2014

M.Sc. Electronic Physics – Radioelectrology
Aristotle University of Thessaloniki
Physics Department

Dimitrios Porlidas
www.porlidas.gr

[IMPLEMENTATION OF A MUSIC SIGNAL SYNTHESIZER IN FPGA]

TABLE OF CONTENTS

1. ABSTRACT

A composite waveform can be analyzed and described in a simpler way that can be processed and reproduced. The Fourier Analysis is a method that gives us this possibility by summing infinite series of sine and cosine functions. With Fourier analysis and by suitable electronic devices, such as waveform generators, it is possible to represent a waveform in the form of electrical signal. A waveform generator can be implemented in an FPGA. The FPGAs are integrated circuits, that have the ability to parameterized with programming and behave like almost any digital circuit or device.

The sound of musical instruments has the characteristics of complex waveforms. It is possible to analyze these sounds into series of sinusoidal functions, and using a waveform generator to implement a music signal synthesizer, in order to reproduce similar sounds and simulates various musical organs. The following describes the method of implementation of a music signal synthesizer in FPGA and the results presented.

2. WAVEFORM GENERATOR

2.1 FURIER SERIES

The behavior of a system of independent and dependent variables can be plotted and find a mathematical expression that defines the relationship between the variables. When an independent variable is associated with a dependent and the mapping of their relationship is plotted in a rectangular coordinate system, the graph may be a curve or may have repetitive segments and thus is a waveform. The mathematical form of the relationship between variables is the function that describes the curve or waveform.

A composite waveform can be analyzed and described in a more simple way in order to be processed and reproduced. Fourier Analysis is a method that gives us this opportunity by summing infinite sine and cosine functions. It is therefore possible, for a waveform defined on an interval (-π, π) to find the frequencies that exist in this and their widths, so that the waveform can be described by the sum of:

$$g(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty}(a_n \cos 2\pi f_n t + b_n \sin 2\pi f_n t)$$

The amplitudes for each frequency f_n = nf, where f is the fundamental frequency, are given by:

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} g(t)\cos(n\pi f t)\,dt$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} g(t)\sin(n\pi f t)\,dt$$

As an example we can consider the step function shown in Figure 1 where the vertical axis represents amplitude and the horizontal time. In Figures 2, 3 and 4 are shown the cosine functions involved to synthesize the step function of Figure 1 and in Figure 5 the sum of these three functions. In this example the series is finite (sum of three cosine) and for this can be distinguished differences in the form between the step function and function calculated by the Fourier analysis.

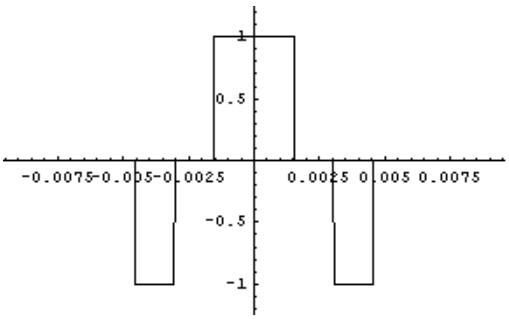

Figure 1. Step function.

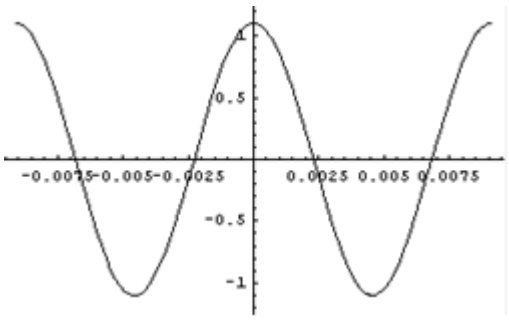

Figure 2. Cosine function f = 220Hz.

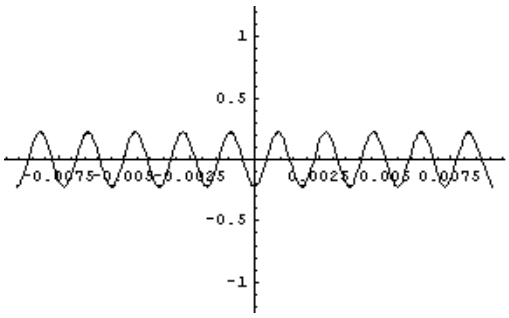

Figure 3. Cosine function f = 1100Hz.

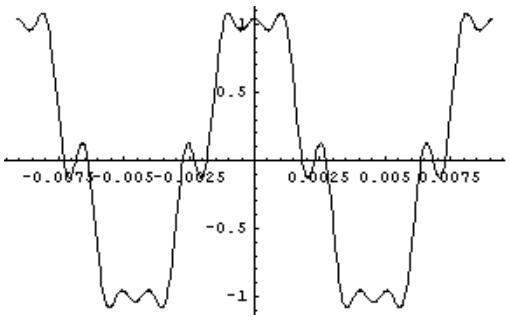

Figure 5. Sum of functions.

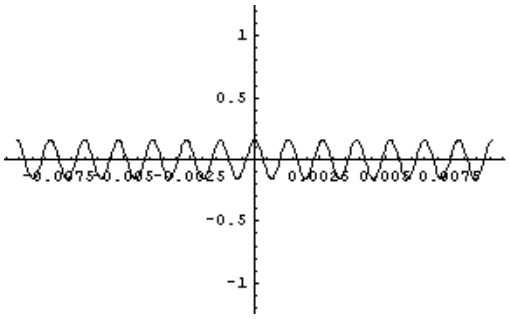

Figure 4. Cosine function f = 1540Hz.

2.2 FPGA

It is possible to use electronic devices in order a waveform can be displayed in the form of electrical signal. Since the waveform has been analyzed in a sum of sines and cosines, and have been calculated the frequencies and amplitudes, electronic circuits can be designed to synthesize these frequencies and aggregate them. The resulting electrical signal is identical to the waveform and the differences between them are minimized as increase the terms of the sum.

The electronic device used in this project as a generator to generate waveforms based on an Integrated Circuit (IC) called Field Programmable Gate Array (FPGA) which operates in combination with a Digital to Analog Converter (DAC). The operating principle of the device based on digital synthesis and summation of frequencies in FPGA.

The FPGAs are integrated circuits, which are designed to be configured by the engineer or the designer of the application for which will be used. For the configuration used a Hardware Description Language (HDL). The FPGA then electronically configured by download the code (programming) with auxiliary devices. After programming, FPGAs can behave as almost any digital circuit or device. The structure includes programmable logic elements called Logic Blocks (LB), and internal connections, which allow LBs linked together and combined to perform complex combinational functions.

In the waveform generator used FPGA XC3S500E from the family of Spartan-3E Xilinx. It consists of approximately 500,000 gates which implement various types of LBs:

- Configurable Logic Blocks (CLBs)
- Input/Output Blocks (IOBs)
- RAM Blocks
- Multiplier Blocks
- Digital Clock Manager Blocks (DCM)

The Logic Blocks are arranged in a suitable manner to allow the combination (Figure 6).

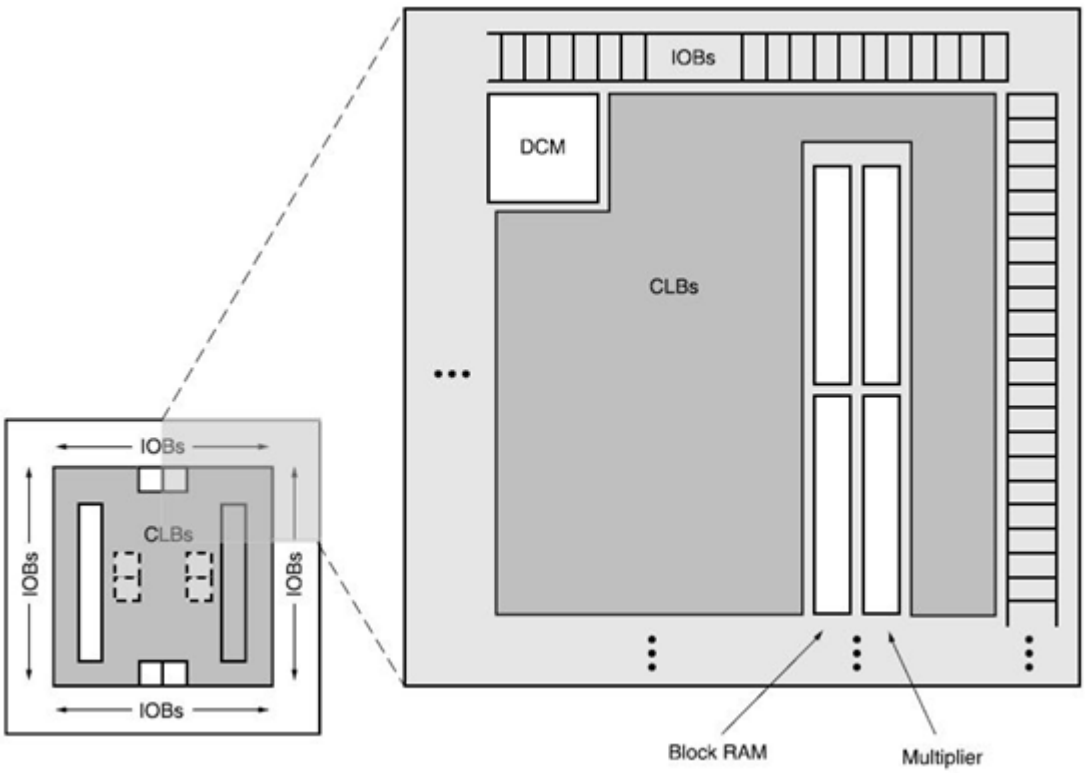

Figure 6. Order of Logic Blocks in FPGA.

