Theremin with Onboard Effects

by
Patrick Tarantino
Shaun Cinnamon
PHYCS 398

Abstract

The theremin is a completely electronic musical instrument which is controlled by hand capacitance effects. The small capacitance formed between an antenna and a person's hand is used to change the frequency of an oscillator, creating a variable tone frequency. Another such circuit is used to control the volume of the tone created. This particular design also utilizes a DSP to alter the basic sound of the output, simulating 3 different instrument sounds.

Table of Contents

1. Basic Information

1.1 Introduction

The basic concept behind the operation of the theremin is that of hand capacitance. Simply put, a small but significant capacitance exists between any antenna or antenna-like device and the hand of a person who is presumably grounded through their connection with the earth. As the hand is moved closer to the antenna, the capacitance value increases, and vice-versa. If such an antenna is added to some form of an oscillator circuit, whether it is an LC resonant circuit, or a more modern function generator, the variable capacitance can be used to alter the frequency, which is outputted by the oscillator. However, the maximum value of hand capacitance is small, and in fact was measured during the course of this project to be around 10-20 pF, depending on humidity and weather conditions. The average value was about 12 pF. Because this value is quite small, the unmodified, or base frequency of the variable frequency oscillator (VFO) should be fairly high, at least several hundred kHz, in order that a capacitance change of this magnitude will create a large enough frequency change. The signal from the VFO is then mixed with that of a fixed-frequency oscillator with an operating frequency equal to the base frequency of the VFO, and the result is sent through a low-pass filter, resulting in a frequency equal to the difference between the two oscillator frequencies. This basic setup is duplicated in the volume control circuit, and the two signals are mixed again in such a way that the first antenna changes the pitch of the output signal, while the other changes its magnitude. With the inclusion of DSP effects, the pitch output will be digitally altered before it is applied to the volume control circuit, but this does not change the fundamental operation of the theremin. The original design specifications of this project were as follows:

1. Pitch should have at least 3 octaves range
2. Antennae should have good range of sensitivity
3. Completed circuit should be fairly temperature insensitive
4. DSP effects should simulate at least one other instrument sound

Fig. 1 is a block diagram of the completed project as it was originally intended to be.

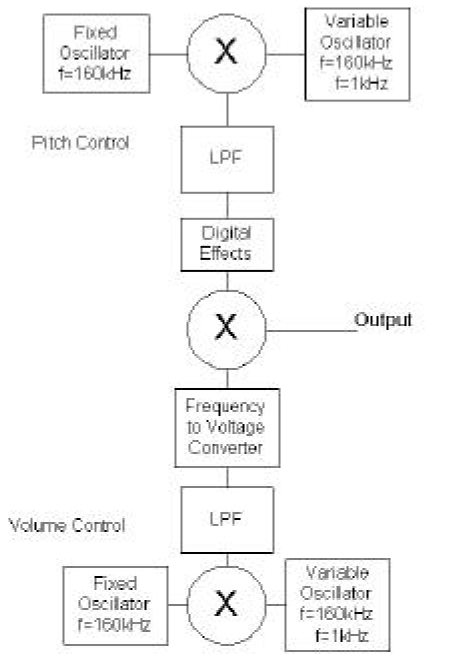

Fig. 1

The design of this project was logically split into two parts, the analog design of the basic theremin, and the programming of the DSP to implement the other instrument sounds.

2. Analog Design

2.1 Analog Theremin Design

In the initial phase of the analog design, frequency generator IC chips were chosen for the implementation of the oscillators. There are several reasons for this decision. First, IC chips are generally very temperature insensitive, especially when compared with resonant circuits using inductors, which are often very sensitive to temperature. This means that the completed circuit would operate well regardless of temperature. Also, IC chips are very inexpensive, and while inductors are also fairly inexpensive, if a specific hard-to-find inductor had been needed, it would have increased the price and made the availability unreasonable. Fewer components are needed when using an IC oscillator, which also reduces cost. Finally, using IC chips makes a circuit smaller, less complex, and easier to repair. With all of this in mind, Intersil 8038 frequency generator chips were chosen. These chips use an external capacitor and two resistors that determine the timing, and the formula for the frequency of the chip's output is seen in Eq. 1.

$$f = \frac{1}{2pRC} \qquad (1)$$

Where R equals $R_1 || R_2$.

