VOICE TRANSMISSION WITH ECHO/REVERB

by

151220114072 Girayhan Sarıkaş

151220122110 Barış Can Çam

151220114071 Ceren Salman

An Engineering Term Project Report

Supervisor, Advisor:

Asst. Prof. Erol Seke

Electrical-Electronics Engineering Department

ESKISEHIR OSMANGAZI UNIVERSITY

January 2016

ABSTRACT

In this project, voice signal which is generated by the voice signal generator, would be first amplified by pre-amplifier and converted to a digital signal by means of Analog to Digital Converter (ADC). Then, required effects are added. At last digital data is serialized and send to slave FPGA through a frame. After transmission achieved, signal is converted to analog by means of Digital to Analog Converter (DAC). Analog signal is amplified by an additional music player system.
(Both echo and reverb effects are available in the study)

ÖZET

Bu projede, sinyal jeneratöründen alınan ses sinyali önce pre-amfi ile yükseltilerek ADC sayesinde dijital sinyale çevrilmiştir. Proje tanımında istenen efektlerin eklenmesinden sonra sinyal seri hale getirilip, bir çerçeve aracılığıyla köle FPGA'e gönderilmiştir. İletim sonunda, reverb/echo efektli ses sinyali ilave bir müzik çalar sistemi ile dinlenmiştir.

TABLE OF CONTENTS

1. Background

Digital voice transmission is composed of three main areas: converting an analog signal into a digital signal. (Analog to Digital Conversion-ADC), processing and sending that digital signal, then creating new analog signal from the processed digital signal (Digital to Analog Conversion-DAC). In voice transmission, the ADC step must take sufficient number of samples per second (sampling frequency). The DAC component must be interacted with an output device to sense transmitted signal. Most times a speaker is used with a combination of amplifier. Another important part of a voice transmission system is communication, to obtain a clear signal transmission probability of error during communication process must be reduced. Due to that, a simple method is used for error correction in this project.

2. Introduction

We set out to implement half-duplex, digital, wired communication system. This project required us to exercise our knowledge of digital communication learned in communications lecture, knowledge of signals learned in signals & systems lecture, knowledge of Very High Speed Integrated Circuit Hardware Description Language (VHDL) coding learned in introduction to VHDL/FPGA lecture.
Our main platform for development was Spartan 3E Starter Kit which includes Xilinx XC3S500E Spartan - 3E FPGA. Spartan 3E Starter Kit contains on board four-output, SPI-based 12 bit Digital – to – Analog Converter (DAC), two – input, SPI based 14 bit Analog – to – Digital Converter (ADC) with programmable gain pre-amplifier. There are up to 232 user – I/O pins, 64 MByte of DDR SDRAM.

We used Xilinx ISE Design Suite to write VHDL codes, synthesis of codes, implementation of design and generation of programming file.

This report details the method we were applied, challenges we faced in implementing the design, the solutions we came up with to address some of these challenges, and the final results of our endeavors.

3. ADC/DAC

As stated above, the Spartan 3E Starter Kit contains a on board 14-bit ADC which was used in the conversion of the analog voice signal received from voice signal source. The purpose of this step is to take enough samples to convert analog voice signal to digital signal. But there was a constraint for us to implement voice effects, these effects requires storage elements to cache collected input data to be able to obtain delayed version of the input data. In order to use memory inside Spartan 3E Starter Kit efficiently we limited sampling rate to about 25 kHz. ADC interface is shown in figure 1.

