Stringless Guitar – Proyecto Final

Cuervo B. Nicolás y Horta C. Juan Daniel

cuervonicolas@gmail.com

***Abstract* - We've been working along the semester developing the course project: the stringless guitar. As a first step, PWM was implemented. From there, the effort was directed to generate the different notes of the guitar, and also, to emulate the guitar's sound. To accomplish this, FIR filters were very useful, as functions for the surrounding wave (which give us the similarity of the guitar sound). The inputs of the system were send as interrupts via IRQ-GPIO using software developing; meanwhile, the synthesis of the sound is made as a hardware task to make it faster. An additional peripheral speaker is connected to a jack plug to hear the guitar, making it a digital "electro-acoustic look alike" device.**

Index terms- FPGA, guitar, PWM, sound synthesis

I. INTRODUCTION

In this document the development of a *Stringless Guitar* is exposed. This was the result of a 6 months project, in which sound generation (by the means of a processor) was achieved by using Pulse Width Modulation - PWM. With this end, different oscillators were developed for each of the frequencies of interest (one for each pitch note), using a tooth-saw waveform, because of the harmonic-richness of this type of waves. The input of the sound synthesizer comes from a pre-signal processing applied to the interruption of light beams which pointed directly to a set of photocells, and so simulating the “strings” of a guitar. In the digital domain, this signals are sent into the processor via the GPIO ports. The output of the system is an audio signal, which can be connected directly to a set of speakers. In this project an LM32 processor was used, and its architecture was understood during the semester in order to be manipulated to suit our needs in our specific implementation.

II. OBJECTIVES

General: Design and implement a *Stringless guitar*.

Specific:

- Generate signals that represent real sound by the means of hardware instantiation.
- Filter the generated signal so that different frequencies could be selected by choice, and so that the resultant filtered signal resembles the sound of a guitar.
- Implement the physical structure for the User Interface.

III. Time table

Activity	Description	Estimated required time	Achieved on time?
Proposal	The project is proposed and the team waits for its approval or request for corrections.	1 Week (6th week)	Yes
R&D	-Generation of the signals. -Filter	3 weeks (7th to 9th	Yes

	development	week)	
UI	Implementation of analogue preprocessing.	1 week (13th week)	Yes
Testing and optimization	Final touches and tests.	1 week (14th week)	Yes
Final Product	Design and assembly of the final product.	1 week (15th week)	Yes

Table. 1. Time Table

V. Behavioral summary

The main idea is to emulate a real guitar, but only that it won't use strings as signal triggers. The device will have instead a set of light emitters-receptors, which will trigger a signal on interruption; i.e. an interruption in the beam will tell the processor that a sound is needed to be generated. The physical shape of the final product does not depend on the implementation and will not differ from a real guitar except on its size.

In total a set of six light emitters/receptors will be part of the device, so that it's design resembles a real guitar. From each of these couples, the processor receives a signal in a different GPIO port, which will tell the processor to generate a different signal, which differs on the frequency with the others. In addition, four push-buttons are embedded into the neck of the guitar, which emulates 4 frets, and will serve as signals to select different frequencies when pressed.

The signal flow that describes the system can be seen in the Fig. 1.