A specification of this type of chip is that the maximum usable frequency that can be provided is about 200 kHz. Next, the mixers were chosen. After experimenting with two other mixer setups, an Analog Devices 633 IC analog mixer chip was chosen, partially for the same reasons above, and partially due to its easy use. The mixer simply multiplies two signals and divides the result by a constant scaling factor, as seen in Eq. 2.

$$\text{output} = \frac{(V_1)(V_2)}{10V} \qquad (2)$$

Where V_1 is one of the signals and V_2 is the other signal.

When two sinusoidal signals are multiplied, the result contains a low frequency component, the difference or beat frequency, and a high frequency component, which is twice the base frequency, as seen in Eq. 3.

$$\begin{aligned} \cos(wt) * \cos((w + \Delta w)t) &= \cos(\Delta wt) * \cos^2(wt) \\ &\quad - \sin(\Delta wt) * \sin(wt) * \cos(wt) \\ &= \frac{1}{2}\cos(\Delta wt) * (1 - \cos(2wt)) \\ &\quad - \frac{1}{2}\sin(\Delta wt) * \sin(2wt) \end{aligned} \qquad (3)$$

Using a low-pass filter allows the difference frequency to be isolated, which is the goal, since the difference frequency is in the audible range. Finally, a National Semiconductor LM2917N frequency-to-voltage chip was chosen for use in the volume control circuit. This would allow the variable frequency signal to be reduced to a dc signal with a voltage proportional to the frequency of the input. This would make volume control easy and straightforward. Other possible choices for this part were an op-amp in a differentiator setup or a transistor design. The chip was again chosen for the benefits of IC devices. This chip, like the frequency generator, also requires the use of external capacitors and resistors, and simply produces a constant voltage which is defined in Eq. 4.

$$\text{output} = f_{in} * V_{cc} * R_1 * C_1 \qquad (4)$$

With all of these elements chosen, the pitch circuit and the volume circuit are complete. Figs. 2 shows a complete diagram of the pitch circuit that was used for the project.

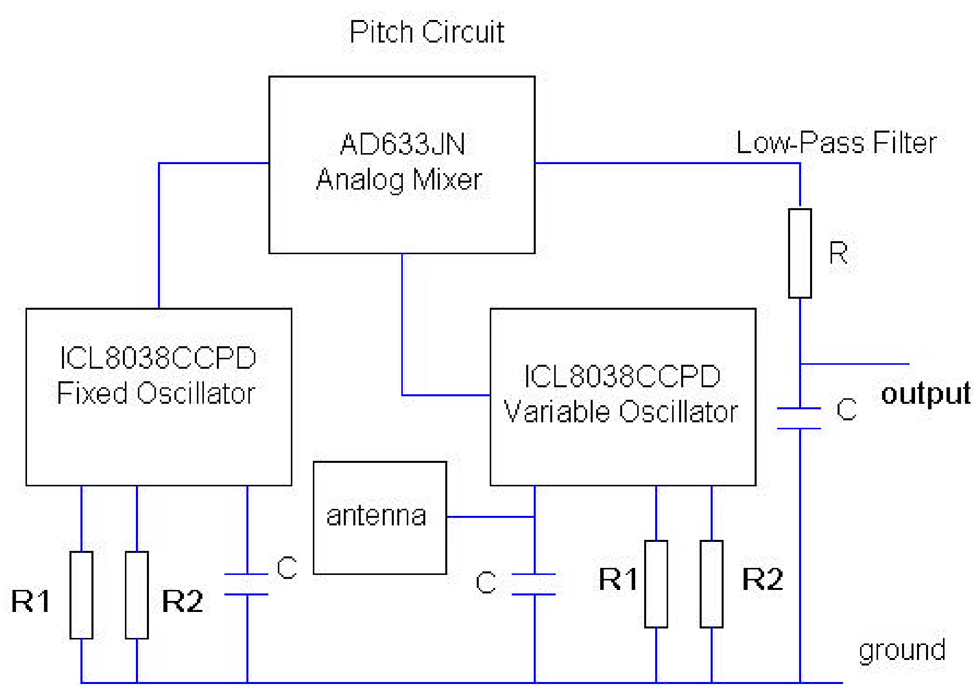

Fig. 2

Fig. 3 shows a diagram of the complete volume circuit that was used.

