3d. Finally, column 3f shows the results of the final step of the quadratic spline method,

multiplying column 3b by column 3e. Note that the vertical scale of column 3e is from -¼ to ¼, while the vertical scale of the other columns is -1 to 1.

From Figure 3, it can be seen that all eight standard "OPL3" waveforms can be approximated according to the quadratic spline method. This is accomplished by appropriately combining, according to the quadratic spline method, the steps of modifying the original phase sawtooth input, shifting the result in phase, taking an absolute value, ANDing the result with certain phase sawtooth bits, and finally a single multiplication. Note that the polarity of the approximated waveforms is treated as being insignificant, as it does not affect the sound or harmonic content of the waveform. However, as will be seen below, with minor modifications to the circuitry described even the polarity can be corrected, if desired.

Figure 4 shows a detailed hardware implementation of the quadratic spline method. A phase angle input 300 provides an input to both a multiplexer/shifter 304 and control logic 314. The phase angle input 300 is a 16-bit unsigned value representing the phase taken modulo 2π. Thus, a phase of zero is hexadecimal 0000, and a phase of almost 2π is hexadecimal FFFF.

The multiplexer/shifter 304 is a multiplexer wired as a modified barrel shifter. The control logic 314 drives the multiplexer/shifter 304 through a control signal 316. In this embodiment, the control signal has two bits for representing the four possible multiplexer/shifter functions. However, as will be evident to those skilled in the art, the control signal 316 can have more than two bits if desired to optimize the logic of the circuit.

The steps for generating each of the standard "OPL3" waveforms will now be described. First, the multiplexer/shifter 304 operates on the phase angle input 300, as described in detail below. The output signal of the multiplexer/shifter 304 for each waveform is shown in column 3b of Figure 3.

In Figure 4, when the control logic 314 sends a control signal 316 of binary 00 to the multiplexer/ shifter 304, the multiplexer/shifter 304 outputs a 16-bit signal identical to the 16-bit phase angle input 300 it received. This produces the output signal shown in rows #0 to #3 of column 3b of Figure 3.

When the control signal 316 to the multiplexer/shifter 304 is binary 01, it shifts the 16-bit phase angle input 300 left one bit, shifts off and ignores the most significant bit ("MSB"), sets the new least significant bit ("LSB") to 0, and inverts the new MSB. Mathematically, this is equivalent to adding $\pi/2$ to the phase angle input 300, multiplying the result by two, and then taking the result modulo 2π. This produces the output signal shown in rows #4 and #5 of column 3b.

When the control signal 316 to the multiplexer/shifter 304 is binary 10, it outputs a fixed hexadecimal 3FFF. This produces the output signal shown in row #6 of column 3b.

Finally, when the control signal 316 to the multiplexer/shifter 304 is binary 11, it outputs the fourteen LSBs of the 16-bit phase angle input 300 unchanged, and sets the two MSBs of the output signal both to the inverse of the next to the most significant bit, i.e. bit 14, of the original input signal. In other words, it outputs the fifteen LSBs of the 16-bit phase angle input 300 plus $\pi/2$, sign extended. This produces the output signal shown in row #7 of column 3b.

Next, a bank 318 of exclusive OR gates further modifies the 16-bit output signal of the multiplexer/shifter 304. The exclusive OR bank 318 consists of two sections. The first section of exclusive OR gates 306 acts only on the MSB of the multiplexer/shifter 304 output signal, while the second section of exclusive OR gates 308 acts on the other fifteen LSBs.

The exclusive OR bank 318 performs two functions, phase shifting and a functional approximation to the absolute value function, or a combination of both, or neither (a pass-through). The output signal of the exclusive OR bank 318 for each waveform is shown in column 3c of Figure 3.

When two control signals 320 and 322 received by both sections of the exclusive OR bank 318 are both a logical "0", no change occurs. This pass-through operation is used to produce part of the output signal shown in rows #6 and #7 of column 3c of Figure 3.

When the first control signal 320 is logical "0" and the second control signal 322 is logical "1", the output signal 324 of the exclusive OR bank 318 is the one's complement of the multiplexer/shifter 304 output signal plus π. In this case, the

one's complement is only one LSB away from the two's complement, which is, to the accuracy required, the same result as obtained by multiplying by -1. Accordingly, taking the one's complement can be used to produce a functional approximation to the absolute value function. This phase-shifting and absolute value operation is used to produce part of the output signal shown in rows #0 and #4 of column 3c.

When the first control signal 320 is a logical "1" but the second control signal 322 is a logical "0", the output signal 324 of the exclusive OR bank 318 is the sum of the multiplexer/shifter 304 output signal and π, since the MSB has significance π. Accordingly, this operation can be used to shift a signal by π. This operation is used to produce all of the output signal shown in rows #1, #2, #3 and #5 of column 3c, and part of the output signal shown in rows #0 and #4.

When the two control signals 320 and 322 are both a logical "1", the output signal 324 of the exclusive OR bank 318 is the one's complement of the output signal of the multiplexer/shifter 304. As discussed above, this operation is a functional approximation to the absolute value function. This operation is used to produce part of the output signal shown in rows #6 and #7 of column 3c of Figure 3.

Next, a bank 310 of AND gates further modifies the 16-bit output signal 324 of the exclusive OR bank 318. A control signal 326 to the AND bank 310 can force its 16-bit output signal to hexadecimal 0000, or leave it unchanged. This performs the ANDing of each of the signals shown in column 3c of Figure 3 with the corresponding signal shown in column 3d, with the output signal of the bank 310 for each waveform shown in column 3e.

It should be noted that all of the Boolean logic described up to this point is parallel in nature, and does not require any addition operations or other logic, such as a carry chain, that requires each higher order bit to be processed as a result of logical operations on lower order bits. Thus, the processing of this data involves only a few gate delays and can be accomplished within a single clock cycle.

A 16-bit by 16-bit signed two's complement multiplier 312 then receives both the 16-bit output signal of the multiplexer/shifter 304, unmodified (as shown in column 3b of Figure 3), and the 16-bit output signal of the AND bank 310 (as shown in column 3e), and multiplies them together. Because most current audio

applications use just 16-bit signals, only the sixteen MSBs of the multiplier 312 output signal are needed, so an abbreviated form of the multiplier can be used. The results of this multiplication for each waveform are shown in column 3f of Figure 3. This completes the processing needed to form each of the standard "OPL3" waveforms.

As will be evident to those skilled in the art, depending on the size of the available multiplier, it may be desirable to have either or both of the arguments of the multiplier be less than 16 bits, since that would have only a minor impact on the waveform fidelity. Moreover, it will also be evident that any type of multiplier could be used, such as a full parallel multiplier, a serial multiplier, or a hybrid parallel/serial multiplier, to accomplish this function.

In addition, the multiplier 312 output signal never reaches more than one fourth of its theoretical maximum output value, since the peak values occur when its inputs are each at an absolute value of half of full scale. The multiplier 312 output signal 328 should be scaled to account for this.

As is evident from the above step-by-step description, the four control signals 316, 320, 322 and 326 output by the control logic 314 must be set appropriately to form the eight "OPL3" waveforms. These control signals 316, 320, 322 and 326 are determined by the waveform number 302 and the two MSBs of the phase angle input 300. These control signals 316, 320, 322 and 326 then appropriately control the multiplexer/shifter 304, the exclusive OR bank 318, the AND bank 310 and the multiplier 312 to create the desired waveform from the phase angle input 300.

Based on the waveform number 302 and bits 15 and 14 of the phase angle input 300, the control logic 314 sets the control signals 316, 320, 322 and 326 to the values shown in the truth table, Table 1, below. In Table 1, PHn indicates the nth bit of the phase angle input 300, so that, for example, PH15 is the most significant bit. ! indicates logical complement, and ^ indicates an exclusive OR.

Table 1

Control Signal Truth Table

Waveform Number	Control Signals: 316	320	322	326
#0	00	PH15	!PH15	1
#1	00	1	0	PH15
#2	00	1	0	1
#3	00	1	0	!PH14
#4	01	PH14	!PH14	PH15
#5	01	1	0	PH15
#6	10	PH15	PH15	1
#7	11	!PH14	!PH14	PH15^PH14

Figure 5 shows in graphical form the various steps, described in detail above, for producing each of the eight "OPL3" waveforms. Column 5a shows the output signal of the multiplexer/shifter 304, column 5b shows the output signal 324 of the exclusive OR bank 318, column 5c shows the output signal of the AND bank 310, and column 5d shows the output signal 328 of the multiplier 312. Note that the vertical scale of column 5d is from -¼ to ¼, while the vertical scale of the other columns is -1 to 1.

As can be seen in column 3f of Figure 3 and column 5d of Figure 5, all of the output waveforms, when created as described above, are simply inverted in polarity from the desired "OPL3" waveforms. Inverted polarity has no audible impact on the sounds of these waveforms. However, the precise waveform, correct in phase and polarity, can be produced by taking the complement of the output waveform. This can be done at a variety of locations in further signal processing circuitry, as part of the multiplier 312, or by an additional circuit. As will be evident to those skilled in the art, a one's complementing of the output signal may be sufficient to accomplish this inversion to the accuracy of the approximations herein described.

Figure 4 provides circuitry, or hardware, which embodies the quadratic spline method. However, as will be evident to those skilled in the art, the quadratic spline method can also be embodied in firmware and software.