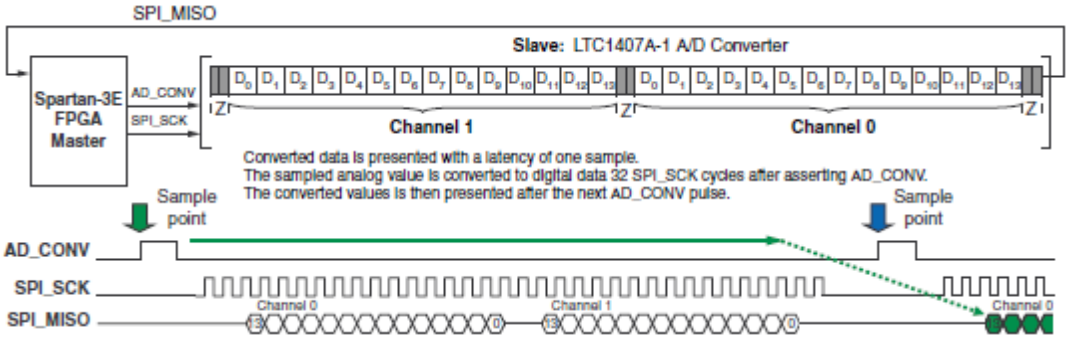

Figure 1. (Analog-to-Digital Conversion Interface)

According to figure 1 when the AD_CONV signal goes high, the ADC samples both analog channels. The results of this conversion are not presented until the next time AD_CONV is asserted, due to that AD_CONV signal determines the sampling rate of the system.

As stated above Spartan 3E Starter Kit contains 12 – bit on board DAC which was used in conversion from digital data to analog signal. We can also communicate with DAC a configuration based on SPI communication. Basic SPI communication protocol for DAC is shown in figure 2.

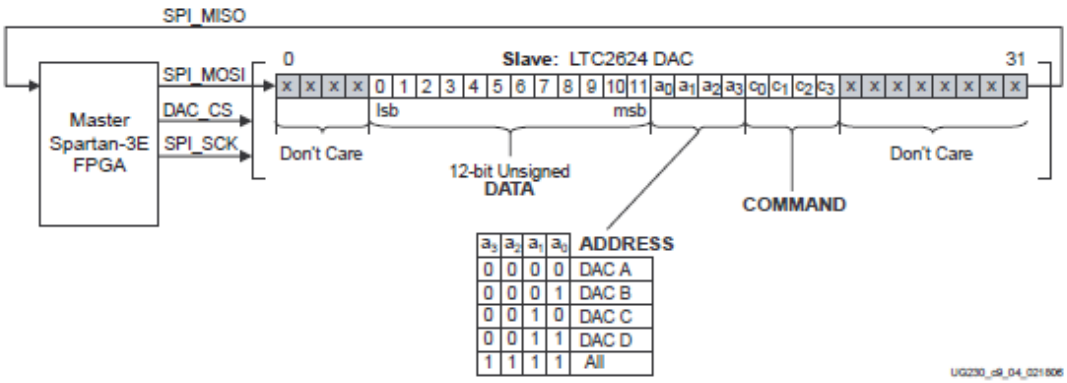

Figure 2. (SPI Communication Protocol to DAC)

As shown in figure 2 serial data is sent to DAC with address for DAC output, a constant COMMAND code. COMMAND code is used to activate DAC conversion.

To work with Digital to Analog converter we first created a test bench. A single period sinusoidal signal is created via MATLAB and generated values are stored

initially in a ROM. Stored data is sent to DAC serially and we see the sinusoid at the output of the as shown in figure 3.