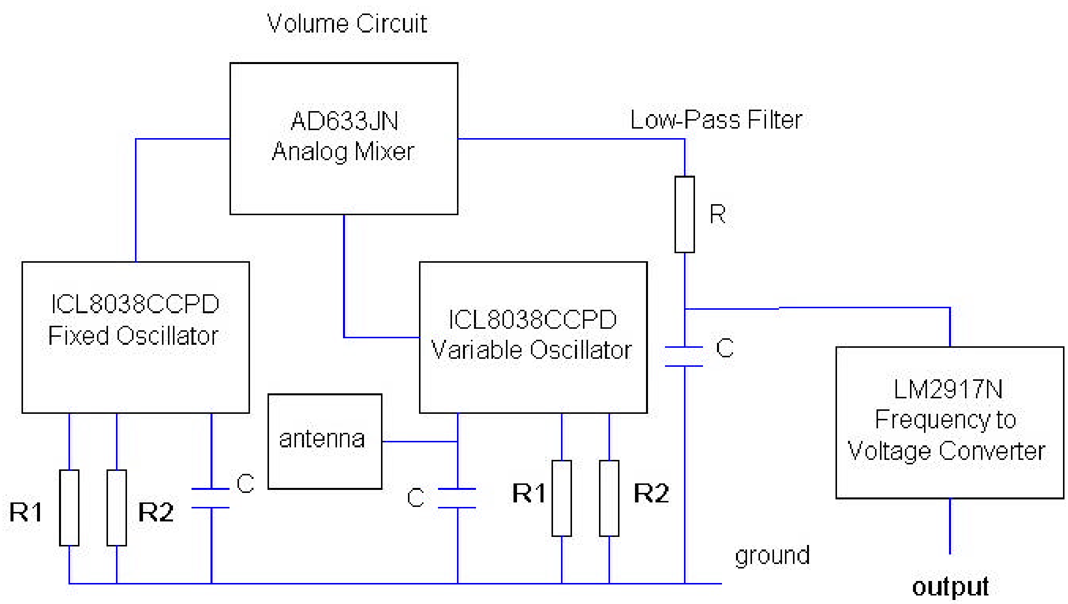

Fig. 3

The pitch circuit generates a sinusoidal output in the audible range, which is equal to zero when no hand capacitance is sensed and increases in frequency as the hand capacitance increases. The volume circuit generates a constant voltage which is equal to zero when no hand capacitance is sensed and increases as the hand capacitance increases. Using a third mixer, these two signals are multiplied, creating a variable pitch with a variable volume. As a final constraint, the oscillator pairs were to be designed in such a way that there would be no difference frequency when no hand capacitance was sensed. This called for using timing resistors and capacitors which were as close to identical as possible for each oscillator pair.

2.2 Design Details

First, the values of the passive timing elements were chosen for the 8038 chips. Since a high frequency was desired, it was decided that 1kΩ resistors would be used and capacitors that would give a frequency as close to the maximum as possible without exceeding it. After testing with several values, and having also been limited by the available capacitors in the stores, a value of 2020 pF was chosen, giving a theoretical frequency of 157 kHz, using Eq. 1.

The actual frequency obtained was about 160 kHz, with some variation due to precision errors in these passive elements. The antennae used in conjunction with both circuits were square aluminum plates, 8 in by 5 in, with a thickness of $48*10^{-3}$ in. Also, at this frequency, the maximum frequency difference was noted to be about 1 kHz. This seemed to fulfill the design specification of at least 3 octaves of musical range, since a signal going from zero to 1 kHz goes through nearly 6 octaves. Using an example circuit provided in the spec sheets for the LM2917N, a 20 pF capacitor and a 100 kΩ resistor were chosen for timing in the frequency-to-voltage converter, with a 1 μF capacitor, a 470 Ω resistor, and a 10 kΩ resistor also used to reduce ripple and ensure proper current levels. With these components, the conversion rate was about 15 V/kHz, an acceptable level, considering the constant scaling factor of 1/10 in the mixers.