As can be seen from the detailed description of the quadratic spline method above, it enables each of the standard "OPL3" waveforms to be produced without requiring large amounts of memory or intensive computation. However, as will be evident to those skilled in the art, it does suffer from approximately 2% harmonic distortion. Accordingly, for applications where the waveform can be an approximation, and high accuracy in the waveform is not required, the waveshaping circuit of the quadratic spline method provides advantages over other methods.

Returning to the description of Figure 1, once the waveshaping circuit 16 of Figure 1 has produced its output, multiplier 18 then multiplies the waveshaping circuit 16 output by an amplitude (A_n) input 24 to produce an audio signal (Y_n) output 20 of the desired amplitude and frequency. As will be evident to those skilled in the art, the phase increment (ω_n) input 10 can be time varying. The amplitude (A_n) input 24 can also be time varying, in ways that will be evident to those skilled in the art, to achieve amplitude envelope characteristics such as attack and decay.

Figure 6 shows a flow diagram for a first order FIR highpass filter, which has been known to those skilled in the art for many years. These and other filters are summarized in the April 1985 article "Introduction to Filter Theory," Report No. STAN-M-20 (Stanford University, Center for Computer Research in Music and Acoustics) by Julius O. Smith, reprinted in *Digital Audio Signal Processing: An Anthology,* edited by John Strawn.

In the filter of Figure 6, the delay operator 34 delays the signal (X_n) input 30 by one sample period. A subtractor 32 then subtracts the delayed input from the original signal input 30, resulting in a highpass filtered output (Y_n) 36. Note that this simple highpass filter involves only one addition and no multiplications. Because of its simplicity, it can be cheaply implemented. However, as will be evident to those skilled in the art, a variety of other filters can be used.

Figure 7 shows a flow diagram for an embodiment of the present invention. In this flow diagram, phase increment oscillators like that shown in Figure 1 and a highpass filter like that shown in Figure 6 are both used. A modulation phase increment oscillator 54 like that of Figure 1 receives a modulator phase increment ($\omega_m n$) input 50 and a time varying modulation index (I_n) input 52 and produces an audio rate modulator waveform output. To take a simple example where ω_m is a constant and the modulation phase increment oscillator 54 is configured to produce a simple sinusoidal waveform, then the output of the modulation phase increment oscillator 54 is, approximately, $I(t)\sin(\omega_m t)$.

A highpass filter 56 like that of Figure 6 then filters this output, which effectively multiplies it by its component frequencies. In the simple example given above, the output of the highpass filter 56 is, approximately, $I(t)\omega_m\sin(\omega_m t)$.

Next, an adder 58 then sums the output of the highpass filter 56 with a carrier phase increment (ω_c) input 62. In the simple example given above, the output of the adder is, approximately, $\omega_c+I(t)\omega_m\sin(\omega_m t)$.

Finally, a carrier phase increment oscillator 60, also like that of Figure 1, receives the sum from the adder 58 and a time varying amplitude (A_n) input 64, and produces an audio output from them. That audio output is a signal representing a musical sound, produced by FM synthesis. In the simple example given above, assuming that the carrier phase increment oscillator 60 is configured to produce a simple sinusoidal waveform, its output is, approximately, $A(t)\sin([\omega_c+I(t)\omega_m\sin(\omega_m t)]t)$. As can be seen from Equation (2) above, mathematically this example approximates the same results as Chowning's MUSIC V patch.

If desired, any or all of the modulation phase increment oscillator 54, the highpass filter 56, the carrier phase increment oscillator 60 and the adder 58 can be designed to utilize the same adder circuitry by using time division multiplexing. As those skilled in the art will recognize, although this may decrease the number of logic gates needed, it will probably be at the expense of the speed and efficiency of the circuit. Accordingly, this tradeoff should be considered in designing a circuit of the present invention.

As will also be evident to those skilled in the art, the entire circuit of Figure 7 can be time division multiplexed. Here again, however, there may be a tradeoff between a reduction in the number of logic gates required and the levels of speed and efficiency that can be obtained.

Finally, as mentioned above, the phase increment inputs to a phase increment oscillator can be time varying. Accordingly, in the circuit of Figure 7, either or both of the modulator phase increment (ω_m) input 50 and the carrier phase increment (ω_c) input 62 can be time varying.

Figures 8 and 9 show additional variations of the present invention in flow diagram form. Figure 8 shows "self-modulation". In this case, the same oscillator serves as both the modulation and carrier phase increment oscillators, and the same ω_n serves as both the modulation and carrier phase increments.

In Figure 8, a phase increment oscillator 76 receives an amplitude (A_n) input 82 and a phase increment input, and produces an audio output from them. The phase increment input to the phase increment oscillator 76 is obtained by feeding the output of the phase increment oscillator 76 back into the circuit, once it has been delayed by a delay operator 71. That same audio output is also used as a signal representing a musical sound, produced by FM synthesis. As self-modulation in FM synthesis is well known, it will be evident to those skilled in the art how the present invention can be implemented in a self-modulation topology.

When the output of the phase increment oscillator 76 is fed back into the circuit, it must be delayed at least one unit by the delay operator 71 to avoid the circuit being an endless loop. A feedback shifter 70 (a shifter is a fixed multiplier by 2^n) is then used, by varying a feedback input 78, to produce a signal of the desired magnitude.

A highpass filter 72 then filters the signal, after which an adder 74 then adds the output of the highpass filter 72 to the current value of the phase increment (ω_n) input 80. This sum is then used as the first phase increment (ω_n) input of the phase increment oscillator 76, completing the self modulation loop.

Figure 9 shows multiple cascaded FM oscillators. A first phase increment oscillator 94 receives a first modulator frequency (ω_{m1}) input 90 and a first

modulation index (I_1) input 92, and produces from them an output, which it then passes on to a first highpass filter 96. A first adder 98 adds the output of the first highpass filter 96 to a second modulator frequency (ω_{m2}) input 100.

A second phase increment oscillator 104 then continues the cascaded modulation by receiving the sum from the first adder 98 and a second modulation index (I_2) input 102. A second highpass filter 106 filters the output of the second modulation oscillator 104, after which a second adder 108 adds it to a carrier frequency (ω_c) input 110.

Finally, a third phase increment oscillator 114 produces an audio output from the sum from the second adder 108 and an amplitude (A_n) input 112. That audio output is a signal representing a musical sound, which has been modulated by two modulation frequencies. Expressed as a formula, in a simple example where ω_c, ω_{m1} and ω_{m2} are all constants, and the three phase increment oscillators 94, 104 and 114 are all configured to produce simple sinusoidal waveforms, the audio output is, approximately:

$$A(t)\sin([\omega_c + I_1(t)\omega_m 1\sin(\omega_m 1t) + I_2(t)\omega_m 2\sin(\omega_m 2t)]t).$$

As will be evident to those skilled in the art, this type of cascading can be used to produce particularly complex and spectrally rich waveforms. As will also be evident to those skilled in the art, cascading can be done with as many levels as desired simply by adding modulation phase increment oscillators, adders and highpass filters. In fact, since a single adder can be used to perform all the additions for the entire circuit by using time division multiplexing, in that case only pairs of modulation phase increment oscillators and highpass filters need be added to add another level of cascading.

These flow diagrams show examples of how the present invention can be applied in a variety of FM topologies. As will be evident to those skilled in the art, they can be combined in further arbitrary and complex ways to produce a variety of modulated waveforms.

Figure 10 shows a functional block diagram of a multichannel real time hardware implementation of the present invention which supports many oscillators, selectively interconnected. In this embodiment, the circuitry operates using time division multiplexing at both the oscillator and state level.

Using multiplexing at the oscillator level means that rather than implementing each of the oscillators as circuitry, the oscillators are each implemented in "virtual circuitry" by accessing their parameters, when needed from a parameter memory 138. As is evident to those skilled in the art, the oscillators could easily be implemented in other ways, such as by "hardwiring" each oscillator so that each has its own dedicated circuitry. The tradeoff between different implementations is usually between simplicity of the circuit and speed and efficiency of the circuit's operation.

Because this implementation does not limit the number of oscillators used, several levels of modulation can be cascaded. However, the simplest case is when only two oscillators are used, one a modulation phase increment oscillator and one a carrier phase increment oscillator. In describing the operation of this implementation, that simple case of only two oscillators will be used, which is shown as a signal flow diagram in Figure 7. In addition, the description below also assumes that ω_c and ω_m are both constants and that the two oscillators are both configured to produce simple sinusoidal waveforms. However, it should be noted that the present invention also allows other waveforms to be used. The choice of waveform is limited, if at all, only by the configuration of the waveshaping circuit 158 that is implemented.

The operation of Figure 10 will now be described using the simple example set forth above. First, a clock generator 136 provides the basic clock for the circuit. The clock generator 136 produces a clock signal at KS times the sample rate, where K is the number of oscillators to be implemented (in this example K=2) and S is the number of clocks or states per oscillator. As is evident to those skilled in the art, K and S are limited for a given sample rate by the speed of the logic and memories.

The clock generator 136 drives a state counter 134, which consists of two sections. The least significant portion of the state counter 134 divides the processing period of each oscillator into S states, one per clock pulse. The most

significant portion of the counter provides a count of the virtual oscillator being processed. In the simple example being described here, only a single bit is needed to count the two oscillators. An overflow of the counter indicates the completion of a sample period.

The circuit of Figure 10 contains three memories. First, a parameter memory 138 contains the controlling parameters for both of the oscillators. These parameters can include such data as the phase increments (both for the modulator and the carrier), amplitudes, envelope time constants and key-on signals, vibrato and tremolo amounts, waveform type and interconnection information for the oscillators. A controlling computer or microprocessor can write this data to be responsive to the required musical notes, which can be generated by a keyboard in real time, by a computer sequencer or by other means. Each of these variables must be set by an application program in ways that are well-known to those skilled in the art.

