An Inexpensive Digital Sound Synthesizer

H. G. Alles

Bell Laboratories
Murray Hill, New Jersey 07974

Introduction

A time-multiplexed microprocessor-controlled all-digital oscillator has been developed which provides 32 independent audio oscillator functions. The circuit includes envelope generators to control the amplitude and frequency of each oscillator. A timer function is provided for each oscillator as well. A frequency modulation input for each oscillator is available. The oscillators may be arbitrarily interconnected and their outputs combined. The circuit when controlled by a microprocessor may be used to digitally synthesize a variety of sounds, but its most powerful application is for real time music synthesis. Approximately 110 low cost digital integrated circuits are used. This circuit combined with a microprocessor is an attractive method for implementing the next generation of commercial music synthesizers and "top of the line" home organs. It may also find use in super game terminals of the future for providing sound effects and background mood music.

Overview

The oscillators are implemented using all digital circuitry. The oscillator output is a series of samples or numbers (32,000 per second) which represents the synthesized sound. All of the synthesis, amplitude control, and sound mixing is done while the sounds are represented digitally by these series of samples. The final output is achieved by converting one series of digital samples to a voltage using a digital to analog (D/A) converter.

Actually, only one high speed oscillator circuit is used, but it is time multiplexed 32 ways to generate multiple oscillator functions. Each oscillator function uses a separate set of eight, 16 bit registers (which are addressed as 16, 8 bit bytes) or control words (bytes) to control and accomplish the calculation. These are implemented as memory locations in a 256 word by 16 bit random access memory (RAM), which supports the 32 independent oscillator functions. This RAM is interfaced to the bus of the controlling microprocessor so that it appears in the microprocessor address space. The RAM is read or written by microprocessor program commands to control all aspects of the oscillator function.

The oscillator circuit performs all the calculations necessary to generate one sample of an oscillator output in about 1 μ sec. A final output sample is required only every 32 μ sec. Each 1 μ sec calculation period is called a "time slot", and these are numbered 0, 1, ..., 31. The output of oscillator number 0 is calculated during time slot 0, the output oscillator 1 is calculated during time slot 1, etc., until the output of oscillator 31 is calculated during time slot 31.

Each oscillator has two inputs which may receive data from the outputs of other oscillators. One of these is a frequency modulation (FM) input used to produce vibrato or "voicing" effects. The second is an "adder" input which is used to combine the output of some other oscillator with the output of the current oscillator. Four temporary data storage registers (DSR) are provided to receive this combined output as well as supply the data for the two oscillator inputs. Three separate bit fields in one of the oscillator control bytes specifies which register is used for each purpose. The oscillator inputs may also be inhibited so that there is no FM or adder input.

The oscillators may be interconnected using the DSRs, as specified by the control bytes. One oscillator output may be used to frequency modulate another oscillator, and the output of this oscillator added to the output of another, etc. The calculation proceeds sequentially (i.e. oscillator 0,1,...,31) with the n^{th} oscillator's output available for any input to the $n+1^{\text{th}}$ or higher numbered oscillator. The output of oscillator 31 is the final output sample, and is put out to the D/A for conversion. This analog output is then connected to an amplifier-speaker system to hear the synthesized sounds.

General Oscillator Function

Figure 1 is a conceptual block diagram of the oscillator. The frequency is determined by three independent and equally weighted terms:

- 1) frequency
- 2) Δ frequency ramp function
- 3) frequency modulation input.

Each of these are 16 bit, 2's complement numbers which are added to the value of the phase from the previous sample to generate the phase of the current sample. The phase is stored as a 16 bit quantity in one of the eight RAM memory locations provided for each oscillator. The value of the phase ϕ is interpreted as

$$-\pi \leq \phi < \pi$$

so that positive full scale (2's complement arithmetic) corresponds to $[(2^{15}-1)\pi]/2^{15}$ and minus full scale is equal to $-\pi$. As positive values accumulate in the phase, overflow occurs causing the value to go from positive full scale (π) to negative full scale ($-\pi$). Each overflow corresponds to one cycle of an oscillation.

The 13 most significant bits of the phase are used as the address to a ROM which is programmed with sample values representing the desired output waveshape. The circuit provides for a choice of up to eight different waveshapes. The values from the waveshape ROM are 12 bit quantities. These 12 bit quantities must be scaled in a smooth and continuous fashion to produce acceptable sound synthesis. This is accomplished without the use of a digital multiplier by using phase cancellation and choosing waveshape functions g , that have the property

$$g(-\phi) = -g(\phi).$$

A second access to the waveshape ROM is made using the value of the phase ϕ plus a phase shift $\Delta\phi$ as the address. The waveshape value at ϕ is subtracted from the value at $\phi+\Delta\phi$. If the waveshape function is a sine wave, then the scaled output will be

$$\sin(\Delta\phi+\phi) - \sin\phi = 2\sin(\Delta\phi/2)\cos(\phi+\Delta\phi/2) \quad (1)$$

As $\Delta\phi$ varies from zero to positive full scale ($0 \leq \Delta\phi < \pi$) the oscillator output amplitude varies from zero to a maximum peak value of 13 bits.

