Distortion effect for electric guitar

Distortion Effect For Electric Guitar Using FPGA

Introduction

Project Goals And Objectives

The goal of the project is to implement distortion effects for electric guitar on an FPGA board. The algorithm that is going to be used is The Extended Karplus Strong Algorithm (Jaffe & Smith, 1983). The analog audio signal from the electric guitar is captured by the analog to digital converter (ADC) module of the board. The FPGA is going to send the digital audio signal to a speaker to be played.

The algorithm is going to be implemented on FPGA instead of using ASIC

design approach. The pros and cons of FPGA design and ASIC design are

discussed on the Xilinx website. The design advantage comparison of FPGA and

ASICand the design flow comparison of FPGA and ASIC (Xilinx Corporation, 2009).

ASIC design has more steps to complete as can be seen . Also, it is suitable for very high volume designs. For a single unit, using FPGA is a better solution. FPGA has no upfront non recurring expenses. It is faster to implement. Manufacturing of ASIC design chips take long time. However, a design can be downloaded to the FPGA and programmed very fast. Considering all these, using FPGA design is more suitable for this project.

Project Deliverables

The deliverables include the Verilog HDL code of the design. It is going to be synthesizable and can be used with suitable FPGA boards. The final project report is going to be delivered. It is going to include the details of the hardware algorithm, the design process and the results obtained from the functional verification and the hardware validation of the system. A demonstration of the project is doing to be done with the developed prototype of the system. The electric guitar is going to be the input of the system. The output from the board is going to be played through speakers.

Technology Trends

Before the invention of FPGA's, CPLD's (Complex Programmable Logic Device) were the most complex programmable logic devices. And before CPLD's, PAL's (Programmable Array Logic) were used frequently.

PAL's were introduced in March 1978 by Monolithic Memories, Inc. They are

only one time programmable. PAL's are consisted of PROM's (programmable

read-only memory). They were mostly used in minicomputers. These devices have

fixed OR and programmable AND arrays. This enables the implementation of sum

of products logic. A simplified programmable logic device. Typically, PAL devices have a few hundred gates.

CPLD devices have higher complexities compared to PAL devices. They have similar features to both PAL devices and FPGA's. Like PAL's, they don't have external ROM’s, which enable the CPLD’s to start functioning just after startup. They have much higher number of gates compared to PAL devices. They have around thousands to tens of thousands of gates. However, this is low compared to FPGA’s, since the number of gates inside the FPGA’s can go up to a few millions.

FPGA’s have the most number of gates and flip-flops compared to the others. They are more flexible but their design is more complex.

The first distortion effect for electric guitar wasn’t produced on purpose. It was mostly caused because of damaged guitar amplifiers. One example was a recording by Johnny Brunette Trio, which caused a fuzz tone effect. (The Train Kept Rollin’, 2009). Electronic based distortion and overdrive effects came to scene in 1960’s and 1970’s. The effects were achieved by diodes, transistors and amplifiers and most of these pedals were analog. With the improvement in the digital signal processing techniques, digital processors became an important part of the technology in the last decade.

Market Research

The digital products in the market nowadays feature more adjustable effects than just a distortion effect. Typically, they have parallel effect modules that can run simultaneously. They also have advanced software. They have preset tones and effect libraries, tuners and even more features. Also, most of them have USB interfaces with a PC or MAC for compatible recording software. So, the project’s features aren’t going to be able to match the products’ features in the market.

Boss, Line 6, Zoom, Korg, Digitech are among the major companies which produce digital guitar effects processors.

The bestselling multieffect electric guitar processors on Amazon. com. It can be observed that Zoom and Digitech have the most market.

Requirements

Functional Requirements

The electric guitar will be connected to the FPGA board’s analog to digital converter input. The analog to digital converter is going to convert the incoming analog signal to an 8-bit digital signal. The sampling frequency is going to be 44100 Hz, which is the standard for most of the digital audio files. The reason for choosing this sampling frequency is the human ear’s ability. The human ear cannot perceive frequencies above 20 KHz. According to the Nyquist Sampling Theorem, a signal can be exactly reconstructed from its samples if the sampling frequency is greater than twice the highest frequency of the signal. If the highest frequency that the human ear can perceive is considered to be 20 KHz, anything above 40 KHz is going to be enough for sampling frequency (Schulzrinne, 2008). The signal is going to be processed inside the FPGA using The Extended Karplus Strong Algorithm (Jaffe & Smith, 1983). The processing should be fast enough so that the human ear cannot understand the delay between the time when the player hits a note on the guitar and the time that the output is played by the speakers. After the processing, the 8-bit signal is going to be converted to analog. Finally, this analog signal is going to be sent to the speakers and played.