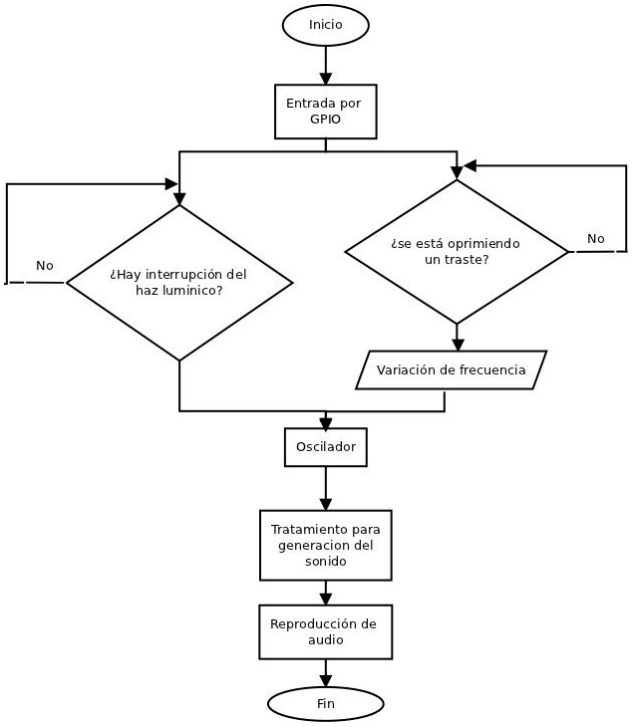

Fig. 1. Signal flow of the system

The device has different types of constraints, that can be categorized as follows:

Temporal constraints: The device generates sounds in real time, as a response to a physical interruption between the beam emitter and the receiver. Furthermore, the frequency of such signal is also affected by the pressing on the "frets". This clearly adds some processing overhead, but it is not long enough to be perceptible.

Physical constraints: The device is installed in a small wooden guitar, whose dimensions are 56cm x 21cm x 9cm.

Functionality: The light is emitted from common "light indicators" or "lasers" such as the ones that are used in common presentations, while the light receptors are photocells. The stable state of the device is when the light indicators are constantly lightning the

photocells. Any interruption in this process will trigger a signal that goes directly into the processor, which reacts by generating a sound. In addition, the buttons that emulate the frets send signals that modify the frequency of the output audio signal.

Electrical constraints: The device has to be fully functional by being feed with a 3.7v battery.

V. SYSTEM ARCHITECTURE

The device's task are divided into hardware tasks and software tasks as follows:

Hardware tasks: The processor is implemented in a Xilinx FPGA, for which the Xilinx compiler and bitstream generator is used. The hardware modules are written in verilog. Mainly the tasks that are done by the FPGA are the ones that are time-critical, from which the intrinsic speed of such a gate array serves for the purpose of the real-time constraints. The tasks done in hardware are:

- Reception of external signals from the push buttons and the light interruptions.
- Generation of the oscillator and thereby post-processing of its output in order to generate the sounds.
- Generate the actual sound signals.

Software tasks: For this part of the system, the compilation tools for the LM32 were used. That processor was described and programmed by writing its driver in C language. The tasks done by the processor are purely system control.This is a very important part of the system, because it deals with the external interruptions and the sequence in which the generated signals are played by the speakers.

Peripheral's control and audio generation: all modules, clearly, have to be interconnected with each other, and all of the with the processor. In order to have a nice communication between all the elements, a *Wishbone* protocol was implemented, in which it was precise to determine "hierarchies" for every block of the system. In this order of ideas, the hierarchy was determined as follows:

- One master: the CPU
- Two slaves: the GPIO and a "sound" module.

The block diagram can be seen in the Fig. 2. All elements are connected by the means of a *Conbus*.

VI. Structure

The device operation can be divided in stages that are characterized by the specific function that they have in the process. These stages can be seen in the Fig. 3.

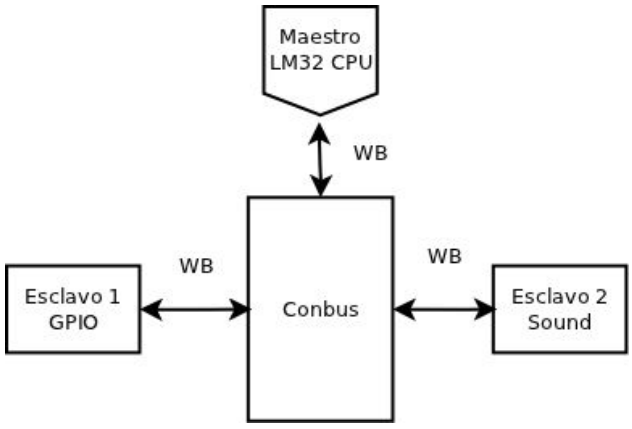

Fig. 2.Wishbone protocol intercomm.