Finally, for the two low-pass filters used, a corner frequency of at least 5 kHz was desired. To this end, two 2700 Ω resistors and two 3 nF capacitors were selected, giving an actual corner frequency of about 20 kHz.

2.3 Verification of the Basic Theremin Circuits

Testing was first conducted on the basic pitch circuit. The output signal from the pitch circuit, when finally assembled, had a peak-to-peak amplitude of about 100 mV, and the circuit was shown to function correctly, but several problems were discovered. First, the circuit could only sense the hand capacitance about two inches and closer. This is much less than the range of a commercial theremin, which can sense the hand's presence from at least a foot away. This is mostly due to the base frequency being limited by the 8038 chip. If the base frequency could have been much higher, at least twice as large, the circuit would have been able to sense the capacitance better. Also, the thickness and material of the antenna had some effect on the sensitivity, but a piece of metal of appropriate thickness, about twice the thickness used, could not be found in time. Next, because of the precision errors in the passive timing elements used, especially the capacitors, the same unmodified frequency could not be obtained in both oscillators of each pair. This meant that a constant offset frequency of about 1 kHz was present in the output signal which could not be filtered out. Although a 1 kHz difference is very small compared to the base frequency of 160 kHz, it is equal to the maximum value of the desired output frequency, and so highly distorts the output from the desired range. As previously stated, a 1 kHz range with a base of zero is about 6 octaves, but a 1 kHz range with a base of 1 kHz is only about a 1 octave range, an immense decrease in musical range.

Furthermore, since these offsets were present in both oscillator pairs, the volume circuit would not function properly. This is because with such a large offset it is very difficult to make sure that the constant frequency scaling factor of the frequency-to-voltage converter reduces the input signal to a usable level while still providing enough of a change due to hand capacitance. In other words, if the scaling factor was adjusted to accommodate the constant offset of 1 kHz, the change due to hand capacitance would become very small, and, for practical purposes, unusable. Even when combined with an op-amp to magnify the output, the changing dc level was unpredictable at best, and so the volume circuit was eventually abandoned.

3. Digital Design

3.1 Introduction to Digital Effects

The digital effects used in this project are transforming the output from an analog Theremin into three different musical instruments, along with the original signal amplified to about the same magnitude as the musical instruments. The transformation of the output is done through Wavetable Synthesis. Wavetable Synthesis is done by saving one period of a musical tone into a table and using table look up to play the signal back. To create a different frequency the table needs to be played back at a different rate. The rate of play back is determined by the step size through the table. For example, one person plays the table back at a step size of 3 and another plays it back at a step size of 1. The first person moves through the table three times as fast as the second person and therefore the frequency is three times that of the second frequency. This is the fundamental idea behind the digital effects used in this project.

3.2 Multiple Wavetable Synthesis

To generate simple waveforms the previous method of Wavetable Synthesis is adequate. However, if more complicated waveforms are to be reproduced then multiple tables are needed to recreate the sound, this type of Wavetable synthesis is called Multiple Wavetable Synthesis. The fundamentals behind Multiple Wavetable Synthesis are the same as Wavetable Synthesis. There is another distinguishing characteristic besides the number of tables that are used and that is amplitude envelopes are also included in recreating the instrument. The amplitude envelopes are used to give a more authentic sound to the recreated signal. The wavetables and amplitude envelopes can be seen in Fig. 4.