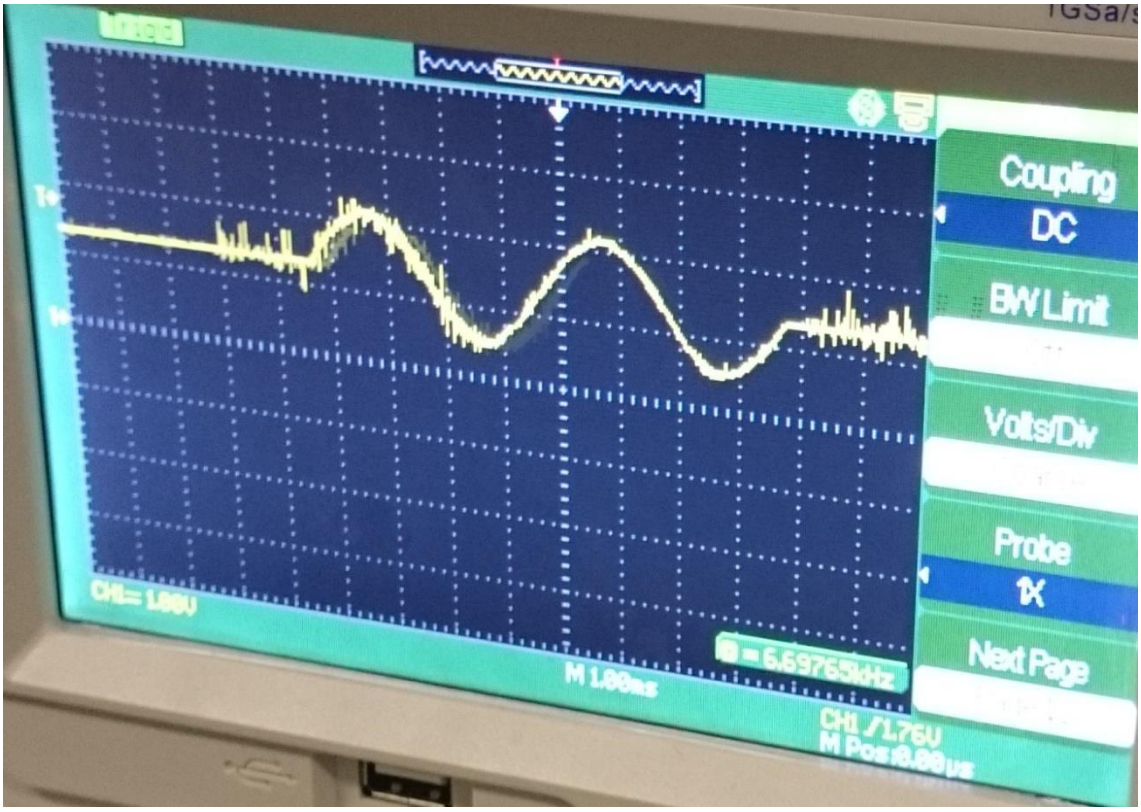

Figure 3. (Obtained Sinusoidal Based on Test-Bench)

4. ECHO / REVERB MODIFICATION

Echo and reverb are effects applied on voice signals stands on similar delayed versions of signals. To obtain delayed versions of signals we used 4096x12 RAM for both echo and reverb modification.

4.1.Echo

In audio signal processing and acoustics, an echo is a reflection of sound, arriving at the listener some time after the direct sound.A true echo is a single reflection of the sound source. In this project a simple approach is followed to generate an echo effect. Simply, original signal and delayed version of the original signal is summed a shown in figure 4.