The CLBs are the main devices that implement synchronous as well as combinational circuits and organized and arranged in columns and rows. Each CLB consists of 4 slices. Each slice contains 2 Look-Up Tables of four inputs (4-input LTUs), which, in 2 of the 4 slices, can operate as memory 16bit (16x1 RAM block) or as shift register 16bit. Also each slice contains 2 Flip-Flops, 2 multiplexers, arithmetic logic gates and circuit for carry. The total number of CLBs of the specific FPGA is 1,164 and matches the total number of slices is 4,656, while the LTUs are 9,312 and similarly the Flip-Flops.

The IOBs are responsible for communicating the internal logic of the FPGA to external data via terminals. The communication protocol is configurable (TTL, CMOS, PCI, HSTL, SSTL) at an operating voltage of 1.2-3.3V (depending on the protocol and operation). Most IOBs are performing bidirectional mode, except for some, the number of which does not exceed 25% of the total, performing only input. The FPGA used has 232 IOBs organized into four areas (Banks).

The RAM Blocks are performing dual port (input-output) and used for storing data. Each Block has two identical ports, A & B, which have access to the memory independently. Each one can write and read data to the memory, but there is also the possibility that one writes data to and reads the other. Each RAM Block has a capacity of 18,432bits, including the parity bits. This FPGA includes 20 Blocks, a total of 368,640bits.

Multiplier Blocks perform multiplications of unsigned numbers 18bits in two's complement and export unsigned result 36bits also in two's complement. Besides multiplications are able to perform cyclic shift and used as storage areas. They also have additional registers to allow their sequential connection. This structure used in digital signal processing algorithms such as FIR filters. 20 Multiplier Blocks are available in the FPGA used, arranged in two columns along with RAM Blocks.

The DCMs undertake the creation and management of clock pulses to FPGA. To perform this function they have a Delay-Locked Loop (DLL) which is a purely digital control system. The DLL has a clock input from an external source and a feedback system generates clock pulses with frequencies multiples or submultiples of the input signal frequency or phase difference of it. Also maintain the characteristics of clock signals stably and with high accuracy. The FPGA in this application has 4 DCMs.

The development board used for the application was SPARTAN - 3E, which, apart from the FPGA as described above, has a number of electronic devices and auxiliary circuits:

- 64MB DDR SDRAM 100+MHz
- 16MB NOR Flash
- 2x16 LCD Screen
- PS/2, VGA, RS232, Ethernet Ports
- Switches, Buttons, LEDs, Rotary encoder
- 2xSPI 14bit ADC & 4xSPI 12bit DAC
- 50MHz Clock Oscillator
- 40 I/O Pins

The basic design of the waveform generator is shown in block diagram in Figure 7.

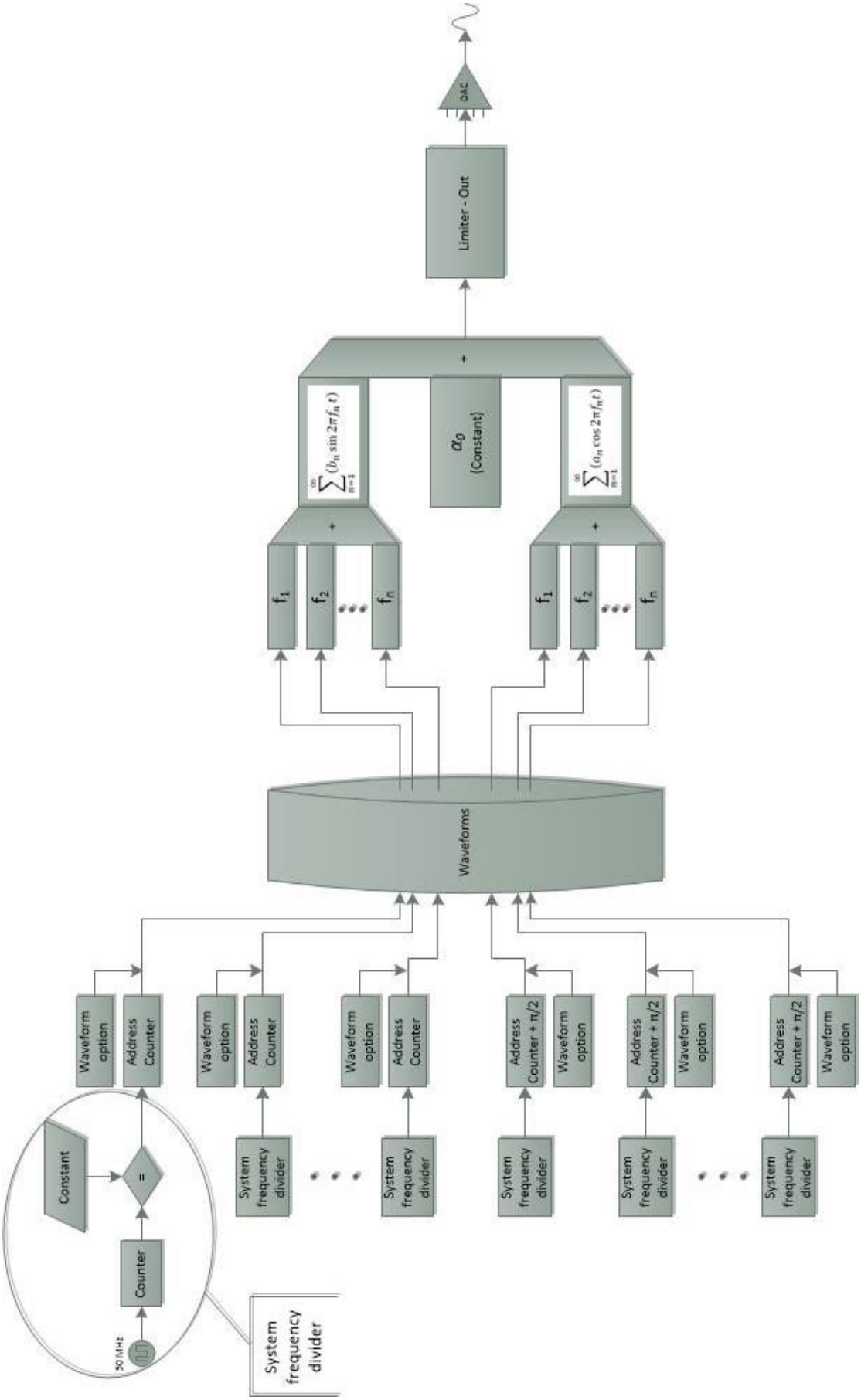

Figure 7. Basic design of the waveform generator.

2.3 DESCRIPTION OF WAVEFORM GENERATOR

The generated waveform is a sum of sines, cosines and constant terms, according to the Fourier series shown at ch. 2.1 (p. 3), sines and cosines are generated in separate generators each, which are sequential circuits, and the constant term is given as arithmetic value and can be customized. The design based on digital summation of terms of the sum. The number of terms is restricted by the size of the registers and the area of the FPGA (the number of CLBs FPGA). The size of the registers, and therefore the terms of the sum, can be changed depending on the needs of the waveform.

2.3.1 Wave composition

In the block diagram of Figure 7 presented the components of the generators. The system clock (50 MHz) clocks a counter (Counter), which compares the current value with a constant (Constant). With this arrangement implemented a frequency divider (System Frequency Divider). DCMs was not been used for the generation of the frequencies used in frequency dividers, because system's architecture needs simultaneously more different frequencies from those that can generate the DCMs. There is opportunity, depending on the hardware, to make use of DCMs to generate the frequency of system clock for groups of generators to increase the flexibility of the waveform generator.

When the counter's value becomes equal to the Constant, increases the wave address counter (Address Counter) by one. The Address Counter gives the address to an array where is stored one period of a sine wave with numerical values, so if it plotted, to form a complete period (Figure 8). The array's size depends on the quality of the waveform needed and affecting the frequencies that the generator is possible to generate. In system's architecture the array size affecting the size of the Address Counters and limited by the needs of the frequencies generated for the application and the free area of the FPGA. Figure 8 shows the graph of a sine wave period as stored in an array of size 1x64 elements of 8bit.

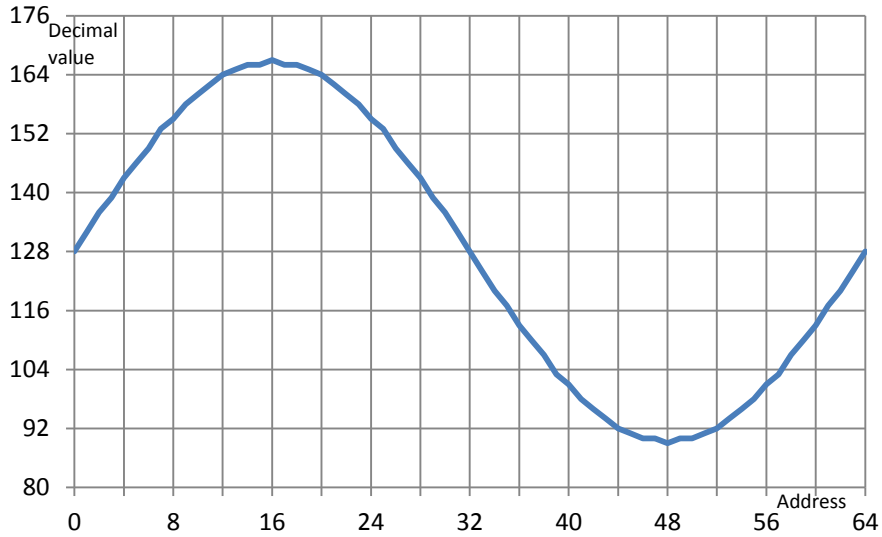

Figure 8. Period of sine wave from array 1x64 byte.