The oscillator circuit includes two ramp function generators which are used to provide linearly time dependent values for Δ frequency and $\Delta\phi$ (which in turn controls the amplitude). These generators combined with microprocessor control can produce envelope functions of arbitrary complexity. The microprocessor specifies a current value, a final value, and a rate of change. The oscillator circuit increments the current value by the rate of change until the final value is achieved. A control byte in the oscillator may be set so that a microprocessor program interrupt will be generated at this time. The final value is used until the microprocessor specifies a new final value and rate of change. Each oscillator function also includes a digital timer which may be enabled to generate a program interrupt after a specified amount of time. These interrupt features are extremely useful for generating envelopes made up of several ramp segments.

Details

The previous sections briefly described the oscillator function. We now present enough detail so that the oscillators may be programmed and controlled. Appendix I is an address map of the oscillator control bytes and registers. Appendix II describes the bit fields in each control byte. The oscillator circuit may be interfaced to a DEC LSI-11 microprocessor, to the S100 bus, or to other similar microprocessors. An address space of 512 eight bit bytes plus four I/O addresses are required. In the following discussion when referring to these control bytes and the two byte, 16 bit registers, byte addresses will be

used which are relative to a starting address that is determined by jumper wires in the oscillator circuit. When octal numbers are used, they will have a leading zero. Sixteen consecutive bytes are used by each oscillator. The relative address for a specific control byte may be calculated using

$$16n + k$$

where n is the oscillator number (0,1,...,31) and k is the control byte (0,1,...,15). Bits within a control byte are numbered 0 through 7 with the least significant bit numbered zero.

1. Patch Control Byte ($k=0$).

Four data storage registers (DSR) are available to pass oscillator outputs to the inputs of other oscillators. The patch control byte contains 3 fields to specify which of these registers is used for the FM input, adder input, and output. The output sample is loaded into the DSR specified by bits 6 and 7. The output of oscillator 31 (the last oscillator) automatically goes to the D/A.

If bit 5 is set to zero, the contents of the DSR specified by bits 3 and 4 is added to the oscillator sample to form the oscillator output. However, if bit 5 is set to 1, then the adder input function is inhibited, and the value zero will be added to the oscillator output (bits 3 and 4 may have any value in this case).

If bit 2 is set to zero, the contents of the DSR specified by bits 0 and 1 is used to frequency modulate (FM) the oscillator. If bit 2 is set to 1, then the frequency modulation will be inhibited.

We now illustrate with two examples how the patch control bytes and the DSRs are used to interconnect the oscillators. The simplest interconnection is to add all of the outputs together. We will arbitrarily choose DSR 1 as the accumulation register. At the start of a sample period, the output of oscillator 0 is calculated first. During the 1 μ sec time slot for oscillator 0, the DSR specified as the FM input is accessed, and its current contents used as the FM source. Then the DSR specified as the adder input is accessed, and its current contents added to the oscillator output sample. Finally, this sum is loaded into the DSR specified to receive the output. The next 1 μ sec time slot is used to calculate the output of oscillator 1, and contents of the specified DSRs at the time of the calculation are used as the inputs. Each successive oscillator sample is calculated, using the current contents of the DSRs as inputs. For this example, the FM input is inhibited for all oscillators. The adder input of oscillator 0 is also inhibited, and its output loaded into DSR 1. All of the other oscillators use DSR 1 as their adder inputs and then load their outputs into DSR 1. At the end of time slot 0, DSR 1 contains the output of oscillator 0. At the end of time slot 1, DSR 1 contains the sum of the outputs of oscillator 0 and oscillator 1. At the end of time slot n , DSR 1 contains the sum of the output of oscillators 0 through n . Finally, the output of oscillator 31 is put out to the D/A at the end of the sample period. DSR 1 contains the final output sample also, but this value is never used. At the beginning of the next sample period, DSR 1 is loaded with the simple output of oscillator 0 (since the adder input is inhibited) and the calculation sequence is repeated again for the next sample.

The patch control byte value for oscillator 0 (relative address 00) is 0144. The value for all the other oscillators (relative addresses $16n$) is 0114.

As a second interconnection example, consider dividing the oscillators into pairs, using the output of first member of the pair to frequency modulate the output of the second member. Then add together all the outputs of the modulated oscillators to form the final output to the D/A. We choose DSR 1 to accumulate the output and DSR 2 to pass the data to the FM input. Oscillator 0 (and all even numbered oscillators) generates the modulating signal, using no FM or adder input, and it loads its output into DSR 2. Its patch control byte value is set to 0244. Oscillator 1 uses DSR 2 as its FM input, inhibits its adder input, and loads its output into DSR 1. Its patch control byte value is 0142. Oscillator 2 may also use DSR 2 to pass its FM data on to oscillator 3 since the data from oscillator 0 has already been used, and will not be needed again. Thus, the patch control byte value for all even numbered oscillators is 0244. Oscillator 3 uses DSR 2 as its FM input, DSR 1 as its adder input; and loads its output into DSR 1. Its patch control byte value (and all other odd numbered oscillators) is 0112. The

output of the last oscillator will go to the D/A.

Since each DSR may be used several times during each sample period to pass data between oscillators, almost any interconnection between oscillators may be accomplished. However, care must be taken to assign the oscillators in the proper order. For a group of oscillators interconnected, the lowest numbered oscillator should be assigned to produce the signal needed earliest in the calculation. In the example above, the modulating signal is calculated before the modulated oscillator output is calculated.