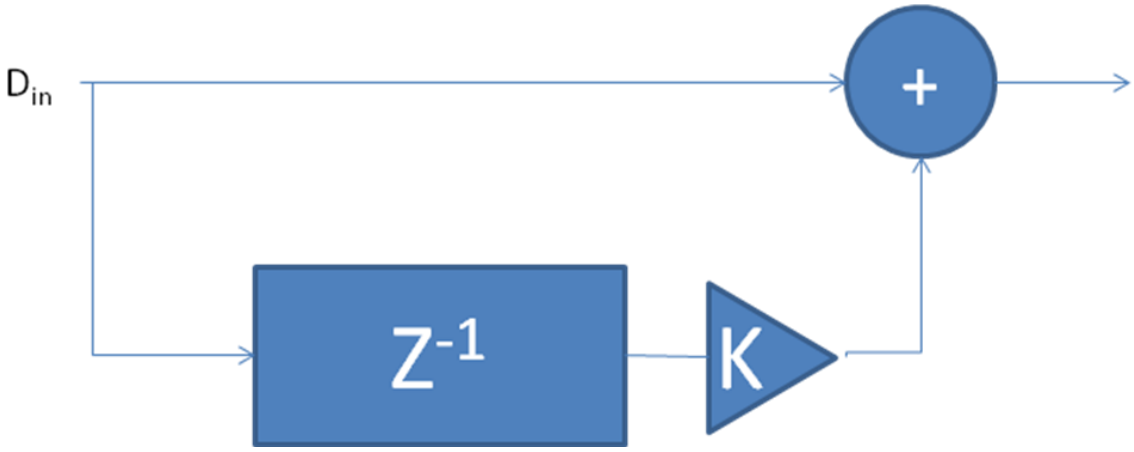

Figure 4. (Block Diagram for Echo Effect)

In this project, we applied echo effect with two different parameters of delay in order to listen echo effect clearly attenuation constant K left as 1. Echo parameters are shown in table 1. Delay parameter is calculated using a simple formula;

$$T_{delay} = ROM_{size} * T_{sampling}$$

Echo Parameters		
	Delay (sec)	**Gain**
Mode 1	*0.32*	*1*
Mode 2	*0.128*	*1*

Table 1. (Echo Parameters)

4.2.Reverb

In definition, reverb is the presistence of sound after a sound is produced. A reverb is created when a sound or signal is reflected causing a larger number of reflections to build up and then decay as the sound is absorbed by the surfaces of objects in space.

To model a reverb effect we used Finite Impulse Response (FIR) Model as shown in figure 5. Delay parameters of reverb effect are also calculated by means of the formula given in echo section. Obtained parameters are given in table 2.

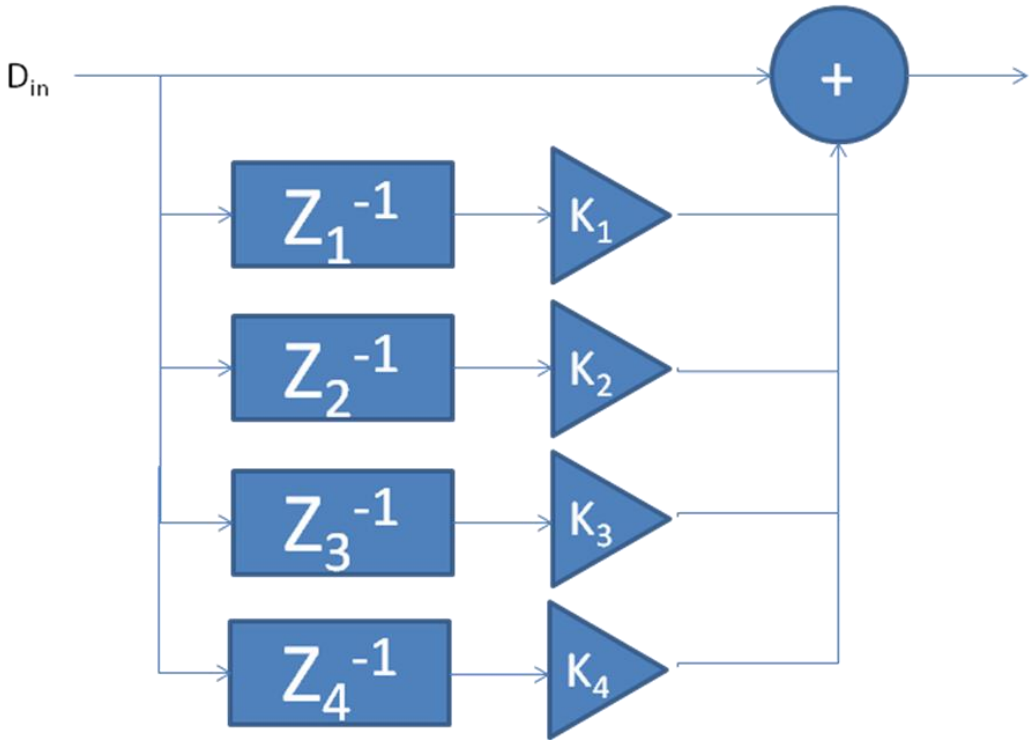

Figure 5. (Reverb Effect Block Diagram)