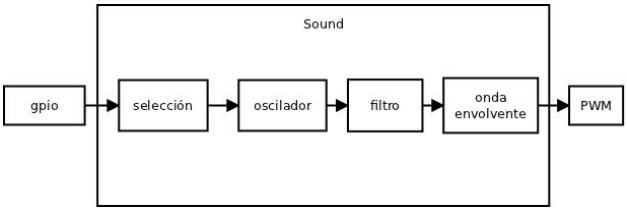

Fig. 3. Block diagram of the device.

Following the data flow from left to right illustrated in the Fig. 3., the first block is the GPIO, which serves as the gate for the external signals (light beam interruption and pushdown buttons). This signals go into the big block "sound", were all the audio is synthesized. This module is divided into subtasks such as:

Selection: Basically, this module selects the frequency of the output signal based in the beam of light that was interrupted (which emulates the selection of the "string" that was played). This selected frequency then is driven into the oscillator.

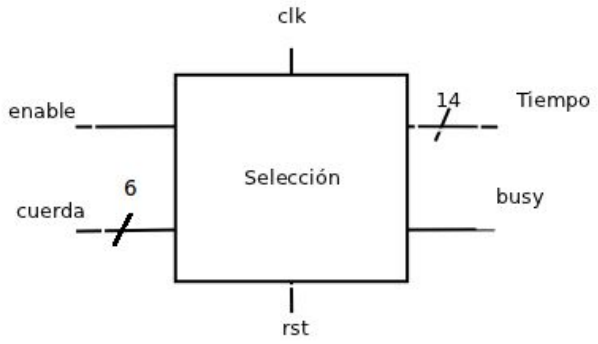

Fig. 4. Selection I/O

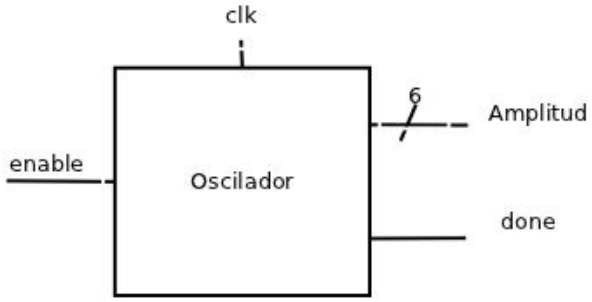

Fig. 5. Oscillator: I/O

Oscillator: in this module a tooth-saw signal is, by the means of hardware instantiation, generated. This is done by using a counter. The module, internally, has a set of parameters that determine the frequency of the output signal. However, these parameters are selected based in the "selection".

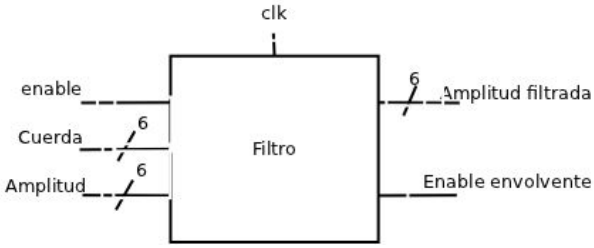

Fig. 6. Filter I/O

Filters: this module limits the bandwidth of the signal generated by the oscillator, so that the amplitude changes are not abrupt. Here, FIR filters were used because they require less resources to be implemented (in comparison with IIR filters). The basic operation of an FIR filter can be seen in the Fig. 7.

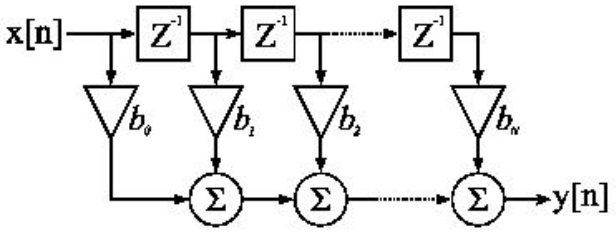

Fig. 7. FIR filter diagram [1]

What is interesting about this filters is that they are *memoryless*, which means that its behavior depends only in the present value of the input signal and not from past values, and their response is always finite.