2. Frequency Control Word (k=4 and 5)

Figure 1 illustrates that three separate terms contribute equally to determining the oscillator frequency. For the moment, suppose that the FM and Δ frequency inputs are zero. The oscillator frequency is the number of times per second that the 16 bit phase accumulator overflows. (The phase is accessible by the microprocessor at addresses k = 2 and 3.) Each sample period, the frequency control word value is added to the current value of the phase to produce the new phase. (The Δ frequency value and the FM input are also added, but for now they have zero value.) The relation between the frequency control word value FCW and frequency f in Hz is

$$FCW = \frac{(2^{16})(32)f}{10^6} = 2.097152f \quad (2)$$

This provides a frequency resolution of approximately 1/2 Hz which is usually sufficient. However, some low frequency applications may require more precise frequency control. The waveshape, octave, and interrupt control byte at address k = 1 may be used to improve the frequency resolution. When bits 3,4, and 5 of this byte are all zero, a new phase is calculated each sample period. When they are non zero, a new phase is calculated only every 2^m th sample period where $m = b_3 + 2b_4 + 4b_5$ ($b_i = 0$ or 1). In effect, frequency f in eq. (2) is replaced by $f2^{-m}$, a division of 2, 4, 8, 16, 32 or 64 (only values of m such that $0 \leq m \leq 6$ are valid). Frequencies as low as .01 Hz and resolutions as small as .01 Hz may be generated using this control.

3. Δ Frequency Rate Control Byte (k = 6)

Δ Frequency Stop Value Control Byte (k=7)

Δ Frequency Current Value Word (k=014 and 015)

These four bytes are used to control and generate a linear time dependent function which is used to modify the frequency specified by the frequency control word. The frequency control word should be thought of as specifying the center frequency with high precision while the Δ frequency specifies positive and negative deviations from that frequency. The circuitry generates a ramp function by adding an increment value specified by the rate control byte to the Δ frequency current value, and the result is compared with a value specified by the Δ frequency stop value control byte. If the sum does not exceed the stop value, then the current value is replaced by the sum. If the sum exceeds the stop value, then the current value is replaced by the stop value. Both positive and negative rates and stop values may be specified. The ramp function is initialized by the microprocessor and the circuitry continually modifies the Δ frequency until the stop value is reached, and then maintains the stop value without any further microprocessor action. The Δ frequency current value has the same effect (i.e. the same scaling) as the frequency control word.

When the ramp function reaches the final value, it may be enabled to generate a program interrupt. Bit 6 of the control byte at address k=1 must be set to 1 to enable the interrupt feature. When this bit is zero, no interrupt will be generated. More details about the interrupt structure are given later.

The actual values specified by the rate and stop value bytes are determined by logarithmically coded read-only-memory (ROM) tables included in the circuitry. Bits 0 through 6 specify the magnitude m (127 non zero values) and bit 7 specifies the sign (0 = positive). The Δ frequency stop value (Δ FSV) is determined by the relation

FIG. 1

$$\Delta FSV = (14764)2^{[(m-127)/12]} \quad (3)$$

The maximum Δ frequency magnitude is 14764 (034654) which is specified by a 0177 (or 0377 for a negative Δ frequency) for the stop value control byte. This corresponds to a frequency of 7040 Hz or A^4 on the standard musical scale. Since there are twelve stop frequency values per factor of two (i.e. per octave), the stop values produce frequency changes equivalent to the pitches on the music scale from D_6 (4.8 Hz) to A^4 (7040 Hz). If the frequency control word ($k = 4$ and 5) is set to zero, then the Δ frequencies may be used to produce the music scale.

The ΔFSV is an integer value, thus the result of equation (3) is rounded to produce the nearest value. When $m = 0$, $\Delta FSV = 0$ also. Appendix III is a list of the actual ΔFSV values stored in the ROM table.

The Δ frequency increment value (ΔFIV) specified by the rate control byte is determined by the relation

$$\Delta FIV = (191)2^{[(m-127)/12]}$$

or

(4)

$$m = 127 + 17.312 \ln\left(\frac{\Delta FIV}{191}\right)$$

where m is the 7 bit magnitude of the rate control byte. Note that a 12 step change in the rate control byte produces a factor of two change in the ΔFIV value. A new value for the Δ frequency current value is calculated at a maximum rate of only 1/4 (8 kHz) of the oscillator output sampling rate. That is, the ΔFIV value is added to the Δ frequency current value only every fourth sample calculation. This rate is reduced by additional factors of two for small rate control byte values to achieve very slow rates of change. Although ΔFIV is an integer value, equivalent fractional values are achieved by automatically reducing the sampling rate of the ramp generating function when m becomes small. The appropriate control parameters to change the sample rate are included in the rate ROM table.

In practice, the Δ frequency ramp function is used to generate a frequency control envelope such as the one shown in Figure 2. This type of envelope is needed to synthesize the sounds of instruments such as horns which have a good deal of frequency and phase modulation during the attack portion of the note. The envelope must be scaled proportionally for different note frequencies so that the positive and negative frequency excursions are the same percentage of the final steady state frequency for all notes. The logarithmic coding of the stop and rate values greatly simplifies this scaling process.

The standard music scale uses 12 steps per factor of two change in frequency. In musical terms, this is 12 semitones per octave. Scaling the envelope for a note i semitones from the standard note is accomplished by changing all the rate and stop control byte values by i steps.

As an example, we will go through the details of generating the frequency envelope shown in Figure 2. Since the following description is a bit tedious, the casual reader should go on to part 4. The microprocessor will have to initialize control bytes at time A, the beginning of the note, and at time B, 50 msec after the note has begun. Remember that here we are describing a real time process where the note generation is initiated by pressing a key, by a clock driven file data source, or some similar event. The microprocessor must respond within a few msec to initialize or change control bytes.