Parameter memory address control logic 140 controls access to the parameter memory 138. Within the processing period for either oscillator, the various parameters will be accessed during different states and stored in the appropriate parameter latches 142 in time to be used by the computational circuitry for calculation of the current sample point for the current oscillator. The controlling computer or microprocessor writes new data into the parameter memory 138 via the data input 130 during states in which no fetch of parameter data is required, as determined by decoding the state S. During those states, parameter memory address logic 140 also sets the write address of the parameter memory 138 to the address of the parameter supplied by the controlling computer via address input 132.

There are two other memories in addition to the parameter memory 138. A phase memory 154 stores the phase, taken modulo 2π, for both oscillators. A delay memory 166 stores two samples of the output of each of the two oscillators. The contents of both of these memories are used during the processing of the respective oscillators.

In Figure 10, the parameter, phase and delay memories are shown as being separate. As will be evident to those skilled in the art, however, in certain

implementations it may be advantageous to combine any or all of the parameter, phase and/or delay memories into one or more memory units.

The step-by-step operation of the circuit of Figure 10 will now be discussed by describing the processing for a particular sample point by both the modulation (Oscillator 0) and carrier (Oscillator 1) oscillators. First, the processing for the new sample point by Oscillator 0, which is the modulation phase increment oscillator 54 of Figure 7, will be described.

It should be noted that during the processing for each oscillator, the address logic 140 for the parameter memory 138 causes the eight parameters for the modulation oscillator to be accessed, based on a signal received from the state counter 134, and they are read into the appropriate parameter latches 142. This need not happen at one time, as long as each parameter latch is set at some point prior to when it is used. The contents of the various parameter latches 142a to 142h, and how they are used in the operation of the circuit, will be described below.

To generate Oscillator 0's phase for the new sample point, the cumulative phase for the previous sample point must be added to the phase increment. In the simple example being described here, this sum becomes the term $\omega_m t$. To compute this sum, the address logic 168 for the delay memory 166 causes the previous phase output for Oscillator 0 to be output, and it passes through a shifter 174, a multiplexer 146 and an adder/subtractor 150, none of which modify it, to a phase accumulator 152, in which it is stored.

Once this is completed, based on control parameters for Oscillator 0 contained in the eighth parameter latch 142h, the control logic 148 causes the multiplexer 146 to access the phase increment for Oscillator 0, which in the simple example being described here is ω_m, from the fourth parameter latch 142d. The multiplexer 146 then passes the phase increment (ω_m) on as one input to the adder/subtractor 150, while at the same time the control logic 148 and a bank of AND gates 144 cause the previous phase that is stored in the phase accumulator 152 to be passed as the other input to the adder/subtractor 150, which adds the two quantities.

Note that this addition can be performed modulo 2π, which scales the size of the phase accumulator word to 2π. The result of this addition is the new phase

for Oscillator 0, or $\omega_m t$ for the new sample point, which is then stored back into the phase memory 154.

Next, $\sin(\omega_m t)$ must be calculated. This is done by a waveshaping circuit 158, which obtains the phase for Oscillator 0 from the phase memory 154 and transforms it into the desired waveform according to the waveshape parameters contained in the third parameter latch 142c. In the simple example being described here, the output of the waveshaping circuit 158 is $\sin(\omega_m t)$ for the new sample point.

Although the phase incrementing operation described above and the waveshaping operation are described as occurring in sequence, it may be a more optimal design of the circuit to have the two operations occur simultaneously, so that two sample points are being processed by the circuit at one time. By carrying out these two operations in parallel, rather than sequentially, the speed of the circuit is optimized. However, this will be at the expense of the more complex control scheme required to enable parallel operation.

An envelope generator 160 then obtains information about the key state (on or off) and the attack, decay, sustain and release parameters from the first parameter latch 142a, and computes the current envelope value for Oscillator 0. An amplitude logic block 162 takes this current envelope value, obtains amplitude parameters stored in the second parameter latch 142b, and computes from these values an amplitude for Oscillator 0 for the new sample point. An amplitude multiplier 164 then multiplies the output of the waveshaping circuit 158 by the amplitude. The output of the amplitude multiplier 164, which is $I(t)\sin(\omega_m t)$ in the simple example being described here, is then stored in the delay memory 166, replacing the twice previous output sample.

The remainder of the circuit enables the summing of the output of an oscillator with that of the previous oscillators for the same sample. This is done by output control logic 172, a bank of AND gates 176, and an adder 178. In addition, an accumulator 180 maintains the cumulative total of the outputs of all the oscillators. In the simple example being described here, these operations are not necessary, and this circuitry is not used. As will be evident to those skilled in the art, however, this circuitry can be used when several notes in a chord are being generated at the same

time, with each note having its own set of modulation and carrier oscillators. In these cases, the results of several oscillators must be summed following the separate processing of the various oscillators or oscillator groups.

This completes the processing for the new sample point for Oscillator 0, and a similar process then begins for the carrier oscillator, Oscillator 1. However, because the output of Oscillator 0 must be highpass filtered and used by Oscillator 1, the processing for the new sample point for Oscillator 1 is somewhat more complex. Note that the parameters for Oscillator 1 must be read from the parameter memory 138 into the parameter latches 142, replacing the parameters for Oscillator 0, at some point prior to the time that Oscillator 1's parameters are accessed during the processing for Oscillator 1.

The processing for the new sample point by Oscillator 1 begins by highpass filtering the output of any modulation oscillator for Oscillator 1. The information latched in the seventh parameter latch 142g, determines which oscillator, if any, serves as the modulator for the carrier oscillator. In the simple example being described here, the seventh parameter latch 142g indicates that Oscillator 0 is a modulation oscillator for Oscillator 1.

As can be seen from the signal flow diagram of a highpass filter in Figure 6, the highpass filtering of Oscillator 0's output requires the two most recent sample outputs of Oscillator 0. These two outputs are stored in the delay memory 166. When required during the processing described below, these two values are fetched from the delay memory 166 and input to the multiplexer 146.

If self-modulation is being done, a shifter 174 is provided to attenuate these delayed signals, if desired. Because the simple example being described herein does not include self-modulation, the shifter 174 is not used in this example. However, it will be evident to those skilled in the art how this shifter 174, which uses a feedback constant contained in the sixth parameter latch 142f, can be used to perform the self-modulation for which the signal flow diagram is shown in Figure 8.

To generate the highpass filtered output of Oscillator 0 for the new sample point, the difference between the two previous sample outputs of the modulation oscillator, Oscillator 0, must be calculated. In the simple example being

described here, calculating this difference results in, approximately, multiplying by ω_m. Accordingly, for the new sample point, the result is, approximately, $I(t)\omega_m\sin(\omega_m t)$.

To compute this difference, the address logic 168 for the delay memory 166 causes the previous phase output for Oscillator 0 to be output, and it passes through the shifter 174, multiplexer 146 and adder/subtractor 150, none of which modify it, to the phase accumulator 152, in which it is stored. Once this is completed, the control logic 148 causes the multiplexer 146 to access the output of Oscillator 0 for the new sample point, which is $I(t)\sin(\omega_m t)$, from the delay memory 166. The multiplexer 146 passes this next sample output as one input to the adder/subtractor 150, while at the same time the control logic 148 and a bank of AND gates 144 cause the previous sample output that is now being stored in the phase accumulator 152 to be passed as the other input to the adder/subtractor 150, which subtracts Oscillator 0's previous sample output from Oscillator 0's output for the new sample point. In the simple example being described here, the result is, approximately, $I(t)\omega_m\sin(\omega_m t)$.

Next, to generate Oscillator 1's phase for the new sample point, the cumulative phase for the previous sample point for Oscillator 1 must be added to the sum of the phase increment for Oscillator 1, which is ω_c, and the output of the highpass filtered output of Oscillator 1's modulation oscillator, Oscillator 0, which output is $I(t)\omega_m\sin(\omega_m t)$. For the new sample point, this sum becomes, approximately, $[\omega_c+I(t)\omega_m\sin(\omega_m t)]t$.

To compute this sum, the address logic 168 for the delay memory 166 causes it to output the previous phase output for Oscillator 1, which then passes through the shifter 174 and multiplexer 146, neither of which modify it, to the adder/subtractor 150, which adds it to the contents of the phase accumulator 152. In the simple example being described here, those contents are the highpass filtered output of Oscillator 0, or $I(t)\omega_m\sin(\omega_m t)$. This is done by having the multiplexer 146 pass the previous sample output for Oscillator 1 as one input to the adder/subtractor 150, while at the same time the control logic 148 and the bank of AND gates 144 cause the previous sum that is stored in the phase accumulator 152 to

be passed as the other input to the adder/subtractor 150, which adds the two quantities. As before, this new sum is stored in the phase accumulator 152.

Finally, the multiplexer 146 obtains the phase increment for Oscillator 1, which is ω_c, from the fourth parameter latch 142d. The multiplexer 146 passes the phase increment (ω_c) on as one input to the adder/subtractor 150, while at the same time the control logic 148 and the bank of AND gates 144 cause the previous sum that is stored in the phase accumulator 152 to be passed as the other input to the adder/subtractor 150, which adds the two quantities.

Note that any or all of the additions and subtractions discussed above can be performed modulo 2π, which scales the size of the phase accumulator word to 2π. The result of all these additions and subtractions is the new phase for Oscillator 1, or $[\omega_c+I(t)\omega_m\sin(\omega_m t)]t$ for the new sample point, which is then stored back into the phase memory 154.