The hardware functionality that the system is going to provide.

Nonfunctional Requirements

The most of important constraint on the system will be the time constraint. The delay between the input and output audio signals must be minimized. This requires the design to be fast. For this purpose, the resources available on the FPGA should be used efficiently.

The most of important constraints on the timing of the design is going to occur due to the algorithm. Floating-point arithmetic might be needed to use according to the algorithm. This might cause the calculations to take longer.

Also, another constraint on the system is the speed of the FPGA. The speed of the FPGA is not going to cause a problem for sampling the incoming analog audio signal. However, the speed of the FPGA is going to put a constraint on the speed of the algorithm. A pipelined algorithm might be used in order to satisfy the requirements for the speed and the timing of the system. There are going to be feedback loops, filters and saturator blocks in the system. So, a pipelined algorithm is going to increase the utilization of these blocks and this is going to result in the increase in the throughput.

If there is a pipelined algorithm, more resources are going to be needed to implement the pipelined system. The limited amount of the resources such as memory blocks and arithmetic units might put a constraint on the design.

Also, another constraint is going to be the data width of the ADC and DAC. Due to the limited number of bits on ADC and DAC, the quality of the digital audio signal is going to be limited.

Product Requirements Analysis

The product requirement analysis is done using Quality Function Deployment (QFD) technique.

The most important criteria for customer satisfaction are low delay time and distortion effect level. Also, good sound quality is very important too. Implementation of additional effects is the least important feature of the product. Low power consumption, low cost, effect adjustability, good bass and treble sounds, good feedback are also expected to have good standards by the customer.

In order to meet the customer expectations, most important step is choosing the distortion effect algorithm correctly. The use of external resources should be kept to minimum level in order to meet the speed requirements of the system. Any use of external memory is going to cause additional memory access time and cause the system to function slower. This is going to result in an unwanted delay time.

Bit resolution is also important. It is going to affect the sound quality. The higher number of bits is going to increase the quality. It might also help us get rid of using floating point arithmetic for implementing the saturation algorithm. However, the higher number of bits might cause a problem with the pipeline implementation.

Project Requirements

FPGA has to capture the analog sound and this signal is going to come from the output of an electric guitar.

The FPGA board that is going to be used is chosen to be Spartan-3A Starter Kit board because of its built in analog to digital converter and digital to analog converter modules. The board also has a stereo mini jack for audio. These features make this board very suitable for audio processing and thus, very suitable for this project. Also, the FPGA chip has 700 K gates (Xilinx Corporation, 2009).

In order to play the output, stereo speakers are going to be connected to the board pin to which the output signal is connected.

The design is going to be done in register transfer level (RTL). The RTL design of the system is going to be described using Verilog HDL. In order to do this, Xilinx's design tool Xilinx ISE Webpack 11. 3, which is free a program, is going to be used. Before prototyping the system, functional verification has to be completed successfully. For this purpose, Modelsim, which is develop by Mentor Graphics, is going to be used.

Before starting the hardware design of the system, the algorithm is going to be simulated and verified using functional blocks in MATLAB Simulink.

The hardware requirements for the system are and the software requirements for the system.

Design

Architecture

As discussed earlier, the algorithm that is going to be used is The Extended Karplus Strong Algorithm (Jaffe & Smith, 1983). The algorithm extensively uses filters. The algorithm is modeled and simulated under MATLAB Simulink. The model consists of functional blocks. The filters are defined by their discrete transfer functions. There is also a feedback loop. The sound is amplified by a gain block and passes through a saturation block. The saturation block basically causes the signal to saturate if its amplitude goes over or below specific thresholds. So, the higher the signal is amplified by the gain block, the more the signal is going to get distorted; since it is going to be saturated from lower amplitude compared to its new peak value. The model of The Karplus Strong Algorithm.

Since there are consequent filter blocks, the signal is going to be delayed. To overcome the problem, the level of parallelism should be increased. Since there are 20 block RAM’s in the FPGA, these can be used for increasing the pipeline depth and the level of parallelism. When an 8-bit sample passes though the first filter, it is going to go to the second one. Instead of waiting and doing the second operation using the same hardware, we should maximize the use of the resources and send the data that passed through the first filter to another resource. During that time, the other sample can pass through the first filter. Usage of block RAM’s might be very beneficial here, in order to increase the throughput and the speed of the system.

Since the data that is going to be processed isn’t going to be large, only the internal block RAM’s might be enough. Also, use of an external RAM is going to put more delay on the line because of the longer memory access time. This is highly undesirable since the most important criterion for the system is its speed.