Physically are the FIR filters the elements that allow to achieve a similarity in the sound from the generated signal with a real guitar. Given the relative complexity of this system (FIR filter of 10th order), a whole sets of parameters, or *taps*, is required, and the modification of such taps affect the frequency response of the filter and, consequently, the sound that is generated from the system.

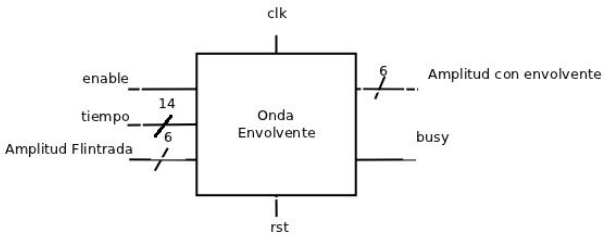

Fig. 8. Surrounding wave I/O

Surrounding wave: This module is in charge of emulating the duration and intensity of the sound of the generated signal. It is easy to identify that, when a string in a guitar is played, the amplitude of the sound wave increases rapidly until it reaches its maximum value, and then it decreases slowly with time. This process has been well studied by the sound scientists, who have identified four stages of a sound wave from the moment it is generated until the sound ceases. These stages are: attack, decay, sustain or release.

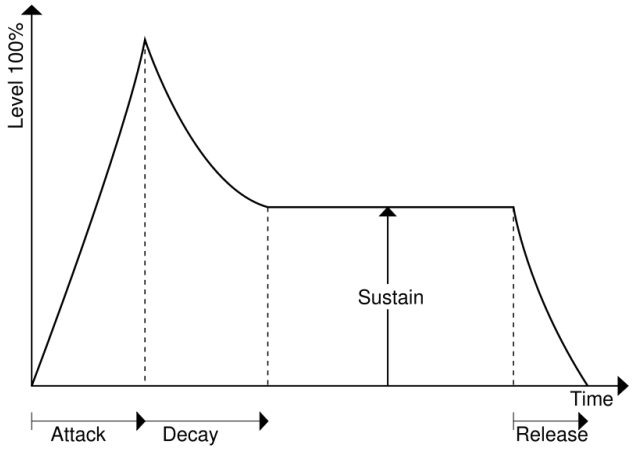

Fig. 9. Shape of the surrounding wave and its stages.

For each of these stages, a different submodule was developed. Each of them was in charge of vary the amplitude of the filtered signal with different rates of changes, and so a similarity with the temporal decay of a note when it is played was achieved.

PWM: Pulse-width modulation is a technique that modifies the width of a periodic pulse. This width is called the "work cycle", whose measure is taken in regard with the period of the signal.

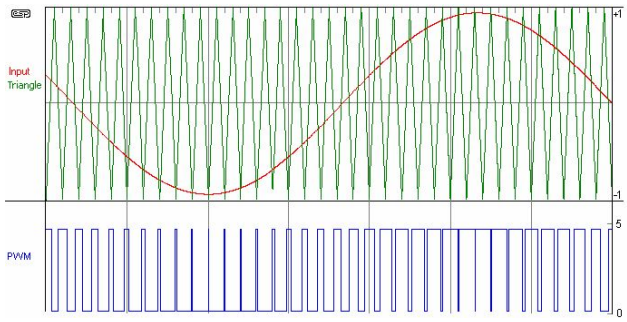

Fig. 10. PWM

With this module, signals with different frequency were realized; the frequency determines the note that is being played.