Next, $\sin([\omega_c+I(t)\omega_m\sin(\omega_m t)]t)$ must be calculated for the new sample point. This is done by the waveshaping circuit 158, which obtains the phase for Oscillator 1 from the phase memory 154 and transforms it into the desired waveform according to the waveshape parameters contained in the third parameter latch 142c. In the simple example being described here, the output of the waveshaping circuit 158 is, approximately, $\sin([\omega_c+I(t)\omega_m\sin(\omega_m t)]t)$ for the new sample point. As noted above, although the phase incrementing operation described above and the waveshaping operation are described as occurring in sequence, in a more optimal design the circuit can have the two operations occur simultaneously.

The envelope generator 160 then obtains information about the key state (on or off) and the attack, decay, sustain and release parameters from the first parameter latch 142a, and computes the current envelope value for Oscillator 1. The amplitude logic block 162 then takes this current envelope value, obtains amplitude parameters stored in the second parameter latch 142b, and computes from these values an amplitude for Oscillator 1 for the new sample point. An amplitude multiplier 164 then multiplies the output of the waveshaping circuit 158 by the amplitude. The output of the amplitude multiplier 164, which is approximately

$A(t)\sin([\omega_c+I(t)\omega_m\sin(\omega_m t)]t)$ in the simple example being described here, is then stored in the delay memory 166, replacing the twice previous output sample for Oscillator 1.

As discussed above, the remainder of the circuit enables the summing of the output of an oscillator with that of the previous oscillators for the same sample. In the simple example being described here, these operations are not necessary, and this circuitry is not used.

After both oscillators have been processed, based on the contents of the fifth parameter latch 142e, the output control logic 172 causes the contents of the accumulator 180 to be passed to the audio output latch 182, and the accumulator 180 to be cleared using the AND gates 176. Processing then begins for a new sample beginning with Oscillator 0. In this way, a signal representing a synthesized musical sound is generated at the audio output latch 182, where it can be amplified, as appropriate, and used to power a speaker where it can be heard. This FM synthesis process occurs in real time.

As noted above, according to the teachings of this specification, the circuit of Figure 10 can be used to support a variety of FM topologies, such as self-modulation, cascading and simultaneous generation of several musical notes, in ways that are well known to those skilled in the art. As also noted above, tonal qualities, such as tremolo and vibrato, can also be implemented in the present invention by varying ω_c, ω_m and A(t) by combining techniques well known to those skilled in the art with the teachings of this specification.

Finally, it should be noted that the parameters can be stored in a variety of formats in the parameter memory 138, which allows for compatibility with existing synthesizers. In such cases, additional logic may be needed to interpret the data in the parameter latches 142 into the correct format.

What is claimed is:

1. A circuit to synthesize sounds comprising a filter interposed between a modulation phase increment oscillator and a carrier phase increment oscillator.

2. The circuit of claim 1 wherein the filter is a highpass filter.

3. The circuit of claim 2 wherein the highpass filter is a first order FIR highpass filter.

4. The circuit of claim 1 wherein the modulation phase increment oscillator comprises waveshaping means.

5. The circuit of claim 4 wherein the waveshaping means produces sinusoidal waveforms.

6. The circuit of claim 4 wherein the waveshaping means produces non-sinusoidal waveforms.

7. The circuit of claim 1 wherein the carrier phase increment oscillator comprises waveshaping means.

8. The circuit of claim 7 wherein the waveshaping means produces sinusoidal waveforms.

9. The circuit of claim 7 wherein the waveshaping means produces non-sinusoidal waveforms.

10. The circuit of claim 1 wherein the modulation phase increment oscillator comprises a multiplier.

11. The circuit of claim 1 wherein the carrier phase increment oscillator comprises a multiplier.

12. The circuit of claim 1 wherein the circuit is time division multiplexed to provide K oscillators.

13. The circuit of claim 1 wherein the circuit is time division multiplexed to provide S states.

14. The circuit of claim 1 further comprising an adder interposed between the filter and the carrier phase increment oscillator.

15. The circuit of claim 14 wherein at least two of the modulation phase increment oscillator, the carrier phase increment oscillator, the filter and the adder utilize the same adder circuitry.

16. A circuit to synthesize sounds comprising a filter for operating on a signal which is then applied to a frequency input of a carrier phase increment oscillator.

17. A circuit to synthesize sounds comprising:

a modulation phase increment oscillator for producing a modulation signal from a modulation phase increment and a modulation index;

a filter for producing a filtered version of the modulation signal;

an adder for producing a sum of the filtered version of the modulation signal and a carrier phase increment; and

a carrier phase increment oscillator for producing an audio signal from the sum.

18. The circuit of claim 17 wherein the carrier phase increment oscillator comprises circuitry for producing the audio signal from the sum and an amplitude envelope.

19. The circuit of claim 17 wherein the modulation phase increment and/or the carrier phase increment is time varying.

20. The circuit of claim 18 wherein the modulation index and/or the amplitude envelope is time varying.

21. A method of synthesizing sounds comprising the steps of:

producing a modulation signal from a modulation phase increment and a modulation index;

filtering the modulation signal;

adding the filtered modulation signal and a carrier phase increment to obtain a sum; and

producing an audio signal from the sum.

22. The method of claim 21 wherein the producing step comprises producing an audio signal from the sum and an amplitude envelope.

23. The method of claim 21 wherein the filtering step comprises using a highpass filter.

24. The method of claim 23 wherein the filtering step comprises using a first order FIR highpass filter.

25. The method of claim 21 wherein the step of producing a modulation signal comprises waveshaping.

26. The method of claim 25 wherein the waveshaping step comprises producing sinusoidal waveforms.

27. The method of claim 25 wherein the waveshaping step comprises producing non-sinusoidal waveforms.

28. The method of claim 21 wherein the step of producing a signal comprises waveshaping.

29. The method of claim 28 wherein the waveshaping step comprises producing sinusoidal waveforms.

30. The method of claim 28 wherein the waveshaping step comprises producing non-sinusoidal waveforms.

31. The method of claim 21 wherein the step of producing a modulation signal comprises multiplying.

32. The method of claim 21 wherein the step of producing a signal comprises multiplying.

33. The method of claim 21 wherein at least one of the steps is time division multiplexed.

34. The method of claim 21 wherein at least two of the steps utilize the same adder circuitry.

35. The method of claim 21 wherein the modulation phase increment and/or the carrier phase increment is time varying.

36. The method of claim 22 wherein the modulation index and/or the amplitude envelope is time varying.

37. A circuit to synthesize sounds by self-modulation comprising:

a phase increment oscillator for producing a self-modulation signal from a first phase increment, where the self-modulation signal is also used as a signal representing a sound;

a delay operator which delays the self-modulation signal;

a filter for producing a filtered version of the self-modulation signal; and

an adder for producing a sum of the filtered version of the self-modulation signal and a second phase increment, where the sum is the first phase increment.

38. The circuit of claim 37 further comprising a shifter, disposed between the delay operator and the filter, which multiplies the delayed self-modulation signal by a fixed amount.

39. The circuit of claim 38 wherein the modulation phase increment oscillator comprises circuitry for producing the self-modulation signal from the first phase increment and an amplitude envelope.

40. The circuit of claim 39 wherein the amplitude envelope is time varying.

41. A circuit to synthesize sounds by cascading comprising:

at least two modulation phase increment oscillators comprising circuitry for producing a modulation signal from a modulation phase increment and a modulation index, wherein at least one of the at least two oscillators receives the filtered output of a previous of the at least two oscillators;

at least one filter comprising circuitry for producing a filtered version of at least one of the modulation signals produced by at least one of the modulation phase increment oscillators;

at least one adder comprising circuitry for producing a sum of the filtered version of at least one of the modulation signals and a second modulation phase increment; and

a carrier phase increment oscillator for producing a signal representing a sound from the sum of the filtered and added outputs of the at least two modulation phase increment oscillators.

42. The circuit of claim 41 wherein the carrier phase increment oscillator comprises circuitry for producing the signal representing a sound from the final sum produced by the at least one adder and an amplitude envelope.