Addresses 0-63 of the array correspond to phase 0-2π wave. The numerical value of each element of the table corresponds to the width of the generated voltage after conversion to an analog signal and may be calculated by the following equation by rounding to the nearest monad:

$$Dv = [V_{DC} + \sin(2\pi \cdot n_m/N_m) \cdot (V_{pp}/2)] \cdot (2^{Nbit} - 1)/V_c$$

Dv: The numerical value of each element of the table

V_{DC}: The amplitude of the DC voltage

V_c: The amplitude of the maximum voltage that DAC can generate

V_{pp}: The voltage amplitude peak to peak of the generated wave

n_m: The number of the address of the element

N_m: The total number of elements in the array (array size)

N_{Bit}: The number of bits of each element

In this example the generated wave has amplitude 1V peak to peak and a DC component with amplitude 1.65V, if used DAC 8bit 0-3.3V. The DC component exists in the design of the waveform generator to avoid the use of negative numbers and negative voltages in the DAC, with reference to simpler design and space savings in the FPGA. In Figures 9a to 9d presented snapshots from the oscilloscope's display of four sinusoidal waveforms generated by the waveform generator using different array size for each.

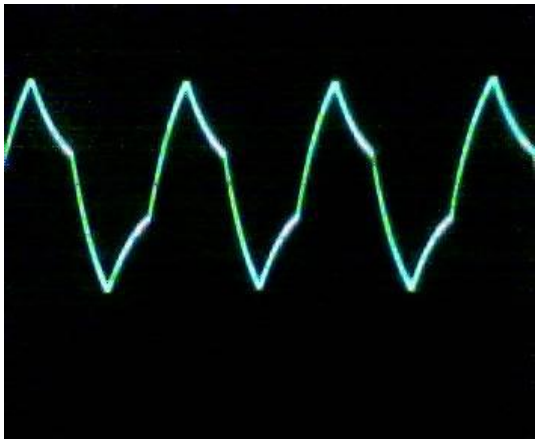

Figure 9a. Array 1x4 Bytes.

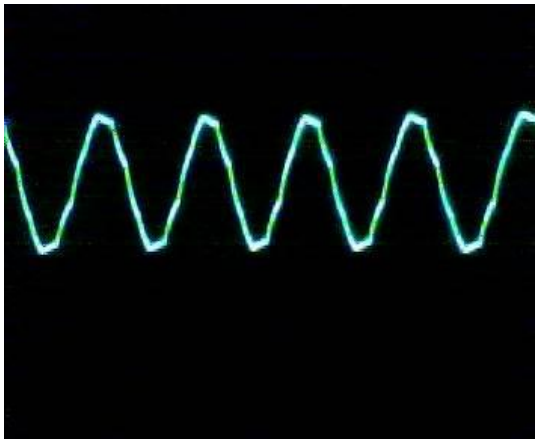

Figure 9b. Array 1x8 Bytes.

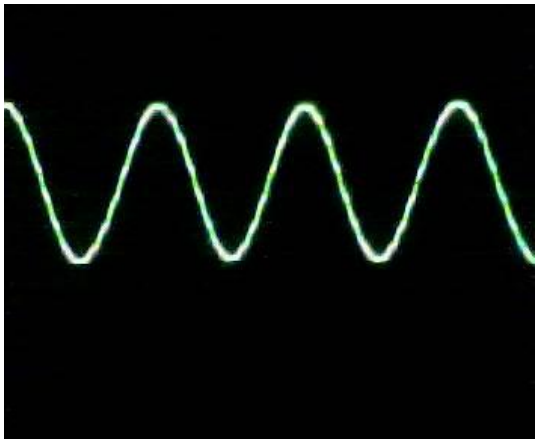

Figure 9c. Array 1x16 Bytes.

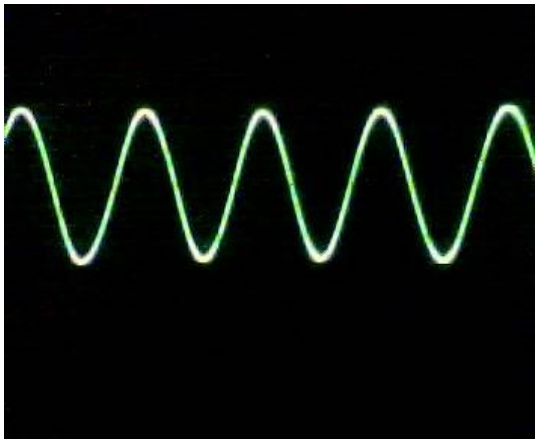

Figure 9d. Array 1x256 Bytes.

To generate the sum is possible to use only one array. In this case the amplitude of the wave for each frequency involved in the aggregation will be formed by repetition of the term. Alternatively various arrays can be made that each one corresponds to a wave with different amplitude. These two approaches are equivalent because they have almost the same complexity in the design, while the space occupied in the FPGA in each case depends on the parameters of the sum. Also is possible to use array that is stored only a quarter of period (0-π/2) since the remaining three quarters are symmetrical in relation to this. The generation of the wave in this case demands additional synthesis steps which increases the complexity. These circuits included in the stage “Waveform Option” of each generator.

The arrays are the system ROM and included in stage “Waveforms”. In system’s architecture selected arrays to be “Logic Vectors” so all generators can have concurrent access to them in any address needed. This would not be possible if the tables were stored in RAM Block.

2.3.2 Create sine – cosine series

As mentioned above, each generator generates its own sine wave independently of the others. For cosine waveforms there are not separate arrays, but the addressing of the array is shifted by a quarter of the size of the array corresponding to π/2. The instantaneous value of each of the wave generator is introduced into the registers f_1 - f_n at the rising edge of the system clock pulse (Figure 10). These registers have the same size with the vectors of the array (8bit in the example).

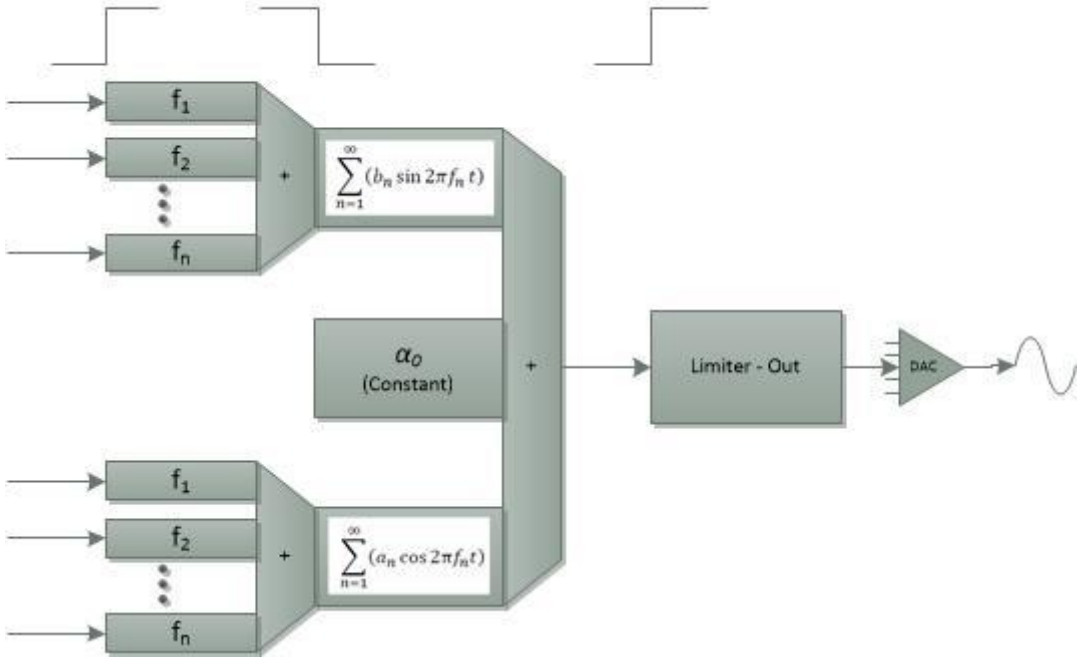

Figure 10. Adders and registers of the waveform generator.

For purposes of orderliness the series of sinus aggregated separately from the series of cosine and each sum imported into a register at the falling edge of the system clock pulse (Figure 10). The size of the register for each aggregation is proportional to the number of terms of the sum, that no carry comes up from the aggregation. If for example the sum of sines (or cosines) contains up to 16 terms of 8bit each, the register for the sum should be 12bit. In aggregation, whereas each wave has a DC voltage and the sum must have a DC voltage too; all proper calculations must be taken care.

The two registers and the constant aggregated and the sum imported in the output register at the rising edge of the system clock pulse (Figure 10). The size of the output

register depends on the constant, that no carry comes up from the aggregation too. In the output register is useful to place a comparator - limiter, so the amplitude of the output can be limited or updated a flag when the amplitude exceeds certain thresholds. In this aggregation also each wave has a DC voltage and the sum must have a DC voltage too and all proper calculations must be taken care. Data input to registers in different phases of clock pulse (as technical pipeline) was preferred in the design to ensure that the changes to all logic circuits that implement the generator would be completed before finalizing the results.

2.3.3 Waveforms

In Figures 11 and 12 presented snapshots from the oscilloscope's display of two different waveforms generated from the waveform generator. The terms of the aggregation of the waveform of Figure 11 were chosen to resemble the theoretical example of Figure 5 (p. 4). For the waveform of Figure 12 the terms were chosen randomly, in order to develop higher complexity.