Suppose we first generate the envelope for a final frequency of 440 Hz. At the beginning of the envelope, the frequency control word ($k = 4$ and 5) should be loaded with the value

$$(2.097152)(440) = 922 = 01632$$

from eq. (2). This will be the value used to generate the frequency after the initial 150 msec.

The microprocessor program should contain a table of values that relate the standard music pitches to the values needed for the frequency control word. The values in Appendix III may be used for such a

FIG. 2

table. Thus, each Δ frequency stop byte value corresponds to a semitone in the music scale. The frequency 440 would correspond to semitone 79 in such a table.

The Δ frequency ramp function must produce a frequency change of 20%, or about 88 Hz. From Appendix III, we find that a Δ frequency control byte value of 51 will produce a Δ frequency of 87.3 Hz, corresponding to an actual value of 183 as it would appear in the Δ frequency current value word ($k = 014$ and 015). During a 50 msec time period, the Δ frequency value must change from -183 to 183 (i.e. from -88 Hz to +88 Hz), or a total change of 366. In a 50 msec period, the Δ frequency value will be incremented 400 times, so that the Δ frequency increment value (ΔFIV) must be $366/400$. Using eq. (4), we find that the Δ frequency rate control byte ($k = 6$) should be set to 34 to achieve this value.

After the microprocessor has initialized the rate control byte and frequency control word, the Δ frequency current value word should be initialized. The same table used to find the value for the frequency control word may also be used to find the value for the Δ frequency current value word. Note that a 20% Δ frequency will always correspond to a table entry 28 semitones below the value used for the frequency control. The negative of the value in the table must be loaded in the Δ frequency current value word since the frequency is to start below the final frequency.

With these parameters initialized, the Δ frequency ramp will reach point B in Figure 2 in about 50 msec. Bit 6 in the waveshape, octave, and interrupt control byte ($k = 1$) may be set so that the oscillator circuit will generate a program interrupt to the microprocessor when the ramp reaches the final value. Thus, after the ramp function has been initialized, the microprocessor may perform other tasks until 50 msec later. The oscillator will then generate an interrupt to regain the attention of the microprocessor.

At point B, the Δ frequency rate must be reduced by a factor of 4 and be made negative. This is accomplished by subtracting 24 from the previous rate value (12 steps per factor 2) and then setting bit 7 to 1 to specify a negative rate. The resulting value is actually 0212. Then the Δ frequency final value control byte is loaded with 0. About 100 msec later, the ramp will reach zero value. The interrupt need not be enabled in this case since the envelop is complete, and no other microprocessor action is required.

Once the envelope generation control bytes are found for one frequency, they may be easily scaled by adding constants to the magnitude values. If the scaling causes the magnitude m to be outside the range $0 \leq m \leq 127$, then the envelope cannot be scaled properly and m should be set to 127, 0, or 1, depending on the desired results. Remember, a zero value for the rate control byte specifies a zero increment value, and the Δ frequency current value will not change at all.

Care should be taken regarding the order in which the rate and stop value control bytes are changed when byte mode addressing is used. The ramp calculation is not synchronized with the initialization of the control bytes by the microprocessor. For example, the next ramp sample may be calculated between the time when the rate value is loaded and the time when the stop value is loaded. Thus, for one sample calculation period, the stop value of the previous segment is used with the rate value of the next segment. If the signs of the current value, the stop value, and the rate indicate an overflow or overshoot has occurred during the calculation for that sample, then the current value will be replaced by the specified stop value. This may produce an unexpected or unwanted result. The problem may be much worse if the stop value is changed before the rate. A positive current value and positive rate with a negative stop value will cause the current value to be directly replaced with the stop value.

4. Waveshape, Octave, and Interrupt Control Byte ($k = 1$)

This control byte contains four separate and independent fields. Bits 0, 1, and 2 specify the waveshape the oscillator will produce. Up to 8 different shapes may be produced, but only four have been implemented at this time. These waveshapes are shown in Figure 3. They are stored in ROM

FIG. 3 WAVESHAPES

tables that have an effective size of 4096 words by 12 bits or 49,152 bits. However, by using symmetry properties and a sample interpolation technique, this is reduced to only 6144 bits, organized so that two waveshape tables are stored in three 512 word by 8 bit read only memories.

The three waveshapes made from straight line segments contain appreciable harmonic energy. Harmonic frequencies f_h which are greater than $1/2$ the sampling frequency f_s may produce audible aliased frequencies f_a which are not harmonically related to the fundamental frequency. The general relation is

$$|f_a| = f_h - \frac{if_s}{2} \quad \text{if} \quad \frac{if_s}{2} < f_h < \frac{(i+1)f_s}{2} \quad (5)$$

for $i = 1, 2, \dots$

These components become more noticeable as the frequency becomes higher, since more of the audio energy is at non-harmonically related frequencies. They are most noticeable as the frequency is ramped because the direction of the change in frequency of the aliased component is opposite that of the fundamental frequency. As the fundamental frequency is ramped up, the aliased component may be heard ramping down.

Bits 3, 4, and 5 may be used to lower the frequency specified by the frequency control word as described in section 2.

Bit 6 is used to enable and disable interrupt requests from the Δ frequency ramp function. When set to 1, the ramp will generate interrupts as long as its current value ($k = 014$ and 015) is equal to the final value specified by the stop value control byte. This bit must be cleared by the microprocessor to disable the interrupts.

Bit 7 is used in a similar fashion to enable and disable interrupt requests from the amplitude ramp function.