43. The circuit of claim 42 wherein the modulation index and/or the amplitude envelope is time varying.

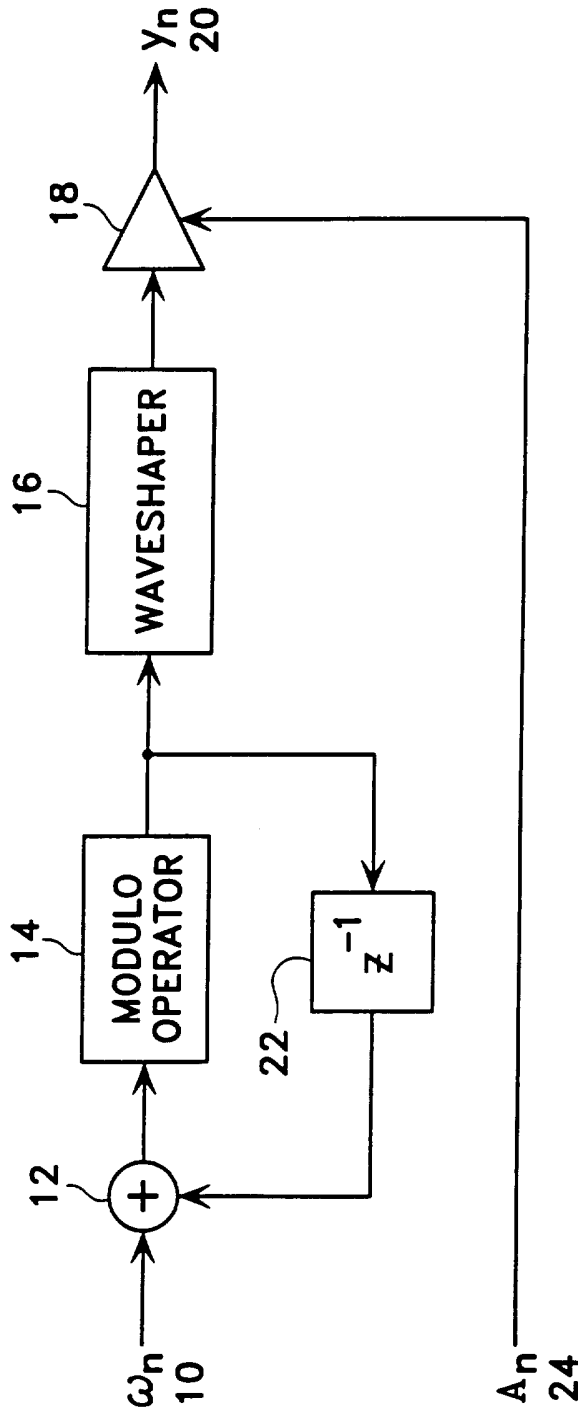

y_n
20
18
16
WAVESHAPER
14
MODULO OPERATOR
22
z^{-1}
12
ω_n
10
A_n
24

FIG. 1

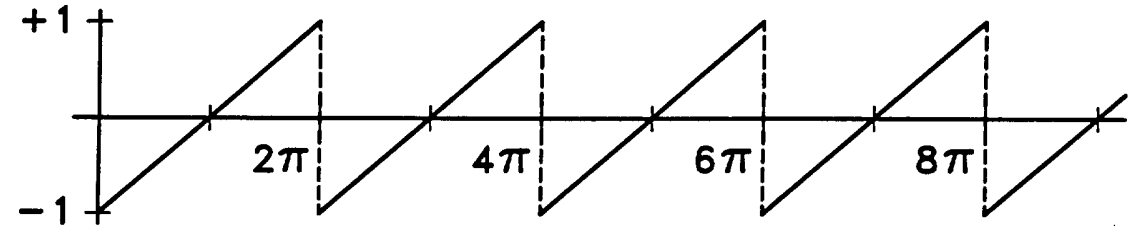


FIG. 2a

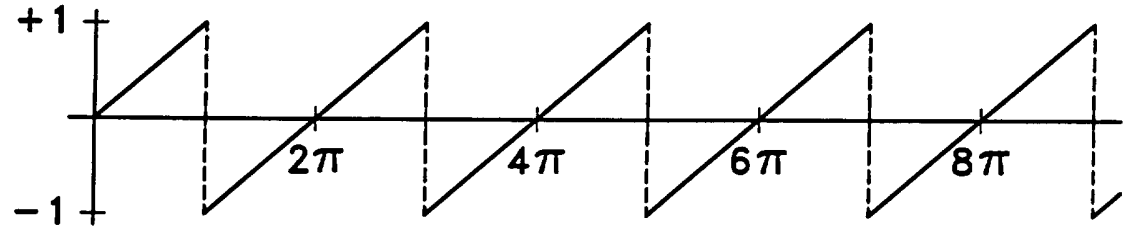


FIG. 2b

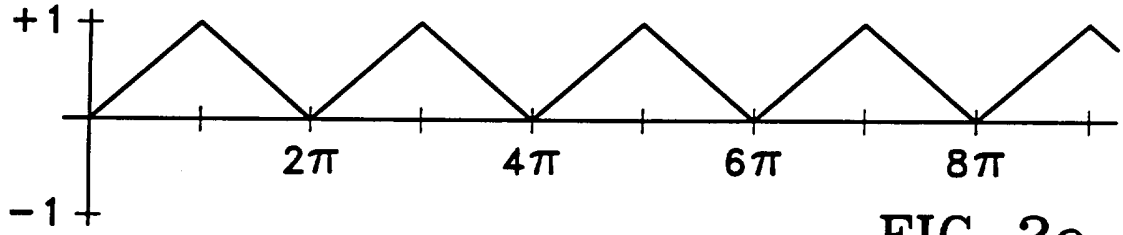


FIG. 2c

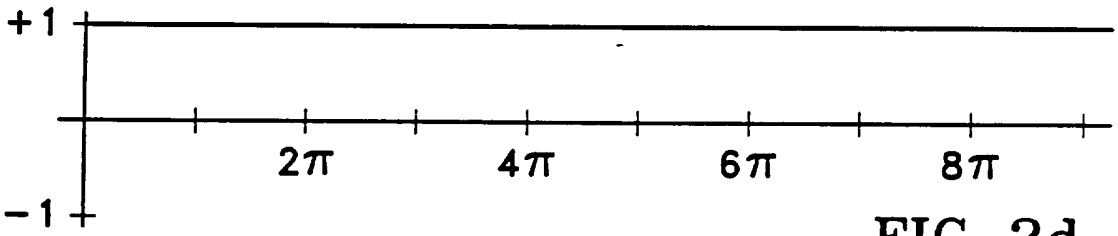


FIG. 2d