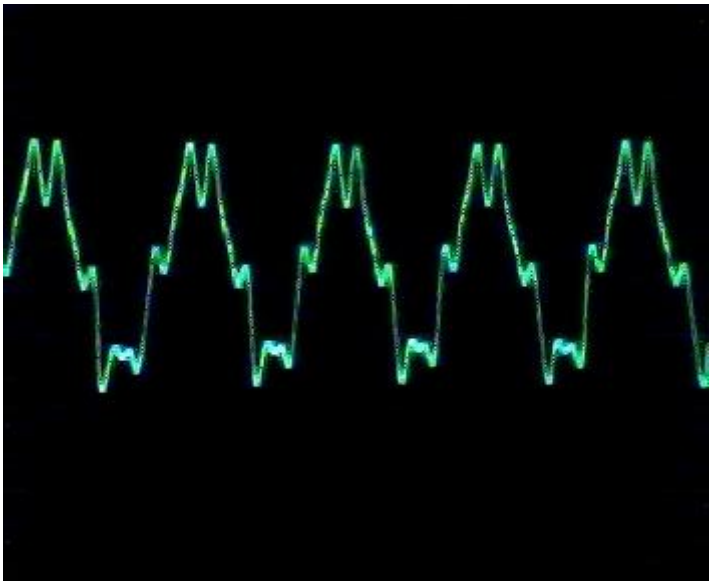

Figure 11. Actual waveform snapshot of sum:

$$g(t) = 10sin2\pi ft + 3cos2\pi 5ft + 2cos2\pi 8ft, (f = 434Hz).$$

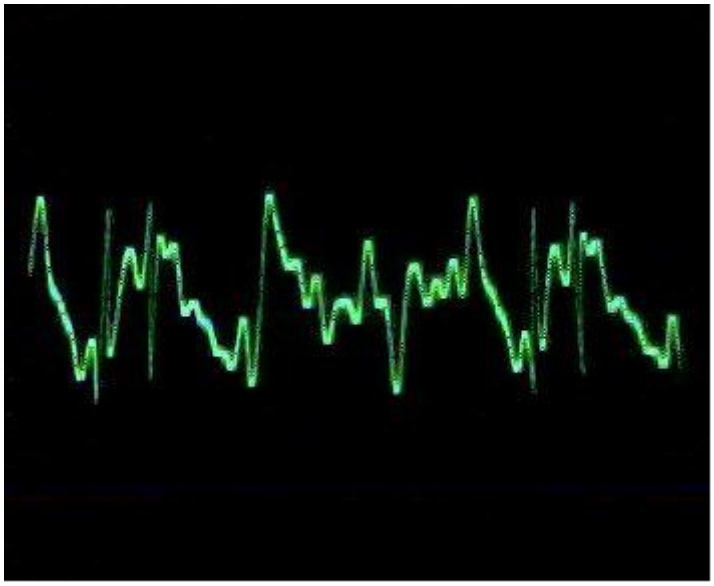

Figure 12. Actual waveform snapshot of sum:

$$g(t) = 10cos2\pi f_1 t + 4cos2\pi 2f_1 t + 2cos2\pi 3f_1 t + 4cos2\pi 4f_1 t + 2cos2\pi 5f_1 t + 4cos2\pi 6f_1 t + 10sin2\pi f_2 t + 5sin2\pi 2f_2 t + 3sin2\pi 3f_2 t + 5sin2\pi 4f_2 t + 3sin2\pi 5f_2 t + 5sin2\pi 6f_2 t, (f_1 = 579Hz,\ \ f_2 = 434Hz).$$

3. MUSIC SIGNAL SYNTHESIZER

3.1 MUSICAL NOTES

The sound is a longitudinal wave, namely the shift of the airwave is in the direction of propagation of the wave. In the air the sound spreads from the vibration and the oscillation of the molecules of air. From a source that oscillates some air molecules generate periodic pressure differences, which spread through the environment, creating sound waves. The human ear responds to these pressure differences.

The notes are sounds at fixed frequencies (fundamental), which when combined properly leave to human ear a pleasant auditory sense. The musical notes are classified in octaves, which are repeated. Two successive octaves include the same notes at twice their fundamental frequencies. In an octave distinguish seven different notes. The nomenclature has prevailed globally assigns the first seven letters of the Latin alphabet in seven basic notes: A, B, C, D, E, F, G. At the same time is used the Latin nomenclature too which is respectively: la, si, do, re, mi, fa, sol. Furthermore, mostly used a number in the name of the note, which indicates in which octave the note belongs. Among some of the seven basic notes exists some more notes, identified as the hash key and designated by adding the # symbol in the name of the basic note. The series of notes finally formed by the following steps: A, A #, B, C, C #, D, D #, E, F, F #, G, G #. The change from one step to another is called semitone, while two successive semitones constitute a tone. The relationship between the fundamental frequencies of two consecutive semitones given by the expression:

$$f_n = f_{n-1} \cdot \sqrt[12]{2}$$

As a characteristic note, the basis of regulating most musical instrument is the A4 with frequency 440Hz. Figure 13 shows the diagram of the notes used in music. The horizontal axis shows the notes and the vertical axis the frequency in logarithmic scale.

Figure 13. Complete scale of the notes used in music.

3.2 TIMBRE – TONE

When a note is produced by any musical instrument, the resulting sound wave contains the fundamental frequency and also multiples of it called harmonics. Each instrument has its own characteristic sound, so we can distinguish it from other instruments and recognize it. This feature is called timbre or tone of the instrument and depends on the harmonics the instrument can produce and the intensity of each shaping in the final sound. So it is possible to determine the frequency spectrum for each instrument and reflected in the chart. Figure 14 shows the frequency spectrum of the violin which distinguished the first ten harmonics. The first harmonic (1) is the fundamental frequency and the remaining harmonics (2-10) is derived first to ninth frequencies (overtones 1st to 9th).

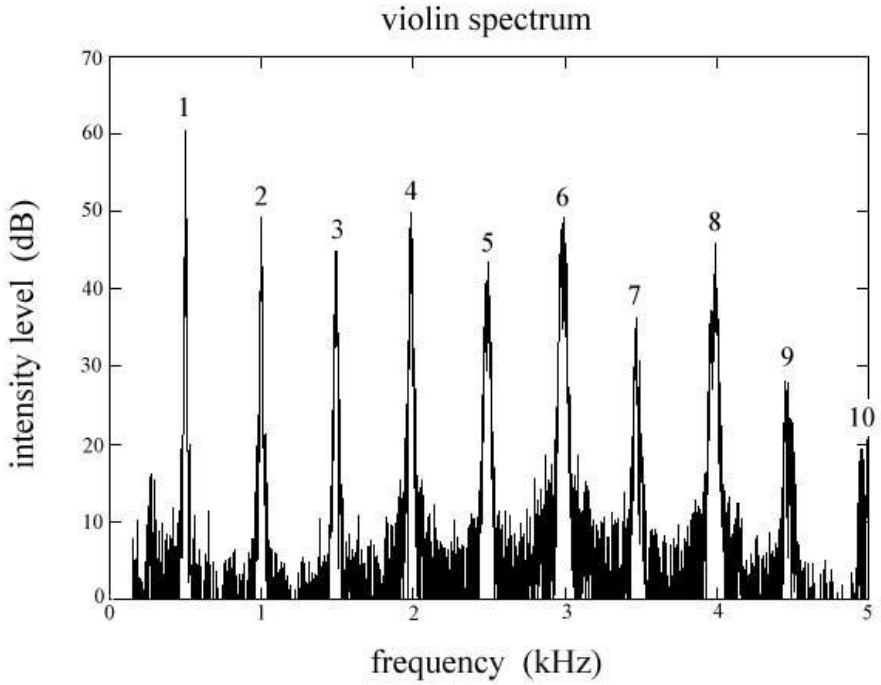

Figure 14. Frequency spectrum of the violin.

Based on the frequency spectrum of a musical instrument each note can represented with a mathematical expression based on Fourier series. Therefore, based on frequency spectrum diagram of Figure 14, the mathematical expression of the note corresponding to the fundamental frequency of 500Hz to a violin is:

$$g(t) = 60sin2\pi ft + 50sin2\pi 2ft + 45sin2\pi 3ft + 50sin2\pi 4ft + 44sin2\pi 5ft + 50sin2\pi 6ft + 36sin2\pi 7ft + 48sin2\pi 8ft + 27sin2\pi 9ft + 20sin2\pi 10ft,$$
$$(f = 500Hz)$$

The above mathematical expression can be converted to electrical signal using the waveform generator described in Chap. 2.3 and then, with suitable devices (amplifier, speaker, etc.), be converted into an acoustic signal. In Figure 15 presented the block diagram of the modified design of the waveform generator in order to meet the requirements of music signal synthesis as a music signal synthesizer.