Structure

The system consists of four main parts. First part is where the user interacts with the system. The user is going to generate an output from the guitar and that output is going to be captured by the FPGA board. FPGA is going to the process the output and pass it to the third part of the system, speakers. The stereo output is going to be played by the speakers. Also, a PC is needed to send the . bit file to program the FPGA.

The FPGA board. It has an audio output port on the right top. If needed, DDR2 SDRAM can be used as external memory. The analog digital circuitry is used for capturing the analog signal to the board. The circuitry has 2-channel 14-bit analog to digital converter and 4-channel 12-bit digital to analog converter. The switches can be used for turning the distortion on and off. Also, they can be used for the same purpose if additional sound effects are added to the system. Rotary knob can be used for adjusting the level of the distortion or the gain or the volume. The quantity that is going to be adjusted can be determined by the switches since there is only one rotary knob.

Interface

There are three interfaces in the system. The first interface is for programming the FPGA. The connection between the FPGA and the computer is going to be achieved with USB 2. 0. Xilinx iMPACT tool is going to be used to program the FPGA.

The second interface is for capturing the analog audio signal from the electric guitar to the FPGA board. The on board analog to digital converter is going to be used for that purpose. Analog to digital converter unit on the board.

The third interface is going to be between the FPGA and the speaker. The digital signal is going to be converted to analog signal using Xilinx's digital to analog converter module and it is going to be sent to the audio jack port of the board. The stereo audio jack module.

Implementation

Implementation Scope

As discussed in Section 3. 2, the system consists of four main parts. The module for sending the . bit file from the PC to the FPGA is already given with Xilix iMPACT tool, so no implementation is required for this.

The second module is the audio input to the board. This is the input module. The input module is going to be implemented with the help of on board analog to digital converter. As discussed earlier, the sampling rate and the bit resolution are the most important parts of the input module. The sampling rate is going to be 44100 Hz and the resolution is planned to be 8 bits. The captured analog signal is going to be converted to digital signal and sent to FPGA module for processing.

FPGA module is going to be responsible for processing the digital signal. For faster and efficient processing, pipelined implementation is going to be done. This is going to be done using RTL description of the hardware with Verilog HDL.

The output module is going to convert the processed digital signal to analog and send it to the board's audio jack port for playing the processed signal using speakers. Xilinx's DAC module is going to be used for the implementation of this module.

Implementation Coverage

The algorithm that is going to be used for implementation is The Extended Karplus Strong Algorithm (Jaffe & Smith, 1983). The block diagram of The Extended Karplus Strong Algorithm.

The output is going to be sent to gain and saturation blocks. There are filter blocks and delay blocks in the system. These functions are going to be implemented inside the FPGA. The first functional block is a pick-direction low pass filter (Smith III, Pick-Direction Lowpass Filter, 2009). The second functional block before the feedback loop is a pick-position comb filter (Smith III, Pick-Position Comb Filter, 2009). In the feedback loop, there is a delay block on the top. The other blocks are again filters. After the delay block, the signal goes through a two-zero string damping filter (Smith III, Two-Zero String Damping Filter, 2009). Before the addition operation in the feedback loop, another pick-direction low pass filter is going to be used. After the loop, there is going to be dynamic level low pass filter (Smith III, Dynamic Level Lowpass Filter, 2009).

After these filters and delays, there is going to be a gain block which is used for increasing the level of distortion. Distortion effect is going to be generated by a saturation block. The saturation can use either hard clipping or soft clipping. Soft clipping has higher complexity. It is a third order polynomial. It results in a smoother sound. However, for more distorted and fuzzy sound, hard clipping is preferred. Since it has a heavier sound and is easier to implement, hard clipping is going to be used. The input-output relations of hard clipping and soft clipping

Develop Or Adopt Decision

The most important part for the project is the FPGA board. It is going to be adopted. If I wanted to design the circuit with a PCB design tool in which I am not experienced, I would have paid a lot of money to get it manufactured. And the design has to be perfect before getting the chip produced. The decision of choosing whether to use FPGA design or ASIC design was discussed earlier in Section. So, buying and using an FPGA board is the best option here. Spartan 3A Starter Kit is going to be used for the project.

For the output interface of the design, Xilinx has a module described in Verilog and is available for free. For DAC and output purposes, that module is going to be used.

If there is an available module for the input port of the system for free from Xilinx, it is going to be adopted. Otherwise, the ADC module is going to be developed according to the ADC hardware available on the FPGA board.