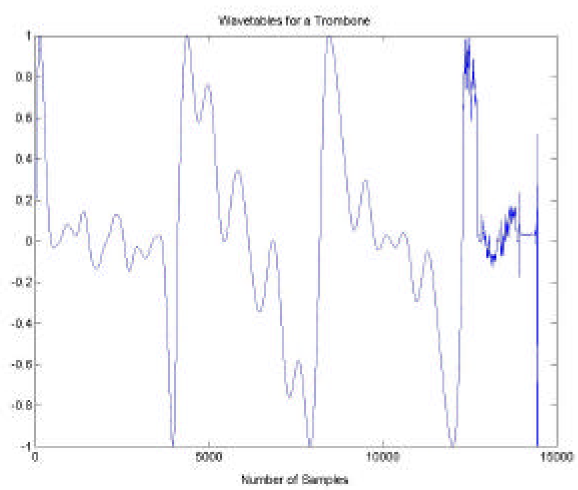

Fig 4

The figure above contains three wavetables, three amplitude envelopes and what is thought to be a frequency shift table. The first wavetable comes from the first 4096 samples and the second comes from the next 4096 and the final wavetable comes from the next 4096. The amplitude envelopes come from the next 1500 samples and each amplitude envelope is 500 samples long coming one after another. The rest of the samples contain the frequency shift table.

Once the wavetables and amplitude envelopes are created there are two variables that need to be determined to recreate the signal. The first is the step size through the table. The step size can be determined by knowing the frequency, in Hertz, that the signal should be played back at, the length of the wavetable, in samples, and the sampling rate, in samples per second. The equation for the step size is Eq. 5.

$$\text{step size} = \frac{\text{frequency} * \text{table length}}{\text{sample frequency}} \qquad (5)$$

The next variable to be determined is the amplitude envelope step size. Eq. 6 shows how to determine the step size for the amplitude envelopes where the duration, in seconds, is how long the sound will be played, the length of the amplitude envelope, in samples, and the sampling rate is the same as that of the wavetable.

$$\text{Amplitude step size} = \frac{\text{table length}}{\text{duration} * \text{sample frequency}} \qquad (6)$$

With these two quantities determined the individual wavetable outputs are calculated using Eq. 7.

$$\text{Output} = \text{Amplitude} * \text{Wavetable} \qquad (7)$$

Once all individual wavetable outputs are determined the total output is calculated by adding all the individual outputs together as seen in Eq. 8.

$$\text{Total Output} = \sum_{i=0}^{N} \text{Output[i]} \qquad (8)$$

Where Output[i] is the output from the individual wavetables and N is the total number of wavetables used.

One question arises as a result of calculating the step sizes this way. That is what happens when there is a non-integer value for the step size? There are three solutions to this problem and those solutions are truncation, rounding, and linear interpolation. In truncation the fractional part of the step size is not used. Whereas in rounding the fractional part is used to determine the closest integer to the current value and in linear interpolation a value is taken in between the two values in the table. There are several factors that determine which method should be used. However, for the amplitude envelope step size linear interpolation should always be used. This is due to the fact that the amplitude envelopes are generally shorter in length and a more realistic sound results if linear interpolation is used in these shorter tables.

For the digital effects that are used for this projects slight modifications of these fundamentals were made to fit into what was to be accomplished. These modifications will be addressed in the following sections.

3.3 Zero-Crossing Detection

Zero-Crossing Detection was used to find the frequency of the output from the Theremin. Due to the Theremin's output being a continuous sinusoid in nature, there was a need to determine frequency of the signal at run time. Knowing that the Theremin's output is a sinusoid the idea that the output will only cross zero twice per period was used to determine the frequency. Counting the number of samples that pass every period, the frequency of the signal can be determined using Eq. 9.

$$\text{frequency} = \frac{\text{sample frequency}}{\text{samples per period}} \qquad (9)$$

A simple test of this algorithm can be seen in the example shown in Fig 5. From Fig. 5 it looks as though the signal crosses zero three times per period. However, the first zero crossing can be considered either in the current period or the previous period. Taking the first zero crossing to be in the previous period it is easily seen that the signal will cross zero twice per period.