5. Amplitude Ramp Rate Control Byte ($k=010$)

Amplitude Stop Value Control Byte ($k=011$)

Amplitude Current Value Word ($k=016$ and 017)

These four bytes are used to control the amplitude ramp function in much the same way as the Δ frequency control bytes described in section 3 control the Δ frequency. Recall that the amplitude is controlled by phase cancellation. The 16 bit value in the amplitude current value word ($k= 016$ and 017) is the actual phase shift $\Delta\phi$ used in equation (1). When its value is zero, the two waves cancel exactly, and the output amplitude is zero. When its value is 077777 , the waves are exactly in phase and the peak output is a 13 bit number. Since the DSRs are 16 bit registers, the outputs of up to eight oscillators may be added together without danger of overflow.

The amplitude stop value control byte is interpreted as a sign-magnitude number, with bit 7 specifying either a positive or negative final amplitude value. The 7 bit magnitude is used to address a ROM table which contain the actual $\Delta\phi$ value need to produce the desired amplitude. The ROM table is coded so that a six step change in the stop value control byte produces a factor of two change in the amplitude if the waveshape being produced is a sine wave. Thus, a one step change in the stop control byte value corresponds to a 1 dB change in the amplitude. A control value of 0177 specifies full amplitude output.

The amplitude ramp rate control byte is also interpreted as a sign-magnitude value and is used to address a ROM table which contains an increment value. This increment value is added to the current value word every other sample period (16000 times per second) until the current value exceeds the specified stop value. The current value will be set to the stop value from then on until the control bytes are changed. The ramp function may be enabled to generate a microprocessor program interrupt when the ramp reaches its final value. Bit 7 of the control byte at $k = 1$ enables the interrupt.

The rate ROM table is coded so that a six step change in the rate control byte will produce a factor of two change in the increment value. The increment value I_A is calculated from the rate control byte magnitude m using the equation

$$I_A = (4095) 2^{(m-127)/6}$$

or

(6)

$$m = 127 + 8.656 \ln\left(\frac{I_A}{4095}\right)$$

When $m = 0177$ (127), it takes only 8 increments or 16 sample periods for the current value word to go from zero to full value. As I_A becomes small, the ROM table also generates control signals which cause the current value word to be incremented only every fourth, or eighth, etc. sample period. In this way, very small amplitude rates of change are achieved. The longest ramp time the circuit will produce is specified by $m = 2$. This causes the current value word to be incremented only 4 times per second. It takes over 2 hours to go from zero to full amplitude at this rate.

The logarithmic coding of the amplitude stop values and increment rates makes it easy to scale the amplitude envelopes. The prototype envelope should be specified so that it will produce the maximum loudness ever needed. If an envelope i dB quieter than this is required, it may be derived from the prototype envelope by subtracting i from each stop value magnitude. The rate magnitude may be reduced by i also if the scaling is to preserve the amount of time required to reach the stop value.

Recall from eq. (1) that the output amplitude is $\propto \sin(\Delta\phi)$ and that the ramp function causes $\Delta\phi$ to vary linearly with time. If the amplitude is ramped from full scale to zero, then the amplitude decay is a sine function and it sounds somewhat unnatural. The sound ends too abruptly. A more natural exponential like decay can be achieved by reducing the ramp rate periodically as the amplitude decays. Since the rate is coded logarithmically, subtracting 6 from its value at a regular rate (say every 400 msec for a decay that lasts 2 sec.) will produce a decay that sounds exponential.

The phase cancellation amplitude control produces an amplitude dependent waveshape for all waveshapes except a simple sine wave. The amplitude of the i^{th} harmonic component ($i = 3, 5, 7, \dots$) is $\propto \sin\left(\frac{i\pi}{2} \Delta\phi\right)$. Thus, each $\Delta\phi$ produces a unique set of harmonic amplitudes. This effect contributes to a more natural and interesting sound in most cases. The sound goes through interesting changes as it decays, similar to the effect a real musical instrument produces.

6. Timer Word (k = 012 and 013)

Each oscillator has an independent timer function that can make a microprocessor program interrupt request. The two bytes of the word are interpreted as a 16 bit 2's complement number. When loaded with a positive number (bit 7 of byte k = 013 is zero), the timer is enabled and will decrement the value once each msec. When the value reaches zero, interrupt requests will be made until the value is reset to a non zero value. The timer may be disabled by loading it with a negative value. The value will not decrement, and no interrupt requests will be made.

When the 16 bit timer is initialized to a value greater than 255 using an 8 bit microprocessor, a special loading sequence must be followed to prevent the oscillator from decrementing the timer word between the loading of the low order byte and the loading of the high order byte. First, the high order byte (k = 013) must be loaded with a negative value (i.e., the most significant bit = 1). This disables the decrementing process. Then the low order byte (k = 012) should be loaded with the proper value. Finally, the high order byte may be loaded. This sequence assures proper timer operation.

7. Interrupt Structure

Each oscillator has three different functions that can generate interrupt requests and there are 32

different oscillators. Thus, there are 96 different possible interrupt sources. These all use the same interrupt vector to the microprocessor. The oscillator circuit prepares a 7 bit interrupt message byte that the microprocessor may read through an I/O port. Bits 0-4 of this byte contain the number of the oscillator making the interrupt request. Bits 5 and 6 code the requesting device (1 = timer, 2 = Δ frequency ramp, and 3 = amplitude ramp).