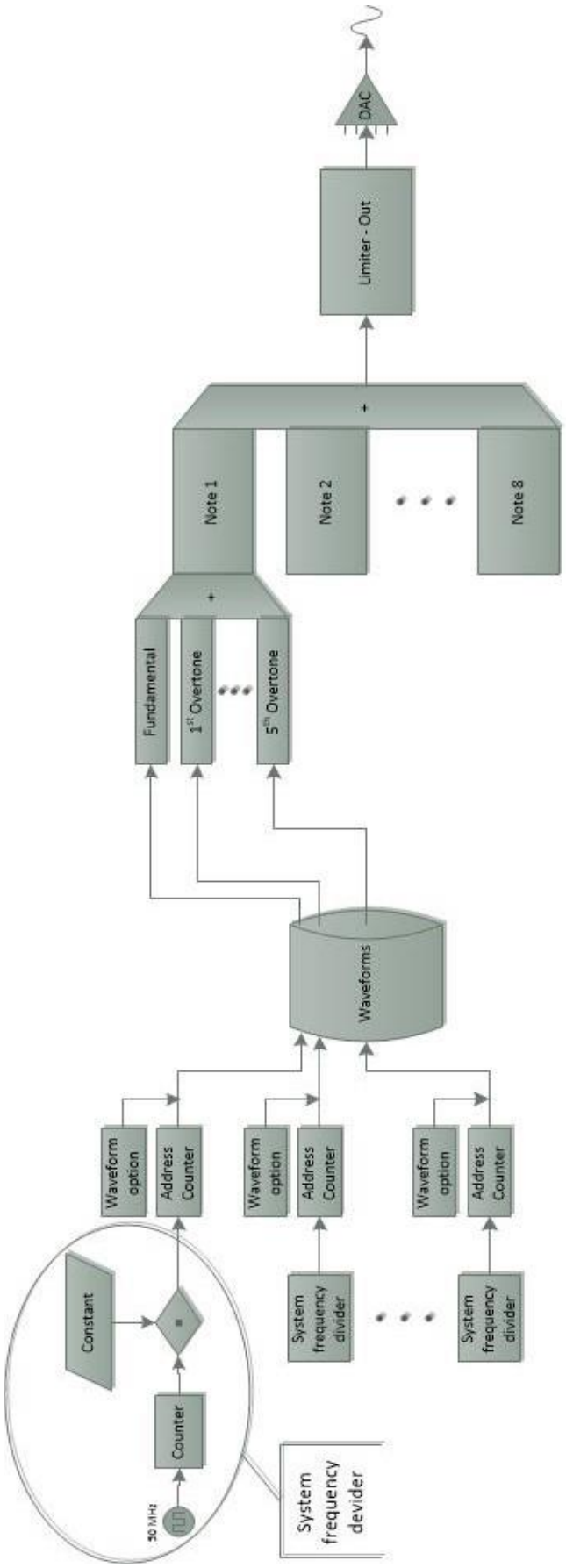

Figure 15. Basic design of the music signal synthesizer.

3.3 DESCRIPTION OF THE SYSTEM

In the system's architecture of the waveform generator made some modifications to meet the new requirements. The most important modification is that the cosine generators were removed. Each note separately synthesized as an aggregation of sine series consists of six harmonics, the fundamental and five overtones. The result is input to the register of the note so that it is possible to be added to the registers of the other notes. The registers that will be added then, and therefore the notes will participate in the final shaping of sound, depend from external parameter, which in this application is a key. So, there is possibility the final sound is coming out from only one note or combination of more notes (chords). Also adjusted a volume dampening mechanism (fade out), in order to produce an audible effect of striking a string.

In stage Waveforms stored 9 arrays 1x64 of 8bit data. Each table can synthesize a sine wave of fixed amplitude from $63mV_{pp}$ to $1V_{pp}$ in steps of 3dB ($V_0 = 1mV$). Figure 16 shows the graph of a period of the sinus waves as stored in arrays in the system ROM.

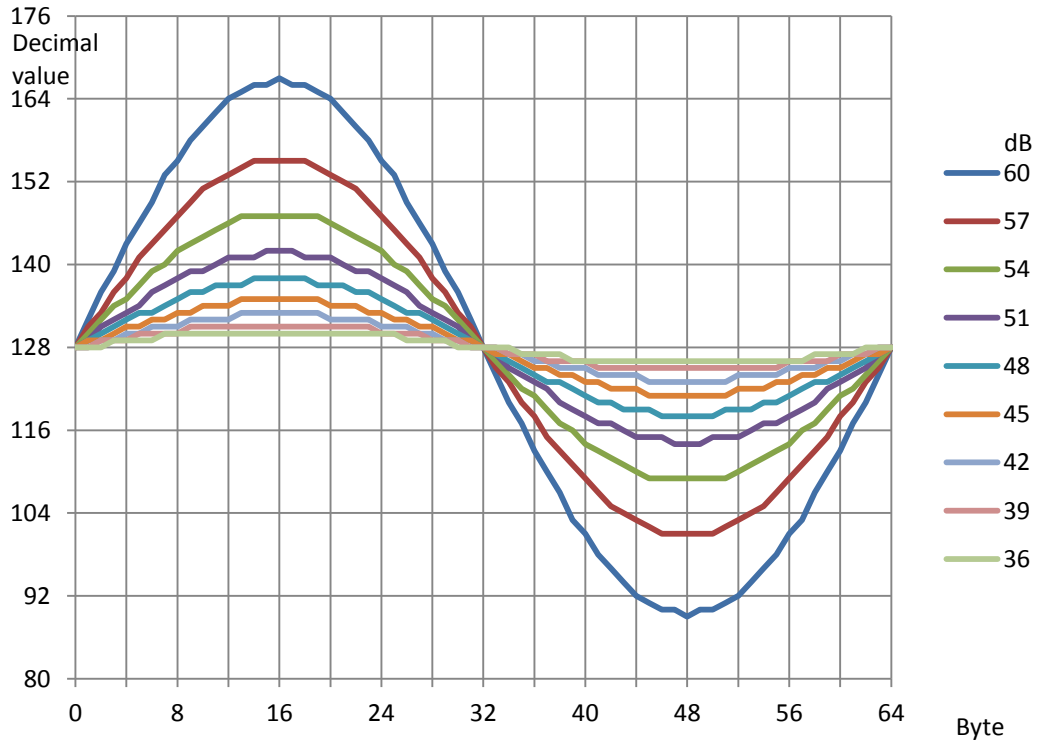

Figure 16. A period of the sinus waves as stored in arrays in the system ROM.

3.4 RESTRICTIONS & TREATMENT

The system's architecture puts some limitations, so in some cases there is deviation of the generated fundamental frequency from the desired and its harmonics, affecting the acoustic effect. These deviations can be eliminated within various ways, which will be described as analyzing the limitations.

The main limitation is the generation of a frequency with high accuracy. The frequency of the system clock is 50MHz and a complete period of the sine wave used in the application is composed of 64 Bytes. Therefore the maximum frequency (f_m) of a sine wave can be produced is:

$$f_m = 50 \cdot 10^6 / 64 = 781{,}250 \text{Hz}$$

Because the frequencies of the generator generated by dividing the clock frequency with a constant which is an integer, the frequencies that can be generated (f_p) are limited to values that gives the result of dividing the maximum frequency with the constant:

$$f_p = 781{,}250 / constant \text{ Hz}$$

For the synthesis of a note used six harmonics (the fundamental frequency and the first five overtones), which are integer multiples of a frequency. Therefore, to generate the overtones, the constants used needs to be numbers resulting from the perfect division of the constant of fundamental frequency by the integers 2, 3, 4, 5 and 6 in order to have frequencies which are integer multiples of the fundamental. So the minimum value that can take the constant of fundamental frequency is the lowest common multiple of numbers 2, 3, 4, 5 and 6, which is 60 and the possible values are integer (n) multiple of 60. This limits even more the frequencies that can be generated for the fundamental frequency (f_f) of a note:

$$f_f = 781.250/60n = 13.020{,}833/n \text{ Hz } (n = \text{integer})$$

In figure 17 presented a diagram showing the deviation of the notes of the DM scale (la major) synthesized by the generator.

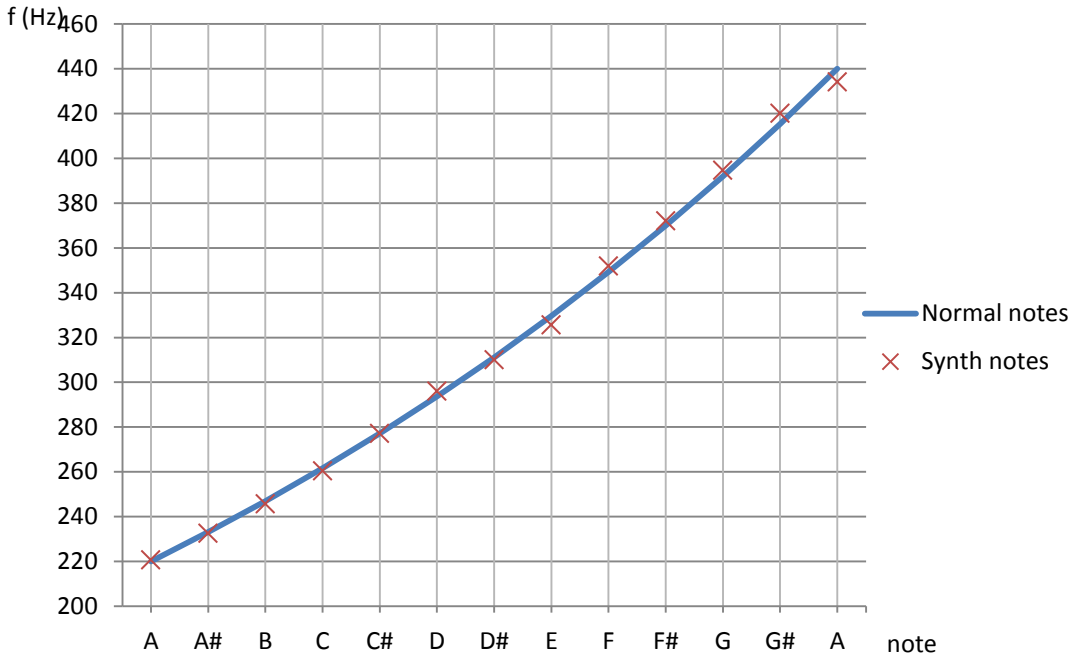

Figure 17. Deviation of the notes synthesized by the generator.