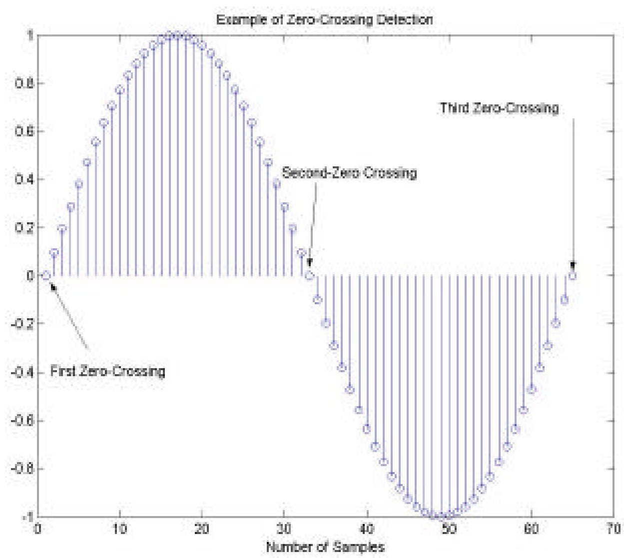

Fig 5

Once the determination of the period is set, the number of samples per period is then calculated by counting the number of samples that pass before the second zero-crossing and the samples that pass between the second zero-crossing and the third zero-crossing. Once the number of samples per period was determined this could then be used in Eq. 5 to obtain the step size through the table.

3.4 Implementation

For this project, three Wavetables were used per instrument, however their corresponding amplitude envelopes were excluded. There are two reasons why the amplitude envelopes were not included. The first reason comes from Eq. 6 in which the amplitude step size is determined by the duration of the recreated signal. Since, the duration of the Theremin can be taken as infinite the amplitude increment is zero and therefore the amplitude envelopes would be stationary and not performing the way they were intended. Another reason that the amplitude tables were excluded is the amount of memory that was to be used. Trying to have three instruments meant trying to store a total of nine wavetables along with their corresponding amplitude. This corresponded to storing 41,364 words in memory along with the program code and the core file that was used. This resulted in more words needing to be store in memory then what was allowed and since the step size for the amplitude envelopes could be considered to be zero, they were excluded for the project.

Once the problem with memory was cleared up and the amplitude wavetables were excluded it was useful to add up the wavetables in Matlab to increase the number of instruments that could be played. However, the number of instruments that were available was very limited. Of these instruments only a number of them sounded good enough to be included in this project. As a result of this a trombone, a bass and a clarinet were used in the project.

The next part of the project that needed to be completed was the zero-crossing detection. Since, the program was working on a sample-by-sample basis the detection was very easy to implement. When a new sample came in the sign of the sample was determined. This was done using branch statements to branch if the sample was negative, positive or zero. Depending upon the branch another comparison was made. For a negative sample, the comparison was to check if the previous sign was positive. If the previous sign was positive, a test bit would be changed to indicate the comparison was false and subsequently there would be a branch. Before the branch would occur the previous sign value would be changed to the current samples sign. There were then two branches to be made, a branch to increase the number of zeroes crossed or a branch to increase the number of samples. The branch to increase the number of zeroes consisted of adding one to the previous amount and comparing the new amount to two. If the number of zeroes crossed is less than two a branch to increase the number samples is made. If the number of zeroes is equal to two then the number of samples would be preserved in another variable and the zero count would be returned to zero along with the number of samples. If the zero count did not need to be incremented then one would be added to the number of samples. This would be done every time through the program. The same types of comparisons were made for both a positive and a zero valued sample.

After the number of samples per period was determined the next step was to determine if an instrument was to be played or if the Theremin's original output should be played. A real time user interface written in Matlab accomplished this task. The Matlab interface is connected through the serial port between the computer and the DSP. The user interface was a slider that could be adjusted between instruments or no instrument. Once the desired effect was chosen in Matlab, a value between zero and three would be sent to a hold variable on the DSP. Depending on the value sent to the DSP, would branch to the appropriate effect. For no effects at all the output would be amplified to bring it up to unity with all the other effects. This was needed because the instruments have constant amplitude to them and in order to use the volume control circuit correctly the amplitudes needed to be in unity. If one of the instruments were chosen then another branch was made to apply the correct effects.

Once a branch was made to one of the instruments the step size for the table was to be determined. Using the value obtained from the zero-crossing detection the step size was obtained. However, the step size was converted from Eq. 5 to Eq. 10 using Eq. 9.

$$\text{Step Size} = \frac{\text{Table Length}}{\text{Number of Samples}} \qquad (10)$$

This caused a slight problem, because the DSP cannot explicitly divide. As a result a search of Texas Instruments', manufacturer of the DSP that was used, web site to find out how to divide on the DSP was performed. Code that would perform the division that was needed was found, however it not written for the DSP that was used. Therefore the code needed to be deciphered, in order for it to be applied. Once the code was deciphered it was put in place to calculate the step size.