Each of the 96 interrupt devices are enabled and disabled individually. A master interrupt enable must be set to allow any of these to actually interrupt the microprocessor. The details of this depends slightly upon the particular microprocessor.

S100 Bus Interface

Four I/O port addresses are used to control the interrupts in a S100 bus interface. The specific start address j of this group of four may be selected by 6 jumper wires. When j is read, bits 0 through 6 contain the interrupt message. When j is written (any value), bits 0 through 6 are cleared. No new interrupt messages will be generated until the microprocessor writes to j . When the oscillator generates a new message, at least one of these seven bits will be set to one.

The interrupts are enabled by writing any value to I/O port $j + 1$. No interrupt request will be generated unless this is done, although new messages may be generated. The interrupts are disabled by writing any value to I/O port $j + 2$.

When an interrupt request is made, the oscillator circuit will respond to a S100 bus INTA signal by asserting an 8 bit code on the data bus. This code may be set to any value using jumpers in the oscillator circuit. After an interrupt is granted, the interrupt enable is automatically cleared, and I/O port $j + 1$ must be written to before another interrupt request will be made.

LSI-11 Bus Interface

Four separate I/O word addresses (8 byte addresses) are used with the LSI-11 interface. The first address j of this group is set using jumpers in the oscillator circuit. The "event signal" is used with the LSI-11 to generate the interrupt. This causes the LSI-11 program to use the address stored at absolute address location 0100 as the starting address of the interrupt service routine.

The interrupt message may be read from location j in bits 0 through 6. Bit 7 will be set to 1 if there is a new interrupt message. This bit is automatically cleared when location j is read, and will be set to 1 when a new interrupt message is generated by the oscillator circuit. Note that a new message will probably be generated by the requesting device before the LSI-11 program can disable the device. Thus, after the program has serviced the requesting device, location j should be read again to clear bit 7.

Interrupts from the oscillator circuit are enabled by writing a 1 to bit 15 at address $j + 2$. The interrupt will be disabled automatically after a request is made, and must be reenabled by the LSI-11 program before another interrupt request will be made. The interrupt enable may also be cleared by writing a zero to bit 15 at location $j + 2$. The state of the interrupt enable may be read in the bit 15 position at location j .

8. Output Scale

A 12 bit D/A converter is connected to the 16 bit output of oscillator 31. This output may be connected in one of two ways, depending upon a control bit that is set by the microprocessor program. Bit 7 at I/O address $j + 3$ in the S100 interface is used for this purpose. Bit 8 at address $j + 4$ is used in the LSI-11 interface. When the scale control bit is zero, the D/A is connected to bits 1 through 12 of the oscillator output. The full amplitude output of a single oscillator will drive the D/A to full scale output with this connection. When the scale bit is set to 1, the D/A is connected to bits 4 through 15. With this connection, it takes the outputs of 8 oscillators added together to produce a full scale D/A output. The D/A output is at standard line level.

Additionally, the full 16 bit output from each oscillator is available time multiplexed at a connector on the oscillator circuit. Thus, the digital output is available to external circuitry for further processing or for a more precise D/A.

Conclusions

This inexpensive digital circuit provides enough features and computational capacity to accomplish substantial high quality real time music synthesis. Since it is designed to be used with small, inexpensive microprocessors, it should find widespread application. Although some compromises have been made to achieve economy, the output quality does meet high fidelity standards. If a 16 bit D/A is used at the output, a 90 dB dynamic range is achieved with about a 65 dB signal to noise ratio for a single oscillator. Properly programmed, it should be able to generate sounds as complex as those produced by four piece bands.

Appendix I

Memory Map

(Base address = Offset + 16*n*)

Relative Address	Control Byte
0	Patch
1	Waveshape, Octave, and Interrupt Control
2,3	Phase
4,5	Frequency
6	Δ Frequency Ramp Rate
7	Δ Frequency Stop Value
010	Amplitude Ramp Rate
011	Amplitude Stop Value
012,013	Timer
014,015	Δ Frequency Current Value
016,017	Amplitude Current Value ($\Delta\phi$)

S100 I/O Port

Address	
<i>j</i>	Read Interrupt Message
<i>j</i> + 1	Write To Enable Master Interrupt
<i>j</i> + 2	Write to Clear Master Interrupt
<i>j</i> + 3	Set or Clear Scale Bit

LSI-11 I/O Port

<i>j</i>	Read Interrupt Message
<i>j</i> + 2	Set or Clear Interrupt enable
<i>j</i> + 4	Set or Clear Scale Bit

Appendix II

Patch Control Byte (k=0)

Bits 0,1	DSR number for FM input source
Bit 2	FM disable (= 1 to disable FM)
Bit 3,4	DSR number for adder input source
Bit 5	Adder disable (= 1 to disable adder input)
Bit 6,7	DSR number to receive oscillator output

Waveshape, Octave, and Interrupt Control Byte (k=1)

Bits 0,1,2	Waveshape select 0 = sine wave 1 = symmetric triangle 2 = skewed triangle 3 = double peak 4 through 7 unused.
Bits 3,4,5	Octave control. Reduces the output frequency by octaves (up to a six octave reduction)
Bit 6	Δ Frequency ramp interrupt enable 1 = enabled 0 = disabled
Bit 7	Amplitude ramp interrupt enable 1 = enabled 0 = disabled

Phase (k=2 and 3)

16 bit 2's complement number which represents the instantaneous oscillator phase ϕ scaled such that

$$-\pi \leq \phi < \pi.$$

Frequency Control (k=4 and 5)

16 bit 2's complement number specifying oscillator frequency.

$$\text{Frequency control value} = 2.097152 f$$

where f = frequency in Hz.