The design on the FPGA is going to be based on an algorithm but it is going to be designed by me.

Also, an electric guitar and speakers with amplifiers are needed for the project. They were already available before the start of the project.

For software, Xilinx ISE, Xilinx iMPACT, Modelsim XE and MATLAB are going to be used. MATLAB is already available and the others have free versions for students.

Implementation Process

Three modules are going to be implemented. Each module can be implemented independently from each other. Finally, all the modules are going to be connected under a top module. DSP module is the main part of the design where the algorithm is going to be implemented. The functional verification of the design is going to be independent from the other modules.

Implementation Resources

The resources for implementation can be grouped into two. First, we need hardware resources. The second group is the software resources.

The most important resource for hardware is the FPGA development board. Spartan 3A Starter Kit is going to be used. This specific board is chosen due to some reasons. This board is suitable for DSP applications. It has ADC and DAC modules. It also has a stereo audio jack for outputting the processed signal. So, this board is going to be used for implementation.

FPGA is going to be programmed from a PC. The hardware of the system is going to be described using Xilinx ISE tool, which requires a PC. So, we also need a PC for implementation. The connection of the board with the PC required a USB cable, which is provided with the board.

We also need an electric guitar and speakers. The required hardware resources for implementation.

Besides the hardware resources, some software resources are going to be needed too. First, before starting writing the code for the hardware, the algorithm is going to be tested and the functional blocks are going to be made clear using MATLAB Simulink software.

For synthesis and implementation, Xilinx ISE is going to be used. It is going to synthesize and implement the hardware described by Verilog HDL. It also includes Xilinx iMPACT tool which is used for sending the . bit file to the FPGA for programming.

For functional verification, Xilinx Edition of Modelsim, developed by Mentor Graphics is going to be used.

Implementation Activities

The project group consists of only one person. So design, verification, implementation and testing are going to be done by me.

During the project, additional training and study is going to be required in digital signal processing and filters. Also, digital filter design should also be studied. Another thing that needs improvement is writing testbench to verify the designed system.

Testing

Testing Scope

The testing of the system consists of two parts. There is a functional verification part and a hardware validation part. For functional verification, Modelsim XE software is going to be used with Verilog HDL.

The parts that are going to be tested are the input module, the output module, the DSP module. After the integration of the modules in order to form the system, the whole system is going to be tested. Also, the hardware validation of the DSP and output modules can be done without a working input module. A randomly generated signal in FPGA can be processed and sent to output module for playing and this can be tested.

Testing Coverage

As explained in Section, the modules are going to be tested individually at first.

The input module is going to get an analog signal from an external source. This might be coming from the electric guitar or directly from a PC. If the input signal is coming from PC, the signal can be adjusted to be simple and therefore testing can be simpler. After the conversion, the signal is going to be observed. Also, if the output module is working, the input signal can be directly transferred to the output module without any signal processing done on it.

A randomly generated signal inside the FPGA is going to be enough to test the output module.

DSP module is going to be tested by functional verification. The filters,

the gain and the saturation blocks are going to be tested. After these, the

whole DSP module is going to be tested. An example of input and output of the

system with hard clipping.

Pass/Fail Criteria

The pass/fail criterion for the input module is going to be its analog to digital conversion performance. If a given analog input can be correctly converted to digital signal, it is going to pass the test. Digital conversion operation with its input and expected output.

The module, the expected output is going to be the signal on the bottom (Azima DLI , 2009). In order to pass the test, the module has to give the correct output for each stimulus applied.

The output module has to do digital to analog conversion and send the signal to speakers. For that, a signal is going to be generated inside the FPGA. This signal's amplitude and frequency is going to be changed. According to the changes, we are going to expect different outputs. The output is going to be listened through the speakers. In order to pass the test, the output module should correctly respond to every amplitude and frequency change.

The DSP module is going to be tested with functional verification. A reference model is going to be constructed in behavioral level. Randomly generated stimulus is going to be applied to the design and to the reference model at the same time. In order to pass the test, the results from the DSP module and the reference model have to match 100%. Another important criterion for the DSP module is its timing. The delay between the input and the output has to be below a determined quantity in order to pass the test.

Testing Approach

In order to test the DSP module, a self checking testbench is going to be written using Verilog HDL. There is going to be a behaviorally modeled reference unit inside the test bench. The test bench is going to generate random stimuli. These stimuli are going to be applied to both a design under test unit (DUT), which is a module from the design, and the reference model. Then, the results are going to be compared in a scoreboard. The verification approach.