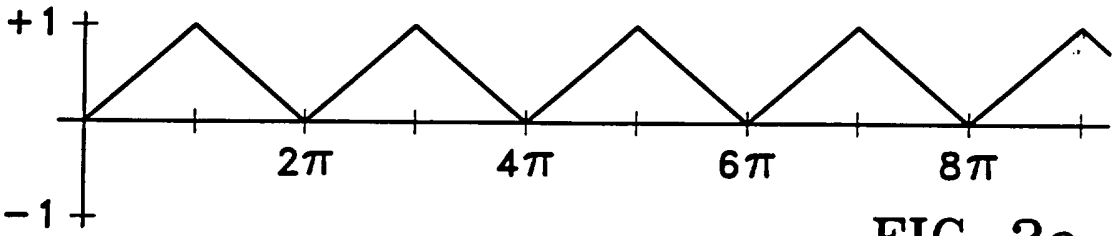


FIG. 2e

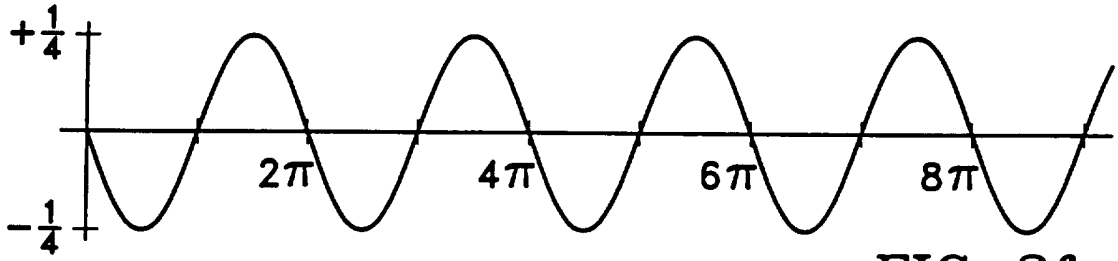


FIG. 2f

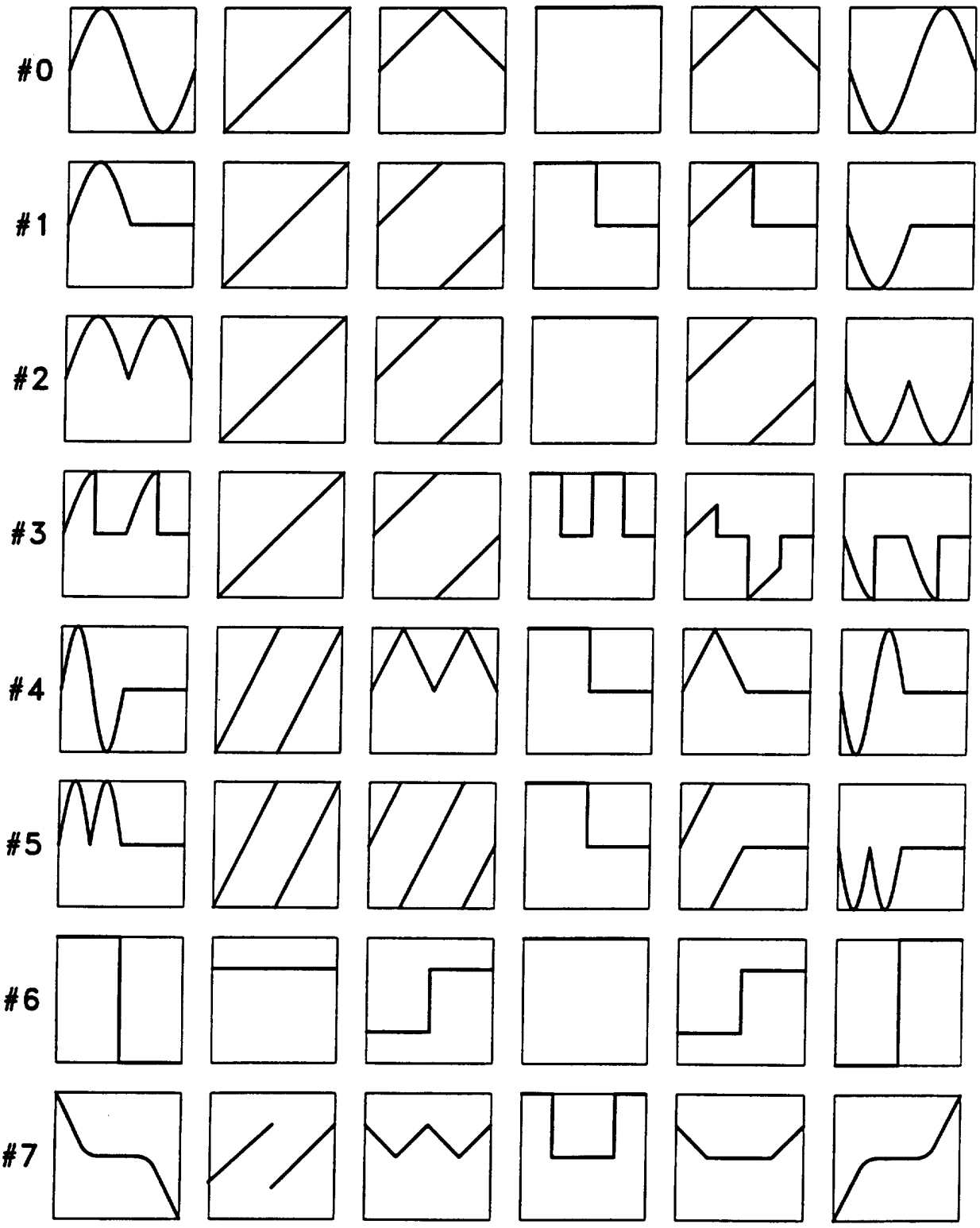


FIG. 3a FIG. 3b FIG. 3c FIG. 3d FIG. 3e FIG. 3f

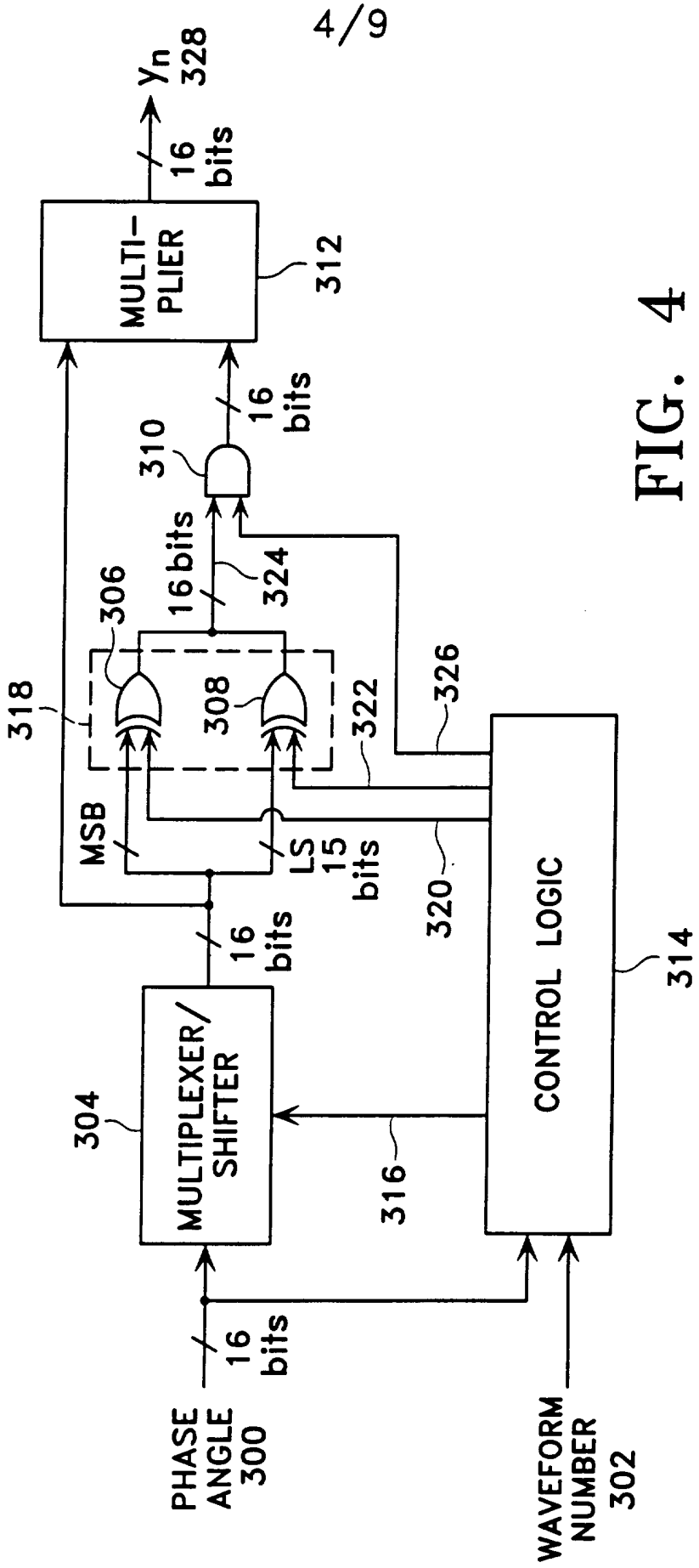

Yn 328
16 bits
MULTI-PLIER
312
16 bits
310
16 bits
324
306
308
318
322
326
MSB
LS 15 bits
320
16 bits
MULTIPLEXER/ SHIFTER
304
316
CONTROL LOGIC
314
PHASE ANGLE 300
16 bits
WAVEFORM NUMBER 302