Δ Frequency Ramp Rate (k=6)

Bits 0-6	Specifies the rate of change of the Δ frequency current value (k = 014 and 015) Rate of change = $1.528 \times 10^6 2^{(m-127)/12}$ per second for $1 \leq m \leq 127$ Rate of change = 0 for $m = 0$
Bit 7	Sign of rate of change (0 = positive)

Δ Frequency Stop Value (k=7)

Bits 0-6 Specifies the value of the Δ frequency current value (k= 014 and 015) when the Δ frequency ramp is completed

$$\text{Value} = (14764)2^{(m-127)/12}$$

$$\text{for } 1 \leq m \leq 127$$

$$\text{Value} = 0 \quad \text{for } m = 0$$

Bit 7 Sign of Δ frequency (0= positive)

Amplitude Ramp Rate (k=010)

Bits 0-6 Specifies the rate of change of the amplitude current value (k = 016 and 017)

$$\text{Rate of change} = 6.552 \times 10^7 2^{(m-127)/6} \text{ per second} \\ \text{for } 2 \leq m \leq 127$$

$$\text{Rate of change} = 2000 \text{ for } m = 1$$

$$\text{Rate of change} = 0 \quad \text{for } m = 0$$

Bit 7 Sign of the rate of change (0= positive)

Amplitude Stop Value (k=011)

Bits 0-6 Specifies oscillator output amplitude when the amplitude ramp has reached its final value

$$\text{amplitude} = (8191)2^{(m-127)/6}$$

Bit 7 No function (Only positive values allowed)

Timer (k=012 and 013)

16 bit 2's complement timer which decrements at the rate of 1000 per second if its value is positive. When it reaches zero, it will generate an interrupt request. The timer is disabled by loading it with a negative value.

Δ Frequency Current Value (k=014 and 015)

16 bit 2's complement number which is used to shift the frequency specified by the frequency control (k= 6 and 7). This value is ramped.

Amplitude Current Value (k=016 and 017)

16 bit 2's complement number which represents the phase shift $\Delta\phi$ used to control the oscillator output amplitude. This value is ramped.

I/O PORTS

S100 BUS

Address j	Interrupt Message
Bit 0-4	Contains the number of the oscillator making an interrupt request
Bit 5-6	Flag and code describing device making interrupt request 0 - no new interrupt message 1 - timer interrupt request 2 - Δ Frequency ramp interrupt request 3 - amplitude ramp interrupt request
Bit-7	Read state of scale bit 0 - bits 1-12 to D/A 1 - bits 4-15 to D/A

Writing to j clears bits 0-6 and allows a new interrupt message to be generated. Bit 7 is not affected.

Address $j + 1$	Set interrupt enable Writing any value enables interrupts This address reads the same as address j .
Address $j + 2$	Clear interrupt enable Writing any value disables interrupts. This address reads the same as address j .
Address $j + 3$	Write Scale Bit The value of bit 7 is written into the scale bit. This address reads the same as address j .

LSI-11 BUS

Address j	Interrupt message
Bit 0-4	Contains the number of the oscillator making an interrupt request
Bit 5-6	Code describing device making interrupt request 1 - timer interrupt request 2 - Δ frequency ramp interrupt request 3 - amplitude ramp interrupt request
Bit 7	Request flag 0 - no new interrupt message 1 - new interrupt message

Reading location j clears bit 7. Any new interrupt message sets it. Note that a new message may come almost immediately.

Bit 8	Scale bit state 0 - bits 1-12 to D/A
-------	---

1 - bits 4-15 to D/A

Bit 15

Interrupt enable

0 - interrupts disabled

1 - interrupts enabled

Address $j + 2$

Bit 15 is written into the interrupt enable. No other bits are affected. This address reads the same as j .

Address $j + 4$

Bit 8 is written into the scale bit. No other bits are affected. This address reads the same as j .

Appendix III

Δ FREQUENCY STOP VALUE CONTROL BYTE		CORRESPONDING NOTE NAME	ACTUAL VALUE		ACTUAL Δ FREQUENCY (H ₃)
127	0177	A ⁴	14764	034654	7040.0
126	0176	G ⁴ #	13935	033157	6644.7
125	0175	G ⁴	13153	031541	6271.8
124	0174	F ⁴ #	12414	030176	5919.5
123	0173	F ⁴	11718	026706	5587.6
122	0172	E ⁴	11060	025464	5273.8
121	0171	D ⁴ #	10439	024307	4977.7
120	0170	D ⁴	9853	023175	4698.3
119	0167	C ⁴ #	9300	022124	4434.6
118	0166	C ⁴	8778	021112	4185.7
117	0165	B ³	8286	020136	3951.1
116	0164	A ³ #	7820	017214	3728.9
115	0163	A ³	7382	016326	3520.0
114	0162	G ³ #	6967	015467	3322.1
113	0161	G ³	6576	014660	3135.7
112	0160	F ³ #	6207	014077	2959.7
111	0157	F ³	5859	013343	2793.8
110	0156	E ³	5530	012632	2636.9
109	0155	D ³ #	5219	012143	2488.6
108	0154	D ³	4926	011476	2348.9
107	0153	C ³ #	4650	011052	2217.3
106	0152	C ³	4389	010445	2092.8
105	0151	B ²	4143	010057	1975.5
104	0150	A ² #	3910	07506	1864.4
103	0147	A ²	3691	07153	1760.0
102	0146	G ² #	3483	06633	1660.8
101	0145	G ²	3288	06330	1567.8
100	0144	F ² #	3103	06037	1479.6
99	0143	F ²	2929	05561	1396.7
98	0142	E ²	2765	05315	1318.5
97	0141	D ² #	2609	05061	1244.1
96	0140	D ²	2463	04637	1174.4
95	0137	C ² #	2325	04425	1108.6
94	0136	C ²	2194	04222	1046.2
93	0135	B ¹	2071	04027	987.5
92	0134	A ¹ #	1955	03643	932.2
91	0133	A ¹	1845	03465	879.8
90	0132	G ¹ #	1741	03315	830.2
89	0131	G ¹	1644	03154	783.9
88	0130	F ¹ #	1551	03017	739.6
87	0127	F ¹	1464	02670	698.1
86	0126	E ¹	1382	02546	659.0
85	0125	D ¹ #	1304	02430	621.8
84	0124	D ¹	1231	02317	587.0