After the step size was calculated the output of the instrument was prepared. Since, the addition of the three tables was done in Matlab already it was simply a matter of getting the value in the table into the output buffer to be sent to the volume control circuit or to a speaker.

Next came the preparation of the table to be used for the next time through. The step size that was calculated earlier was added to the previous point in the table. Since, the step size was not always going to be an integer value, rounding was used to determine the next table location. The one feature that came from the division code used, was that the quotient and the remainder were saved into variables. The quotient was used as the step size. While, the remainder was then used to round up or down to the next table location. There was no simple way to do the rounding, because of the continuously changing frequency and therefore the denominator was continuously changing too. So a simple way to test this was to subtract the current denominator from the entire fractional part. If the result was less then zero then no rounding occurred. However, if the result was greater than or equal to zero then rounding occurred and the total fractional part was zeroed out. Rounding was the simplest method to get the range of frequencies that the Theremin would output.

Succeeding the determination of the new table location, a check was performed to make sure that the new location was not off the end of the table. To check where the location was, the highest the value that the table could get was subtracted from the current table location. If the result was less than zero then nothing was done to the current location. If the result was positive then the result was added to the beginning of the table and this became the new table location.

Once the check was performed the programmed looped back to the beginning to perform all the calculations over until the user stopped the program.

4. Integration

4.1 Integrating the Two Sections

When both sections were completed, an integration was attempted. Although the volume circuit was not functional, the pitch circuit and the DSP worked together as planned with no extra adjustments. The output signal from the DSP had a constant amplitude of about 200 mV peak-to-peak.

Each instrument sound was tested to ensure that it was functioning correctly with the pitch circuit. Although everything worked fine, it was difficult to tell the difference between the instrument sounds while using the pitch circuit. This is because of the constant 1 kHz offset. These instrument simulations were clearly designed to work best at relatively low frequencies. When the instrument sounds were retested using a lab function generator at 100 Hz, the instruments were easily distinguishable and sounded much better.

5. Final Analysis

5.1 Costs

Labor cost estimates:

Patrick – ($50/hr)(55 hr)(2.5) = $6875
Shaun – ($50/hr)(110 hr)(2.5) = $13,750

Parts cost estimates:

$50

Total estimated cost:

$20,675

5.2 Conclusions

The theremin, even in conjunction with digital effects, can easily and cheaply be implemented. Even using cheap components and passive components with very poor precision, the basic circuit will work. The prime considerations for a marketable theremin are stable, precise, easily set, high frequency oscillators, and appropriate antenna construction and configuration. While the benefits of IC's are well known, here the operation was hindered by the use of IC's. A good possible alternative would be to have a transistorized oscillator setup, such as a Colpitts circuit, in which the two oscillators of each pair could be coupled so as to lock in to zero frequency when no hand capacitance is sensed. An alternative sensing scheme would include a photodetector setup, which could either work on the reduction of incident light by a human hand, or, with an infrared detector, could work based on human heat emission. These alternatives, while quite a bit more complicated and possibly more expensive, would definitely be more precise, and would lend themselves well to a completely digital theremin design.

Another possible alternative would be to include the amplitude envelopes when doing the calculations for the digital effects. This could be accomplished by having more memory available to use in conjunction with the DSP. Also, a constant duration time could be used which was tried. This was tried but due to the memory limitations the program would not run correctly.

6. References

Horner, Beauchamp, and Haken, "Methods for Multiple Wavetable Synthesis of Musical Tones," Journal of the Audio Engineering Society, vol. 41, No 5, pp. 336-356, May 1993.

Lent, " An Efficient Method for Digitally Shifting Digitally Sampled Sounds," Computer Music Journal, vol. 13, no. 4, pp.65-71, Winter 1989.

Peter Shargo and Greg Bishop, "Multiple Wavetable Synthesis," ECE 320 Final Project Report, Univ. of Illinois, Urbana, IL, 1999.

TMS320C54x DSP Reference Set, Volume 4: Applications Guide, SPRU173, Texas Instruments Inc., 1996.