FIG. 4

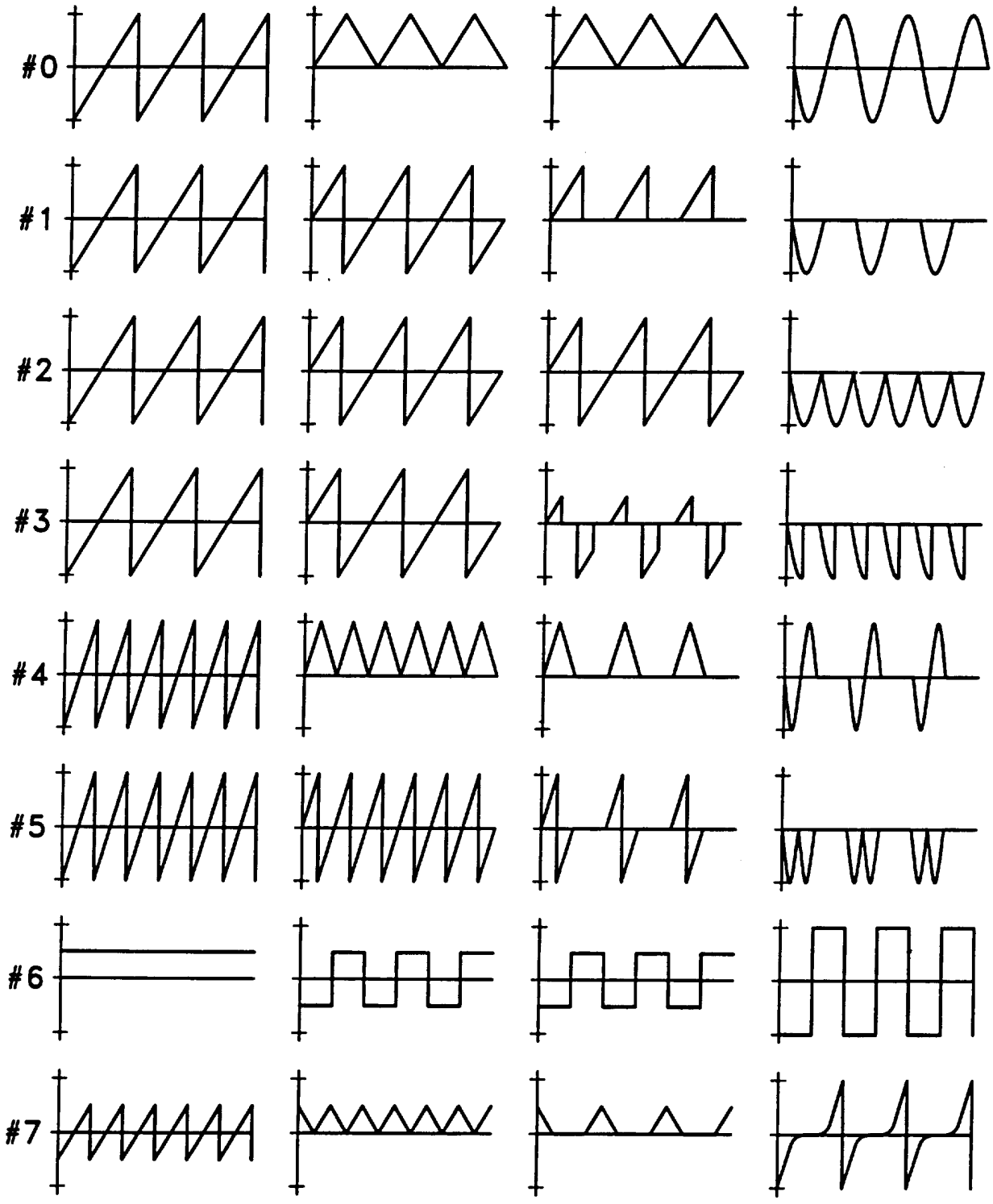


FIG. 5a FIG. 5b FIG. 5c FIG. 5d

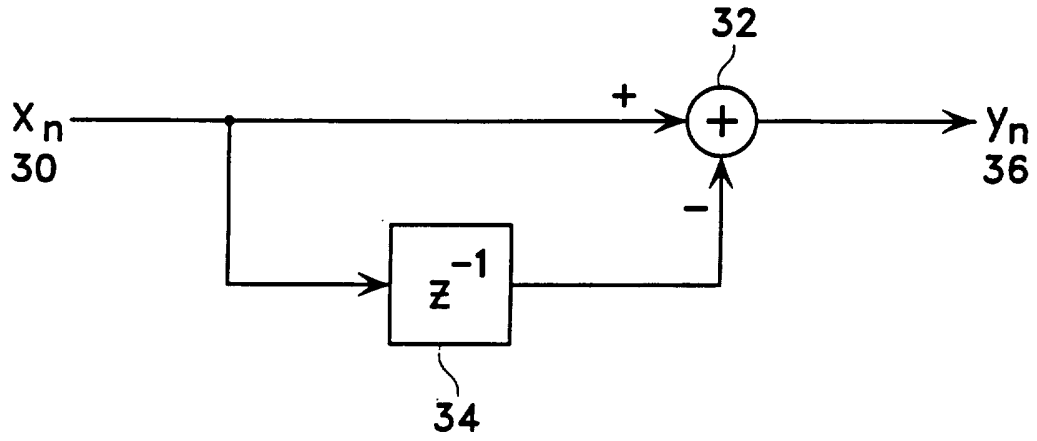

32
+
x_n
30
y_n
36
−
z^{-1}
34

FIG. 6

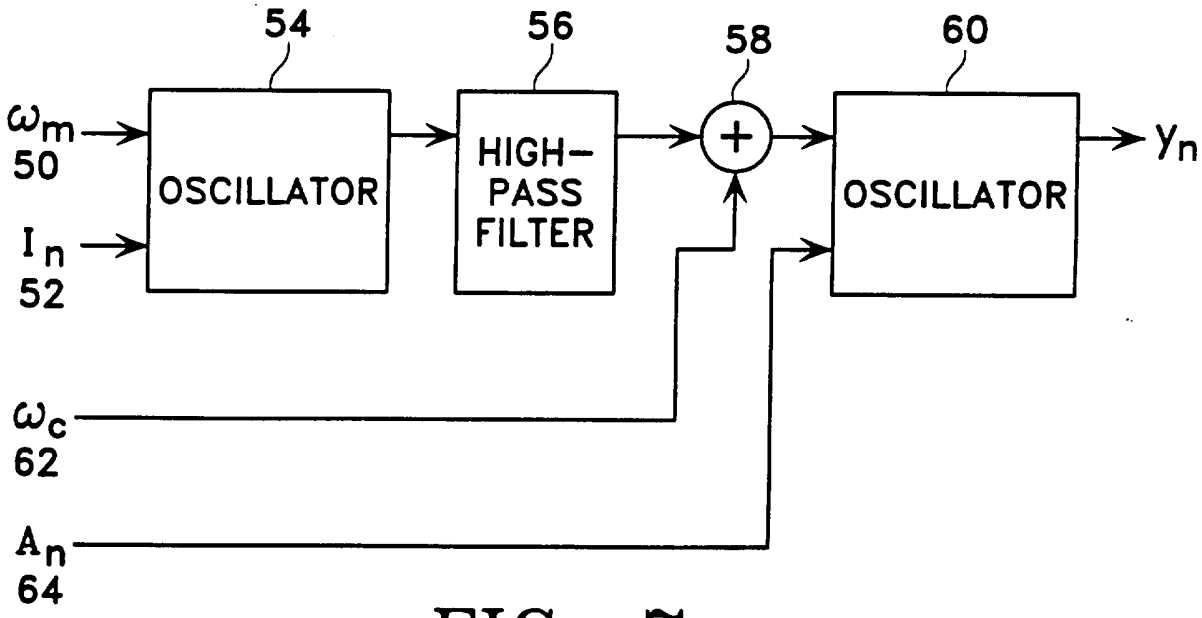

54
56
58
60
ω_m
50
I_n
52
OSCILLATOR
HIGH-PASS FILTER
OSCILLATOR
y_n
ω_c
62
A_n
64

FIG. 7

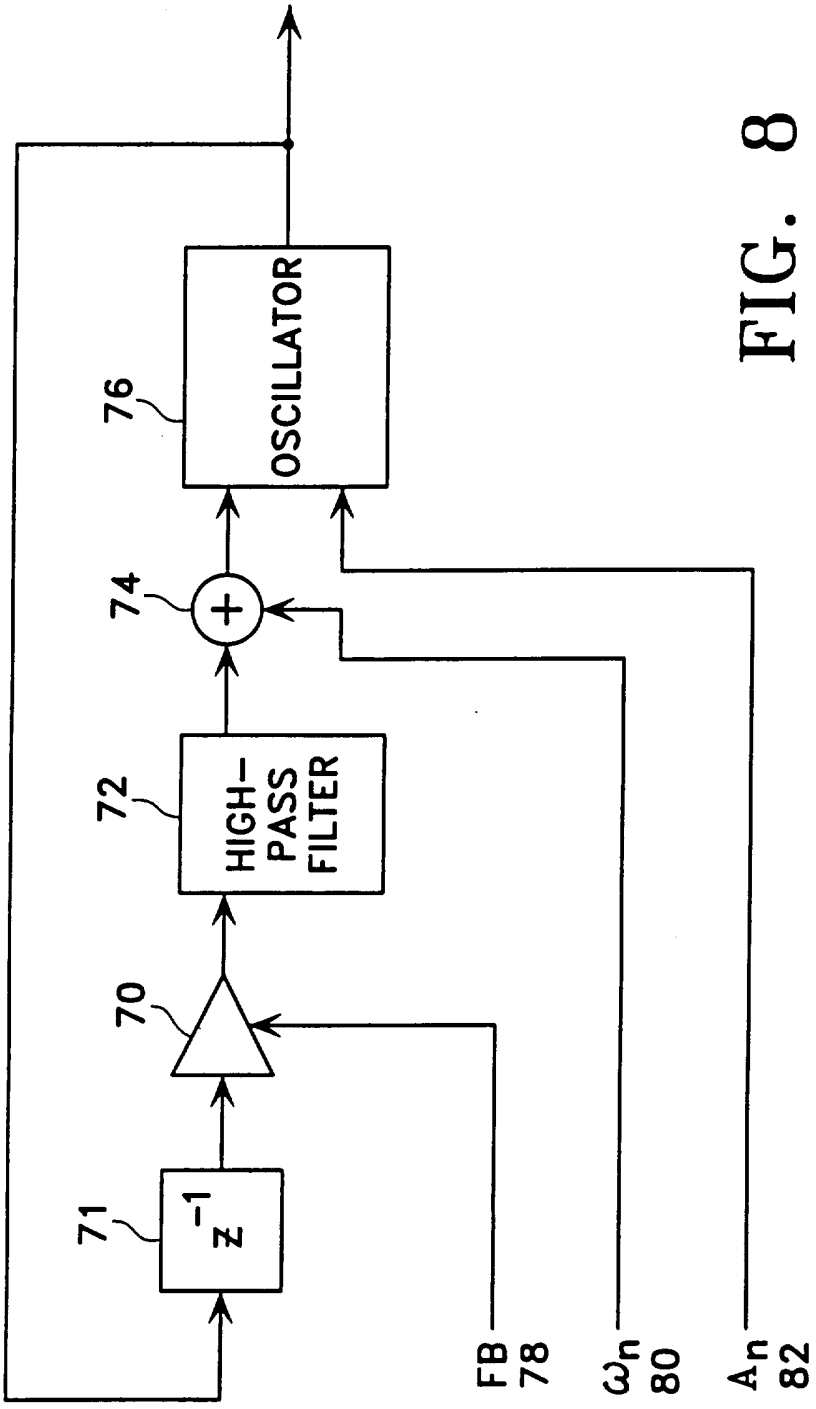

OSCILLATOR
76
74
HIGH-
PASS
FILTER
72
70
71
z^{-1}
FB
78
ω_n
80
A_n
82

FIG. 8

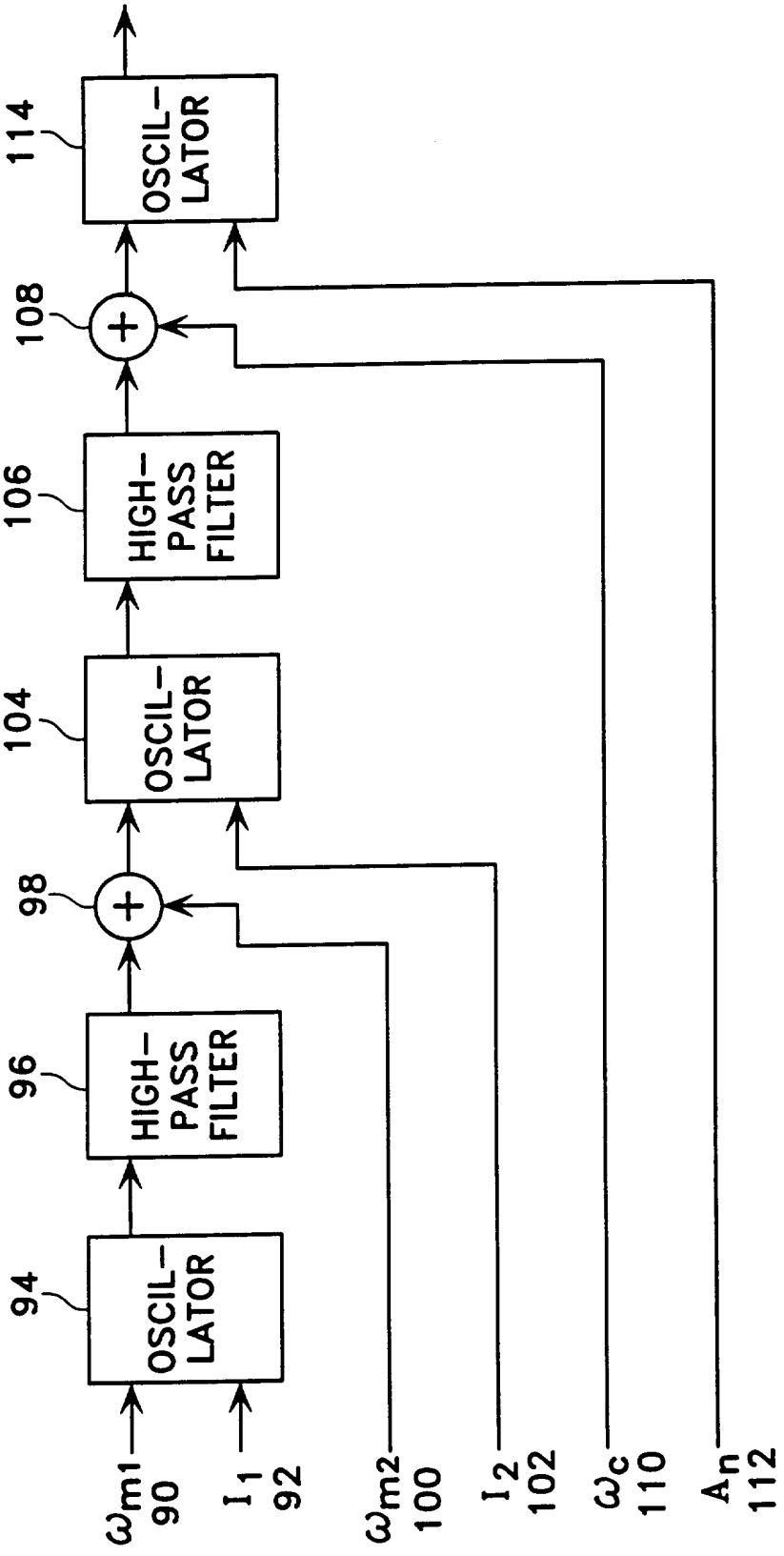

OSCIL-LATOR
114
108
HIGH-PASS FILTER
106
OSCIL-LATOR
104
98
HIGH-PASS FILTER
96
OSCIL-LATOR
94
ω_{m1} 90
I_1 92
ω_{m2} 100
I_2 102
ω_c 110
A_n 112

FIG. 9

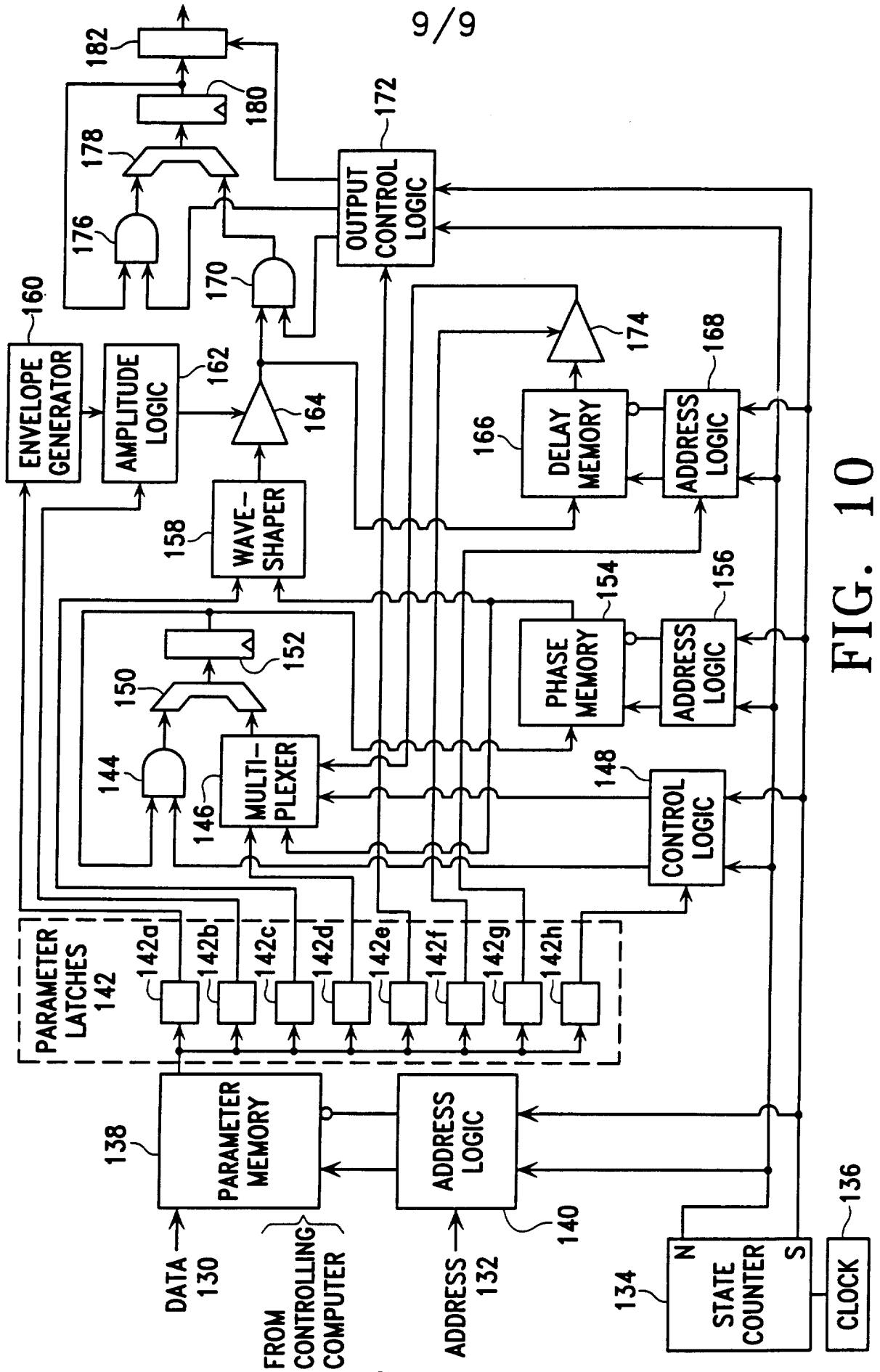

182
180
178
176
172
OUTPUT CONTROL LOGIC
160
ENVELOPE GENERATOR
AMPLITUDE LOGIC
162
164
170
174
168
166
DELAY MEMORY
ADDRESS LOGIC
158
WAVE-SHAPER
154
156
PHASE MEMORY
ADDRESS LOGIC
152
150
144
MULTI-PLEXER
146
148
CONTROL LOGIC
PARAMETER LATCHES 142
142a
142b
142c
142d
142e
142f
142g
142h
138
PARAMETER MEMORY
ADDRESS LOGIC
140
DATA 130
FROM CONTROLLING COMPUTER
ADDRESS 132
134
N
STATE COUNTER
S
CLOCK
136

FIG. 10

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IB 96/00460

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G10H1/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G10H

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP,A,0 454 047 (CASIO COMPUTER CO LTD) 30 October 1991 see page 12, line 19 - page 15, line 10; figures 13-16	1,4-6, 11,17-22