Reverb Mode 1 Parameters		
Components	**Delay (sec)**	**Gain**
1	*0.04*	*1/2*
2	*0.08*	*1/4*
3	*0.16*	*1/8*
4	*0.32*	*1/16*

Table 2a. (Reverb Parameters)

Reverb Mode 2 Parameters		
Components	**Delay (sec)**	**Gain**
1	*0.016*	*1/2*
2	*0.032*	*1/4*
3	*0.064*	*1/8*
4	*0.128*	*1/16*

Table 2b. (Reverb Parameters)

5. Communication

In this project we used Serial Peripheral Interface based serial communication method. We used frame based error correction to accomplish a succseful data transmission.

5.1. Transmitter

Transmitter section of communication method consists of frame inclusion and serialization of data, structure of generated frame is shown in figure 4.

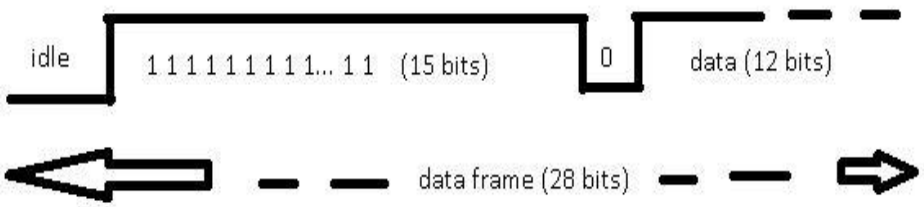

Figure 4. (Frame Structure Used in Communication)

5.2.Receiver

Receiver section of communication method consists of a register whose size is 28 bit (16 bit frame, 12 bit data). To accomplish a succesful data receive most significant 16 bits of register is checked, if this check operation captures dedicated frame remaining 12 bits captured as data. Register structure is shown in figure 5.

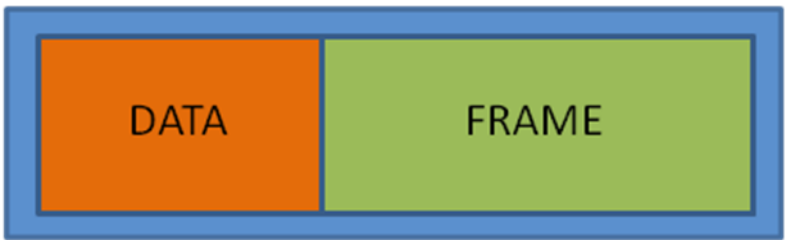

Figure 5. (Register used to in receive process)

6. Conclusion

In conclusion, we succesfully implemented serial communication method to transmit voice signal, we also succesfully implemented requested effects on voice signal. By means of this project, we gained experience how to handle a real-life problem for FPGAs using VHDL.

7. References

[1] Volnei A. Pedroni, Circuit Design with VHDL, 2nd Ed. MIT Press, 2004

[2] Oppenheim, Willsky, Nawab, Signals & Systems, 2nd Ed. Pearson Prentice Hall, 2004

[3] M. Morris Mano, Charles R. Kime Logic and Computer Design Fundamentals, 4th Edition, Pearson Prentice Hall, 2008