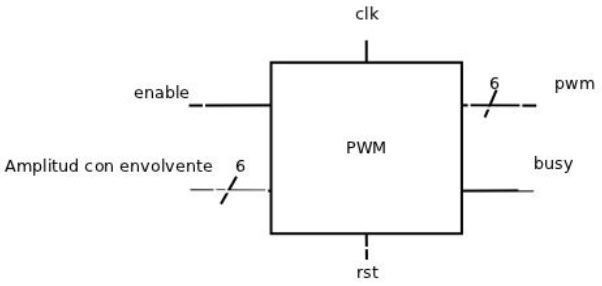

Fig. 11. PWM Module

The last part of the audio generation was the PWM and its use to perform a ADC conversion without the need of an additional ADC circuitry. In this module, the width of a work cycle for each pulse is closely related with the amplitude of the signal after going under the effect of the surrounding wave. With this method, it is possible to get at the output of the system a signal that can be directly connected to a speaker.

VII. Assembly

In order to accomplish the signal preprocessing, a small PCB was developed, to which the light sensors are connected. The signal from the sensor is then connected to the FPGA. The circuit schematic is depicted in the Fig. 12.

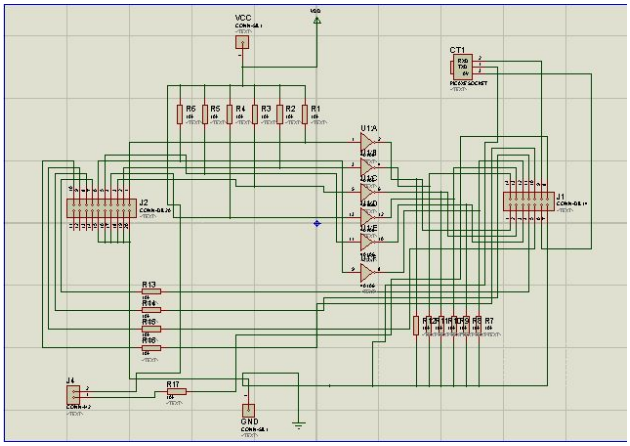

Fig. 12. Circuit schematic.

Given the continuous value of the signal measured by the sensor, a type of quantization was implemented by the means of a “Schmitt trigger”.

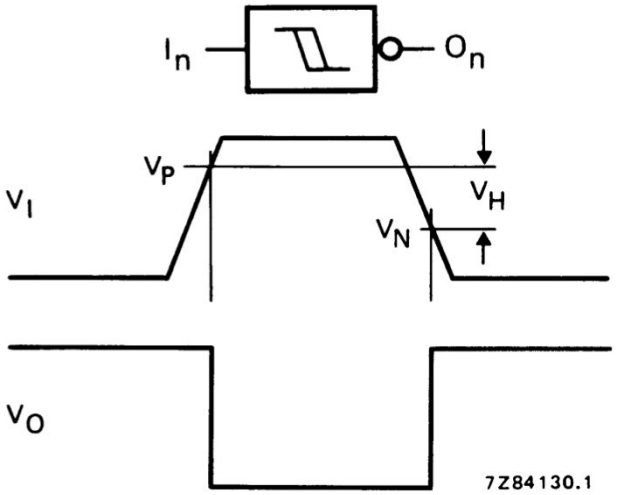

Fig. 13. I/O Waveform of a Schmitt trigger gate.

This gate works in a straightforward fashion: a voltage level is set up by the means of feeding resistors, and this voltage level works as a threshold. When the input surpasses the threshold, a HIGH signal is put at the output. The output is LOW otherwise.

After this preprocessing, the PWM output signal is sent to external speakers, which are connected to a 3.5mm Jack port. This procedure is low cost and reduces the need of further amplification and filtering of the signal.

The PCB design was done using “Proteus”. The final design can be seen in the Fig. 14.