Δ FREQUENCY STOP VALUE CONTROL BYTE		CORRESPONDING NOTE NAME	ACTUAL VALUE		ACTUAL Δ FREQUENCY (H ₃)
83	0123	C ¹ #	1162	02212	554.1
82	0122	C ¹	1097	02111	523.1
81	0121	B	1035	02013	493.5
80	0120	A [#]	977	01721	465.9
79	0117	A	922	01632	439.6
78	0116	G [#]	870	01546	414.8
77	0115	G	822	01466	392.0
76	0114	F [#]	775	01407	369.5
75	0113	F	732	01334	349.0
74	0112	E	691	01263	329.5
73	0111	D [#]	652	01214	310.9
72	0110	D	615	01147	293.3
71	0107	C [#]	581	01105	277.0
70	0106	C	548	01044	261.3
69	0105	B ₁	517	01005	246.5
68	0104	A ₁ [#]	488	0750	232.7
67	0103	A ₁	461	0715	219.8
66	0102	G ₁ [#]	435	0663	207.4
65	0101	G ₁	411	0633	196.0
64	0100	F ₁ [#]	387	0603	184.5
63	077	F ₁	366	0556	174.5
62	076	E ₁	345	0531	164.5
61	075	D ₁ [#]	326	0506	155.4
60	074	D ₁	307	0463	146.4
59	073	C ₁ [#]	290	0442	138.3
58	072	C ₁	274	0422	130.7
57	071	B ₂	258	0402	123.0
56	070	A ₂ [#]	244	0364	116.3
55	067	A ₂	230	0346	109.7
54	066	G ₂ [#]	217	0331	103.5
53	065	G ₂	205	0315	97.8
52	064	F ₂ [#]	193	0301	92.0
51	063	F ₂	183	0267	87.3
50	062	E ₂	172	0254	82.0
49	061	D ₂ [#]	163	0243	77.7
48	060	D ₂	153	0231	73.0
47	057	C ₂ [#]	145	0221	69.1
46	056	C ₂	137	0211	65.3
45	055	B ₃	129	0201	61.5
44	054	A ₃ [#]	122	0172	58.2
43	053	A ₃	115	0163	54.8
42	052	G ₃ [#]	108	0154	51.5
41	051	G ₃	102	0146	48.6
40	050	F ₃ [#]	96	0140	45.8
39	047	F ₃	91	0133	43.4
38	046	E ₃	86	0126	41.0
37	045	D ₃ [#]	81	0121	38.6
36	044	D ₃	76	0114	36.2
35	043	C ₃ [#]	72	0110	34.3
34	042	C ₃	68	0104	32.4
33	041	B ₄	64	0100	30.5
32	040	A ₄ [#]	61	075	29.1

Δ FREQUENCY STOP VALUE CONTROL LYTE		CORRESPONDING NOTE NAME	ACTUAL VALUE		ACTUAL Δ FREQUENCY (H ₃)
31	037	A ₄	57	071	27.2
30	036	G ₄ [#]	54	066	25.7
29	035	G ₄	51	063	24.3
28	034	F ₄ [#]	48	060	22.9
27	033	F ₄	45	055	21.5
26	032	E ₄	43	053	20.5
25	031	D ₄ [#]	40	050	19.1
24	030	D ₄	38	046	18.1
23	027	C ₄ [#]	36	044	17.2
22	026	C ₄	34	042	16.2
21	025	B ₅	32	040	15.3
20	024	A ₅ [#]	30	036	14.3
19	023	A ₅	28	034	13.4
18	022	G ₅ [#]	27	033	12.9
17	021	G ₅	25	031	11.9
16	020	F ₅ [#]	24	030	11.4
15	017	F ₅	22	026	10.5
14	016	E ₅	21	025	10.0
13	015	D ₅ [#]	20	024	9.5
12	014	D ₅	19	023	9.1
11	013	C ₅ [#]	18	022	8.6
10	012	C ₅	17	021	8.1
9	011	B ₆	16	020	7.6
8	010	A ₆ [#]	15	017	7.2
7	07	A ₆	14	016	6.7
6	06	G ₆ [#]	13	015	6.2
5	05	G ₆	12	014	5.7
4	04	F ₆ [#]	12	014	5.7
3	03	F ₆	11	013	5.2
2	02	E ₆	10	012	4.8
1	01	D ₆ [#]	10	012	4.8
0	0	-	0	0	0