Also, the timing of the system is going to be considered since it is one of the most important parts of the project. After the functional verification, the timing analysis of the implemented system must be done using Xilinx ISE.

Testing Resources

First, in order to test the algorithm, MATLAB Simulink is going to be used.

In order to test the input module, preferably a PC or an electric guitar is going to be needed as discussed in the second paragraph.

To test the output module, speakers or headphones are going to be needed.

For functional verification of the DSP module, Modelsim XE is needed. Also, for the timing analysis of the design, Xilinx ISE is going to be used.

Test Cases

After these inputs are applied, the outputs from the reference model and the DUT are also going to be stored in response file, which is going to be in . txt format. Finally, a log file is going to show where the errors occurred, if there are any errors or it is going to show that no errors occurred in the simulation. Looking at the log file and the response file, we are going to able to see where exactly the errors occurred.

Test Activities

Since the group has just one member, every part of testing is going to be done by me. More training about writing self checking test benches using Verilog HDL should be done.

6. Schedule

If we look at the PERT chart, we can calculate the critical path. The critical path consists of the following activities: A-F-G-H-I-J-K. This path leads to a completion time of 133 days.

If the most optimistic and the most pessimistic completion of each activity is estimated, we can calculate the expected completion time and the variance of the project. The expression for the expected completion time is given in Equation and the expression for variance is given in Equation. Using these equations, the completion time and the variance are calculated. The activities in the critical path are highlighted and the calculations are done according to the critical path.

PERT calculation gives almost the same result with the CPM result. CPM result was 133 days. PERT calculation gives an estimated project completion time of 133. 166 days. Also, the variance turned out to be 26. 58. This means the project can be completed 26. 58 days earlier or later.

The Gantt Chart of the project is given. The estimated start date of the project is December 27, 2009. The project is planned to be completed on May 9, 2010.

Bibliography

Azima DLI . (2009, February 8). Analog to Digital Conversion. Retrieved November 29, 2009, from Azima DLI Corporation Web Site: http://www. azimadli. com/vibman/analogtodigitalconversion. htm

Collicut, M. (2009, March 3). Extending the Karplus-Strong Algorithm to Simulate Guitar Distortion and Feedback Effects. Retrieved November 29, 2009, from McGill University Web Site: http://mt. music. mcgill. ca/~collicuttm/MUMT618/KSA_distortion_and_feedback. html

Jaffe, D. A., & Smith, J. O. (1983). Extensions of the Karplus-Strong plucked string algorithm. Computer Music Journal , 56-69.

Schulzrinne, H. (2008, January 9). Explanation of 44. 1 kHz CD sampling rate. Retrieved November 27, 2009, from Columbia University Web Site: http://www. cs. columbia. edu/~hgs/audio/44. 1. html

Smith III, J. O. (2009, March 21). Dynamic Level Lowpass Filter. Retrieved November 28, 2009, from Stanford University Web Site: https://ccrma. stanford. edu/realsimple/faust_strings/Dynamic_Level_Lowpass_Filter. html

Smith III, J. O. (2009, March 21). Pick-Direction Lowpass Filter. Retrieved November 28, 2009, from Stanford University Web Site: https://ccrma. stanford. edu/realsimple/faust_strings/Pick_Direction_Lowpass_Filter. html

Smith III, J. O. (2009, March 21). Pick-Position Comb Filter. Retrieved November 28, 2009, from Stanford University Web Site: https://ccrma. stanford. edu/realsimple/faust_strings/Pick_Position_Comb_Filter. html

Smith III, J. O. (2009, March 21). Two-Zero String Damping Filter. Retrieved November 28, 2009, from Stanford University Web Site: https://ccrma. stanford. edu/realsimple/faust_strings/Two_Zero_String_Damping_Filter. html

Sullivan, C. R. (1990). Extending the Karplus-Strong Algorithm to Synthesize Electric Guitar Timbres with Distortion and Feedback. Computer Music Journal , 26-37.

The Train Kept Rollin’. (2009, November 21). Retrieved November 21, 2009, from allmusic: http://www. allmusic. com/cg/amg. dll? p= amg&sql= 33: jjfoxzq0ldte

Xilinx Corporation. (2009, April 8). Getting Started with FPGAs – FPGA vs. ASIC. Retrieved November 20, 2009, from Xilinx Corporation Web site: http://www. xilinx. com/company/gettingstarted/fpgavsasic. htm

Xilinx Corporation. (2009, October 6). Spartan-3A Starter Kit. Retrieved November 27, 2009, from Xilinx Corporation Web site: http://www. xilinx. com/products/devkits/HW-SPAR3A-SK-UNI-G. htm