This deviation, although it is difficult to distinguished by the human ear, can be reduced or even eliminated. In the system's architecture of the generator used the system clock (50MHz) for the frequency generation, thus raising the limitations mentioned above. There is opportunity, depending on the hardware, to use DCMs to generate the system frequency for each generator to increase the flexibility of frequency generation.

The deviation becomes larger by increasing the fundamental frequency of a note, that is as we go higher in the musical scale, and becomes sensate. Instruments do not have the same range of frequencies across the range of notes they can play. It happens, even go as high as to limit the amplitude of the higher harmonics. This is due to vibration capability of the instrument by its manufacture, but also constrained by devices work with the instrument. For example, the guitar body vibrates less on high notes, and because the vibration of the body involved in the generation of harmonics, when playing high notes the higher harmonics have significantly lower amplitude, so that they can be considered to not participate at all in produced sound. Furthermore, the electric guitar amplifiers and loudspeakers used in amplifiers have low response at frequencies more than 4kHz. The fundamental frequency of the notes of the last octave on a guitar is greater than 1kHz, thus above the third harmonic practical cannot be heard. This has consequence, when musical instruments playing high notes to change its tone and is relatively difficult to identify them.

There is therefore the possibility for high notes to not use six harmonics, but only four. This forms the least common multiple of 60 to 12, making the number of values that can be achieved for the fundamental frequency larger. Figure 18 shows a diagram of the notes generated with six harmonics (red marks) and with four harmonics for the last three notes (green marks), where we can see the deviation.

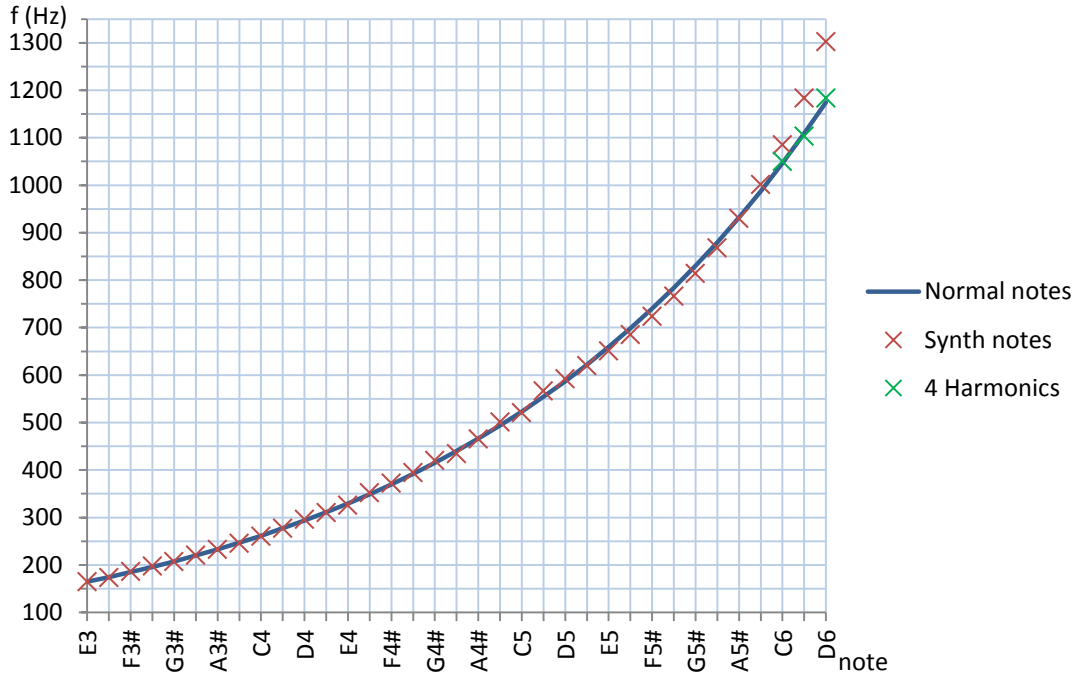

Figure 18. Notes generated with six harmonics and with four harmonics for the last three notes.

As mentioned above, a full period of sine wave used in the application is composed of 64 Bytes. It is possible to use lower resolution, making the appropriate arrays, to generate the waveforms without having such a significant impact to their quality. Figure 9c (p. 11) shows that for resolution 16 Byte and more the quality of sinus waveform is satisfactory. Based on calculations of ch. 3.4 (p. 21) we can conclude that for each halving of the resolution doubles the capability of the system to produce frequencies.

The system's architecture also allows for the same note different harmonics to have different resolution. In figure 16 (p. 20) we can observe that as the amplitude of the sine wave reduced, less information about its composition exists. It is possible then, for the low amplitude sine waves to use a lower resolution, since, anyway, there is lost information in high resolutions. And this technique can improve the resolution of the system for generating

frequencies, increases, however, the complexity and therefore the space occupied by the design in FPGA.

3.5 COMPOSITION & PLAYING GUITAR SOUND

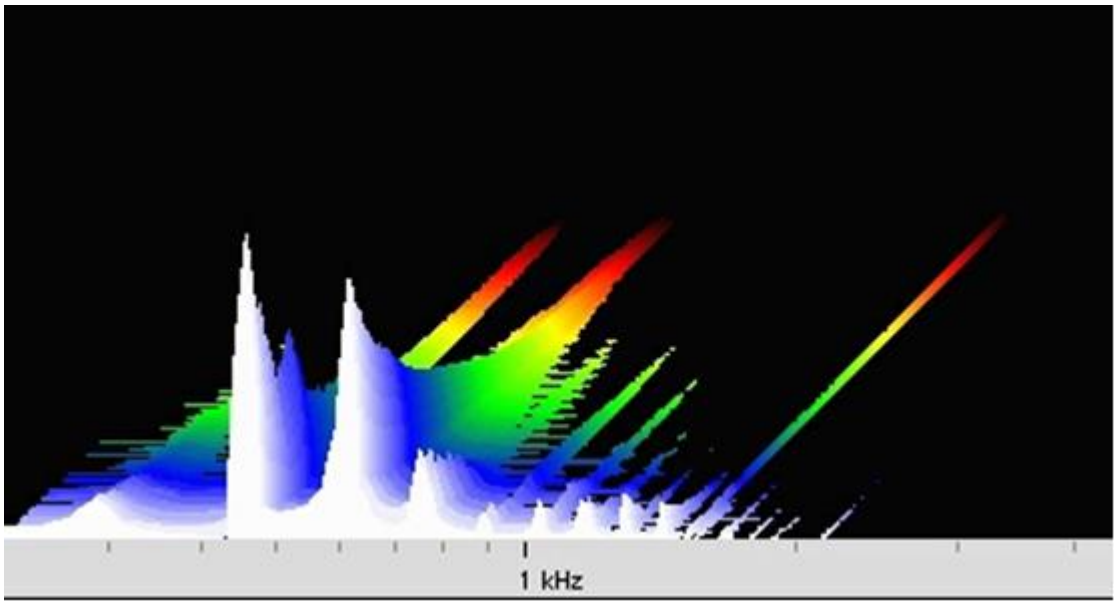

Figure 19. Guitar spectrum diagram.

To play guitar sounds designed volume dampening mechanism (fade out), as mentioned in ch. 3.3 (p. 20), which is reducing the volume as synthesized notes on generators. To implement this operation the amplitude of the sine wave of each harmonic decreases, based on some function of time, by selecting different array in specific time points.

The function of time that reducing of volume takes place derived experimentally from the acoustic effect in combination with the image of the chart. Actually there is no specific rate, because each type of guitar has its own characteristics by manufacture and thus there are sensate differences in sound dampening or in tone from an instrument to another. With appropriate changes in the parameters of the waveform generator is possible to achieve appropriate sounds to cover these differences, or even to present new features. Figure 20 shows a diagram of the reduction of the amplitude of the fundamental frequency of a note in relation to the time played.

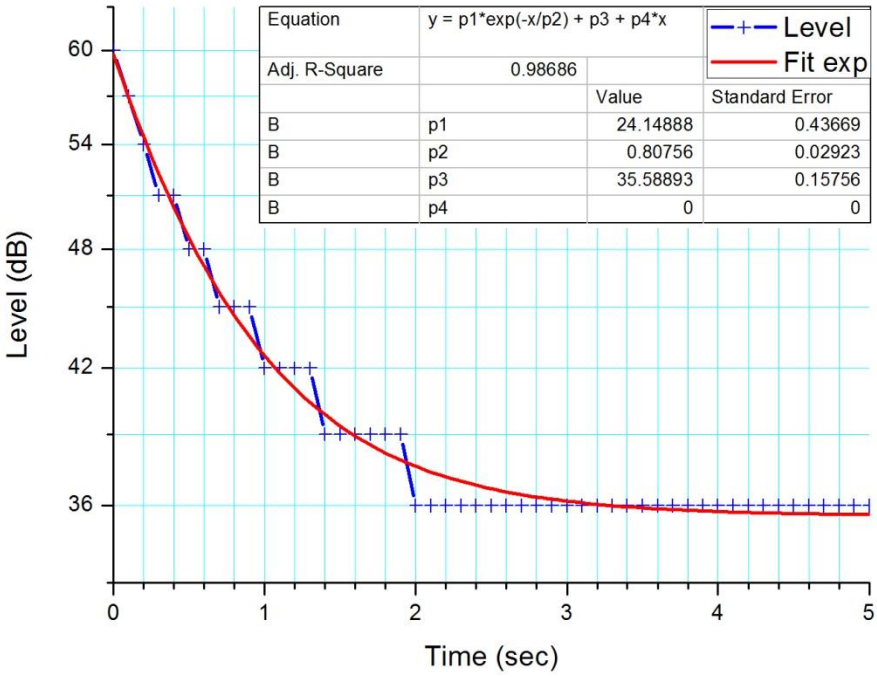

Figure 20. Reduce of the amplitude of the fundamental frequency of a note in relation to the time of reproduction.