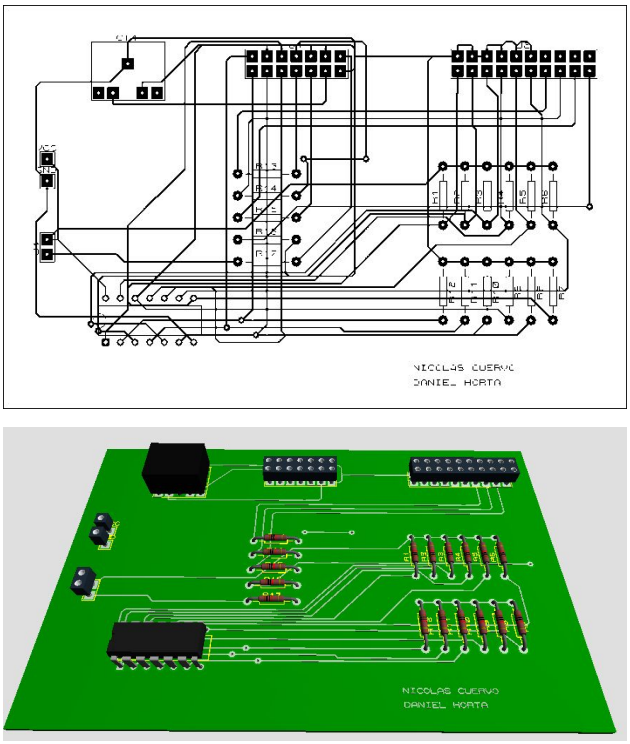

Fig. 14. PCB design

IIX. COSTS

The costs listed in the following picture are in colombian pesos.

Development cost per uint				
Category	Item	Unit price	#	Total
Physical structure	Wooden	$35.000	1	$35.000
	guitart			
	Light indicators	$2.500	6	$15.000
	Photocells	$800	6	$4.800
	Acrylic	$10.000	1	$10.000
	Screws	$200	18	$3.600
Electronics	Passive components (Resistors)	$50	14	$800
	Schmitt trigger gate	$1.400	1	$1.400
	PCB 7cm x 14cm	$28.000	1	$28.000
	PCB 3cm x 7cm	$14.000	1	$14.000
	Connectors	$7.800	1	$7800
	Dev kit (spartan starter 3E)	$230.000	1	$230.000
Engineering	Dev.	$5’400.000	2	$10’800.000

Total= $11’150.400

IX. FINAL REMARKS

- There are a lot of different waveforms that can be generated by the means of hardware synthesizing, but it is required to observe the specific characteristic of each of them in order to determine it they are suitable for a specific application. We used in this project a sawtooth wave because of its spectral richness.
- FIR filter are used to cut out spectral content that is not wanted in the application, and its usefulness was clearly shown in this project. They are easy to implement in hardware as well, because they are not recursive and thereby they don’t require much

resources in comparison with the IIR filters.

- In order to accomplish a sound quality that resembles a real music instrument, the surrounding waveform suits the needs to emulate the variation over time of such real sounds. Without this processing, the generated sound would seem constant and plain since the moment of its generation and for the given play time.
- Regarding the “laser” emitters, the main idea was to buy the commercial and inexpensive devices that are used in common presentations and then remove the electronics from the case in order to then adapt it to our custom design. However, those devices were very sensitive to further manipulation such as mechanical stress and soldering, which led to the decision of keeping them in the original case and modified our custom design instead. This made our electronic design slightly change as well, because we opted to feed the indicators with the original batteries too. This part of the design requires further improvements in order to have a unique power source in the device.
- The final product represents the work and development during a whole semester of the digital design course. This development required the understanding and application of the LM32 processor architecture. Beyond the sound synthesizing and reproduction, this project focuses more on innovation, where it tries to apply existing technologies in novel ways.

X. References

[1] R. Cádiz, FIR filters, [Online] Available at: ”http://www.rodrigocadiz.com/imc/html/ Filtros FIR.html” Date: 13-06-2012

[2] H. Taigen, Musical Blocks, [Online] Available at: ”http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2009/jvt6 th389/jvt6 th389/finalproject.html” Date: 6-06-2012

[3] S. Vanheesbeke, Swapping bits improves performance of FPGA PWM counter, EDN Europe: Electronics DEsign, Strategy, News [Online] Available at: ” http://edn.com/design/analog/4318138/Swapping-bits-improves-performance-of-FPGA-PWM-counter” Date: 07-05-2012