Appendices

Appendix 1

BLOCK DIAGRAM OF DESIGN

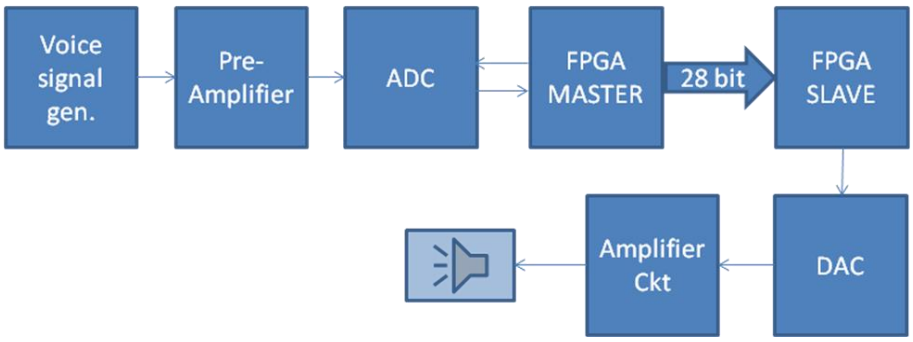

Figure 6. (Block Diagram of the Whole System)

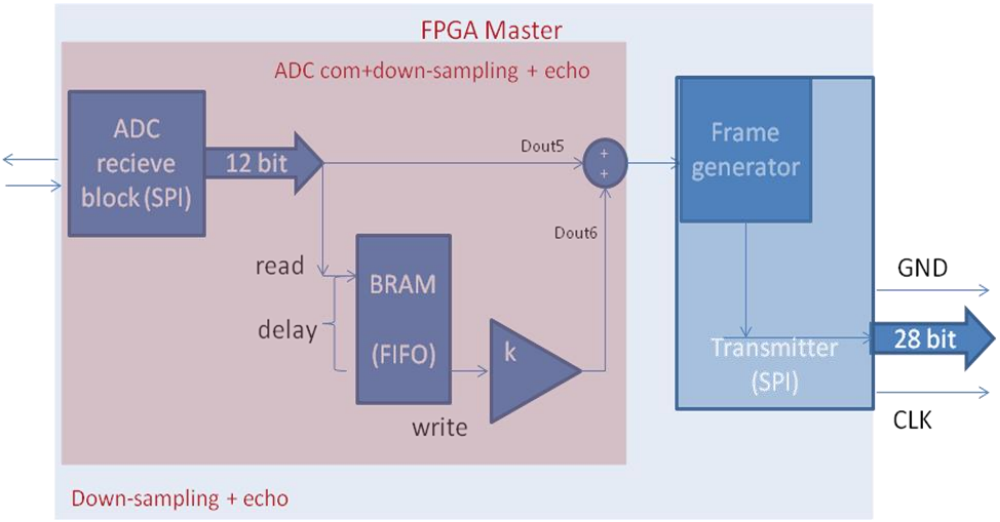

Figure 7. (Block Diagram of Master FPGA)

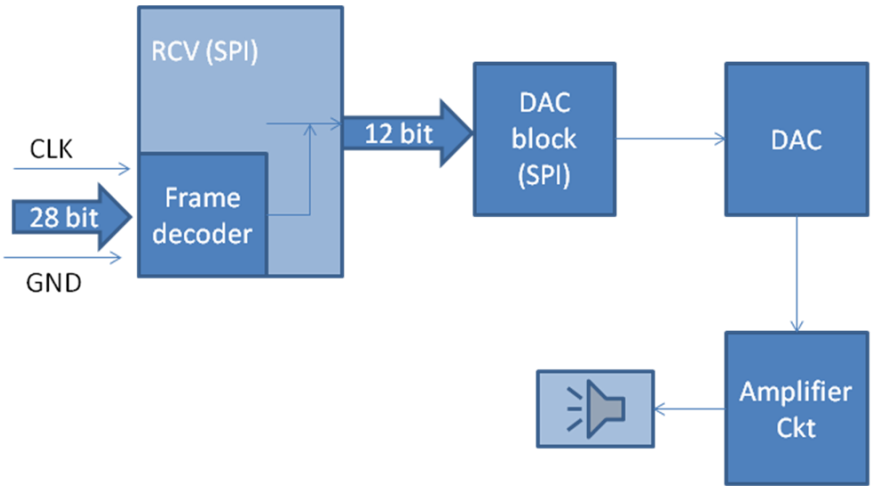

Figure 8. (Block Diagram of Slave FPGA)