It is also possible to incorporate in system's architecture digital filters to simulate various effects used by musicians for electric guitar. In this application modified the limiter output stage to behave like Overdrive.

The Overdrive is an effect used by guitar players very often. The operation of this effect based on overdriving the preamp stage, in order to go to saturation. The result is to cut the tops of the sound's waveform and becomes distorted. Figure 21 shows an actual snapshot of guitar audio waveform from the generator after the implementation of the Digital Overdrive.

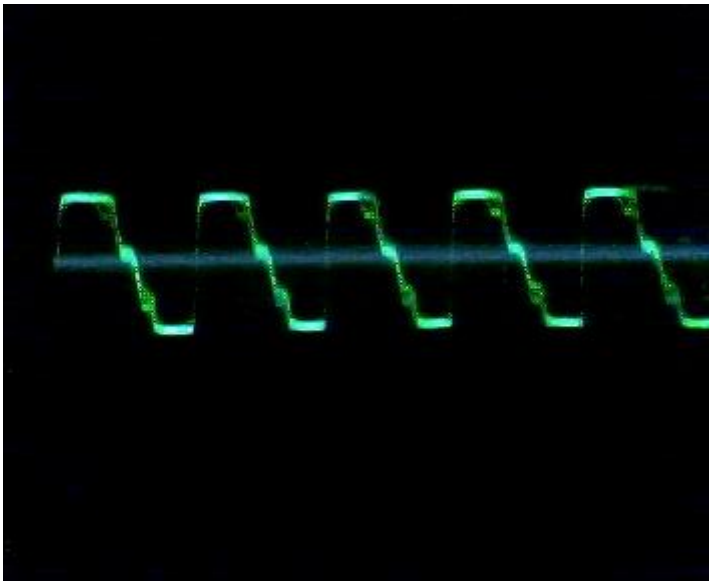

Figure 21. Actual snapshot waveform guitar sound from the generator after the implementation of the Digital Overdrive.

3.6 COMPOSITION & PLAYING VIOLIN SOUND

The sound of the violin has a constant intensity, which depends on the pressure exerted by the musician in the chord with the bow as it drags on. It is also possible to modify slightly the tone of the sound, depending on how the bow fits to the string. These features can be simulated to produce the music signal synthesizer's sound similar to the sound of the violin. Figure 22 shows the frequency spectrum diagram of violin, where with red color are the harmonics that compose the actual sound of violin and with blue color the harmonics used in the generator.

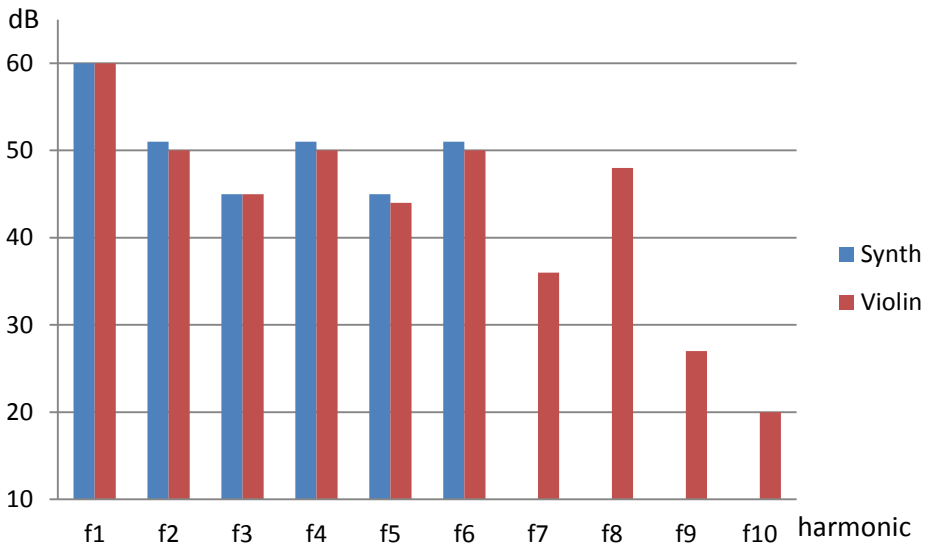

Figure 22. Violin spectrum diagram.

3.7 COMPOSITION & PLAYING SAXOPHONE SOUND

The sound of a saxophone has a constant intensity which depends on the force that the musician blows to the mouthpiece of the instrument. It is also possible to modify greatly the tone of the sound too, depending on how the musician blows or depending on the material of the mouthpiece. For this reason there is neither a specific range of frequencies nor a specific tone for the saxophone, but both depend on the musician and is the trade mark of each one. These features can be simulated to produce the music signal generator's sound similar to the sound of the saxophone in order to cover the differences or even to present new features. Figure 23 shows the frequency spectrum diagram of saxophone which is more common, where with red color are the harmonics that compose the actual sound of saxophone and with blue color the harmonics used in the generator.

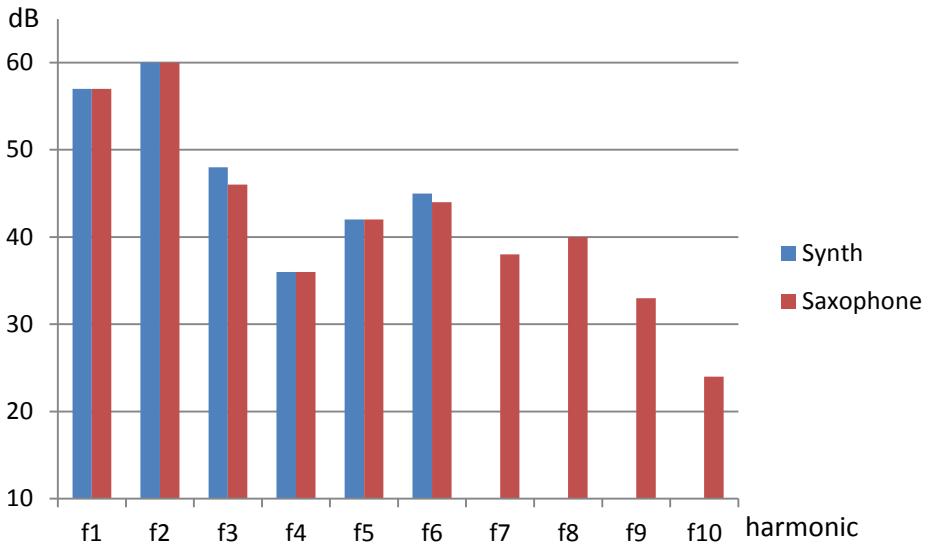

Figure 23. Saxophone spectrum diagram.

3.8 IMPLEMENTATION COMPLETION

The following are actual screenshots from the oscilloscope to display four waveforms corresponding to sounds from different musical instruments as produced from the waveform generator.

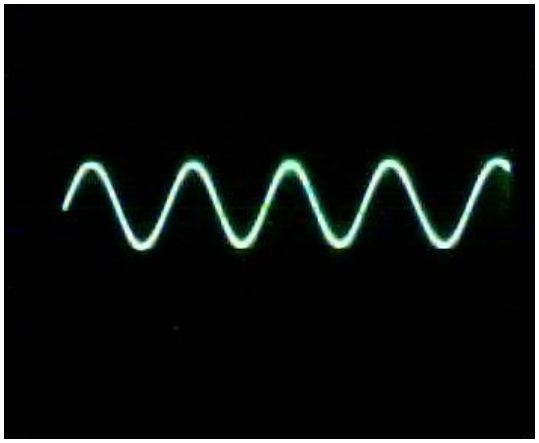

Figure 24a. Hammond (8 0000 0000).

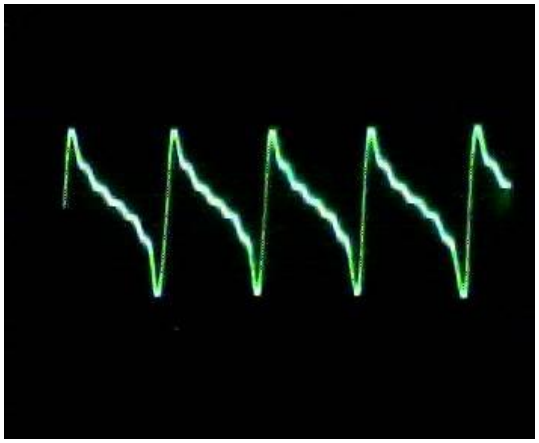

Figure 24b. Guitar.

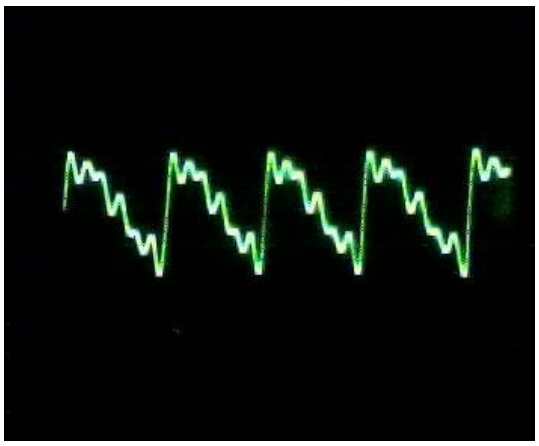

Figure 24c. Violin.