A	US,A,5 157 215 (NAKAE TETSUICHI ET AL) 20 October 1992 see column 11, line 20 - column 13, line 58; figures 13-17	1-3, 17-24
A	WO,A,95 06859 (MEDIA VISION INC) 9 March 1995 see page 6, line 1 - page 8, line 13; figures 1-5	37-40

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
28 August 1996	06.09.96

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Pulluard, R

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/IB 96/00460

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US,A,4 554 858 (WACHI MASATADA ET AL) 26 November 1985 see column 14, line 22 - column 15, line 10 see column 17, line 44 - column 18, line 50; figures 7,10	2,3,23, 24
A	GB,A,2 103 005 (NIPPON MUSICAL INSTRUMENTS MFG) 9 February 1983 see page 1, line 39 - page 3, line 51; figures 1,4A,4B,5D	2,3,23, 24

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte ·onal Application No
PCT/IB 96/00460

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
EP-A-0454047	30-10-91	JP-A-	4005692	09-01-92
		JP-A-	4060693	26-02-92
		JP-A-	4060694	26-02-92
		KR-B-	9405988	30-06-94
		US-A-	5340938	23-08-94
US-A-5157215	20-10-92	JP-A-	4097295	30-03-92
		JP-A-	3107200	07-05-91
WO-A-9506859	09-03-95	US-A-	5543578	06-08-96
		AU-B-	8009994	22-03-95
		EP-A-	0717871	26-06-96
US-A-4554858	26-11-85	JP-C-	1576727	24-08-90
		JP-B-	2001314	11-01-90
		JP-A-	59030598	18-02-84
		JP-C-	1668696	29-05-92
		JP-B-	3028720	19-04-91
		JP-A-	59049595	22-03-84
GB-A-2103005	09-02-83	JP-B-	1015075	15-03-89
		JP-C-	1549196	09-03-90
		JP-A-	58014191	26-01-83
		DE-A-	3226619	24-03-83