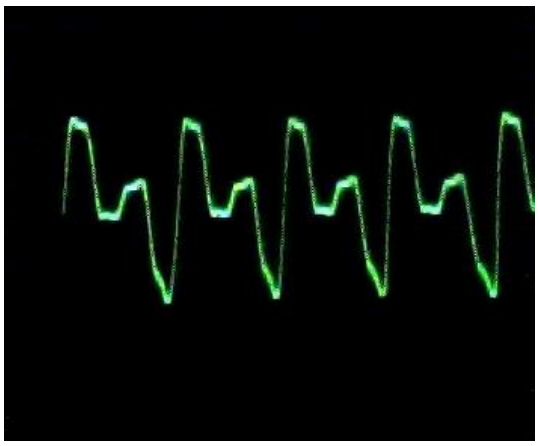

Figure 24d. Saxophone.

To convert the digital signal to analog used a simple R2R Ladder, the theoretical circuit of which presented in Figure 25. This method causes small variations in the accuracy of conversion and has high output impedance. These variations are not important for the application in the developed stages. The oscilloscope used for the impression and the device used for the amplification of the sound and playback have also high input impedance both. For these reasons it was not necessary to use DAC with better characteristics.

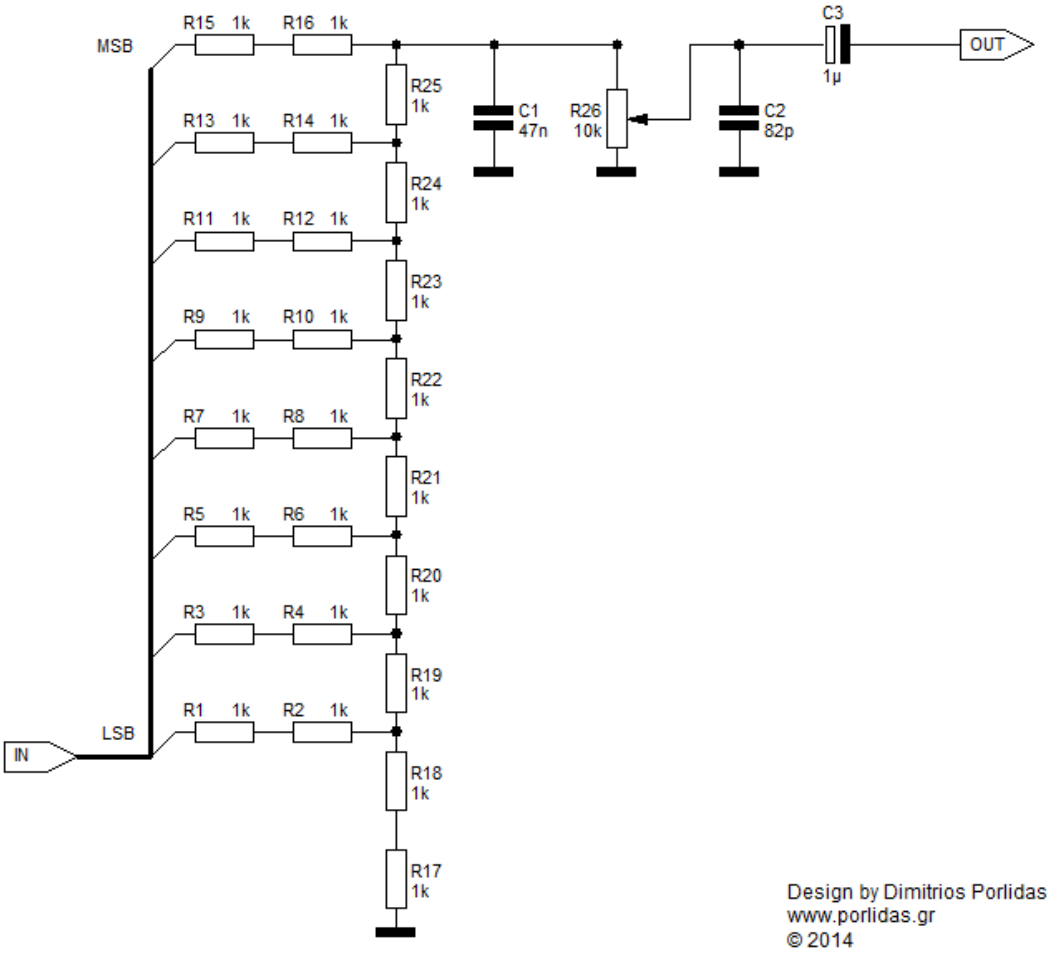

Figure 25. R2R Ladder to convert digital signal to analog (DAC).

Figure 26 shows the statistics of the usage of FPGA to implement the waveform generator and the music signal synthesizer. The implementation of the music signal synthesizer has significantly high demands, having occupied almost the entire FPGA. The application that exported the reports includes: eight notes with corresponding keys, four different instruments (one of which is the guitar with the sound damping system) with choice of two switches and four different overdrive settings with choice of two switches too. Figure 27 shows a diagram of the entire audible range, and musical instruments to those areas that can reproduce sounds.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	336	9,312	3%	
Number of 4 input LUTs	350	9,312	3%	
Number of occupied Slices	285	4,656	6%	
Number of Slices containing only related logic	285	285	100%	
Number of Slices containing unrelated logic	0	285	0%	
Total Number of 4 input LUTs	508	9,312	5%	
Number used as logic	350			
Number used as a route-thru	158			
Number of bonded IOBs	76	232	32%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	3.32			

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	1,504	9,312	16%	
Number of 4 input LUTs	7,794	9,312	83%	
Number of occupied Slices	4,636	4,656	99%	
Number of Slices containing only related logic	4,636	4,636	100%	
Number of Slices containing unrelated logic	0	4,636	0%	
Total Number of 4 input LUTs	8,796	9,312	94%	
Number used as logic	7,794			
Number used as a route-thru	1,002			
Number of bonded IOBs	149	232	64%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	4.61			

Figure 26. Statistics of the usage of FPGA to implement the waveform generator and the music signal synthesizer

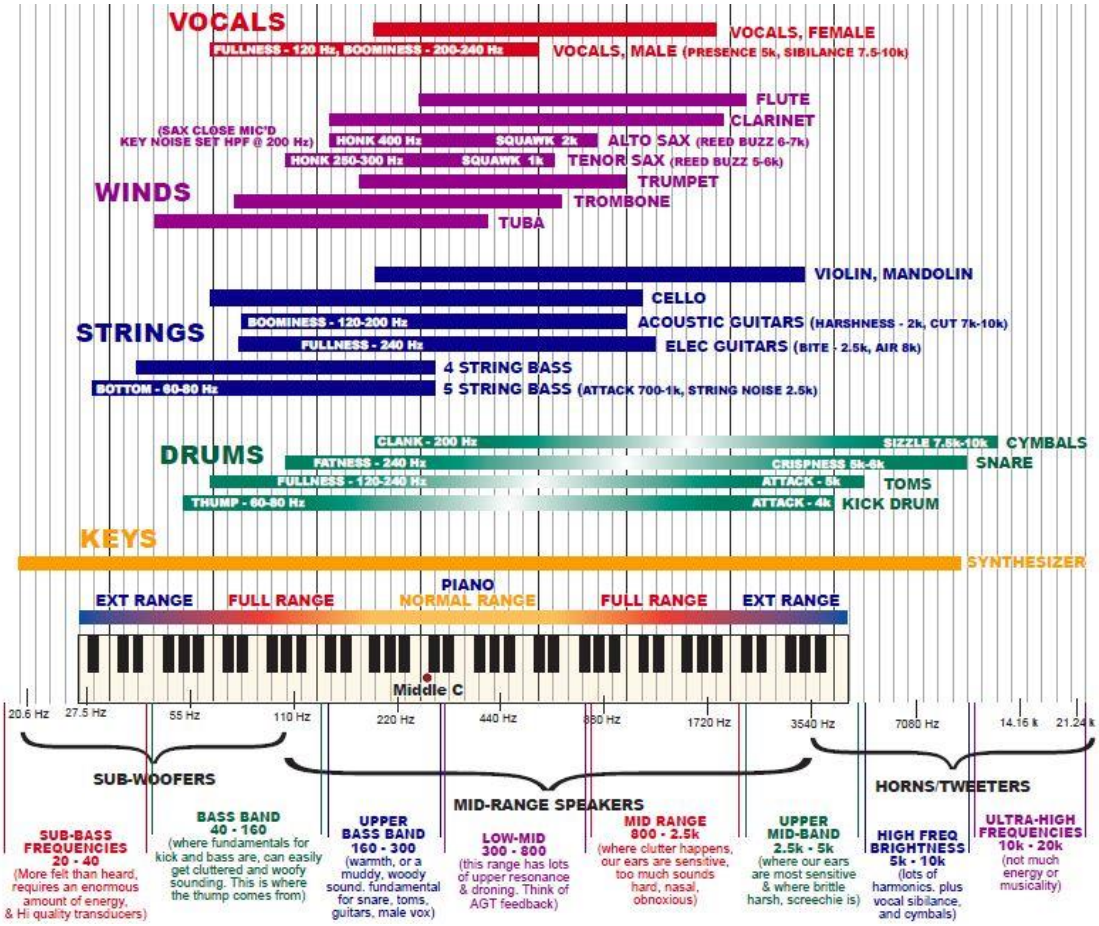

Figure 27. Diagram of the entire audible range.

4. SOURCES

- Kendall Milar (FAST FOURIER TRANSFORM ANALYSIS OF OBOES, OBOE REEDS AND OBOISTS: WHAT MATTERS MOST TO TIMBRE?)
- XILINX
- http://www.feilding.net/
- http://www.phy.mtu.edu/
- http://home.cc.umanitoba.ca/
- http://computermusicresource.com/
- http://www.philbarone.com/blog/saxophone-news/