EDA385 Final proposal

September 17

2012

A brief summary of the problems and solutions that will be encountered and employed during the project.

Frequency Visualizer and Echo effect

Jakub Górski, D07 (dt07jg8@student.lth.se)

Usman Farroq, (aso10ufa@student.lu.se)

Adeel Muhammad Hashmi(mas09am10@student.lu.se)

Course supervisor: Flavius Gurian

Contents

Introduction

The purpose of this project is to create a VGA audio visualizer that will be implemented in VHDL in its entirety. Apart from visualizing audio, the system will be able to optionally add an echo effect to the inputted sound, and consequently output it through a speaker.

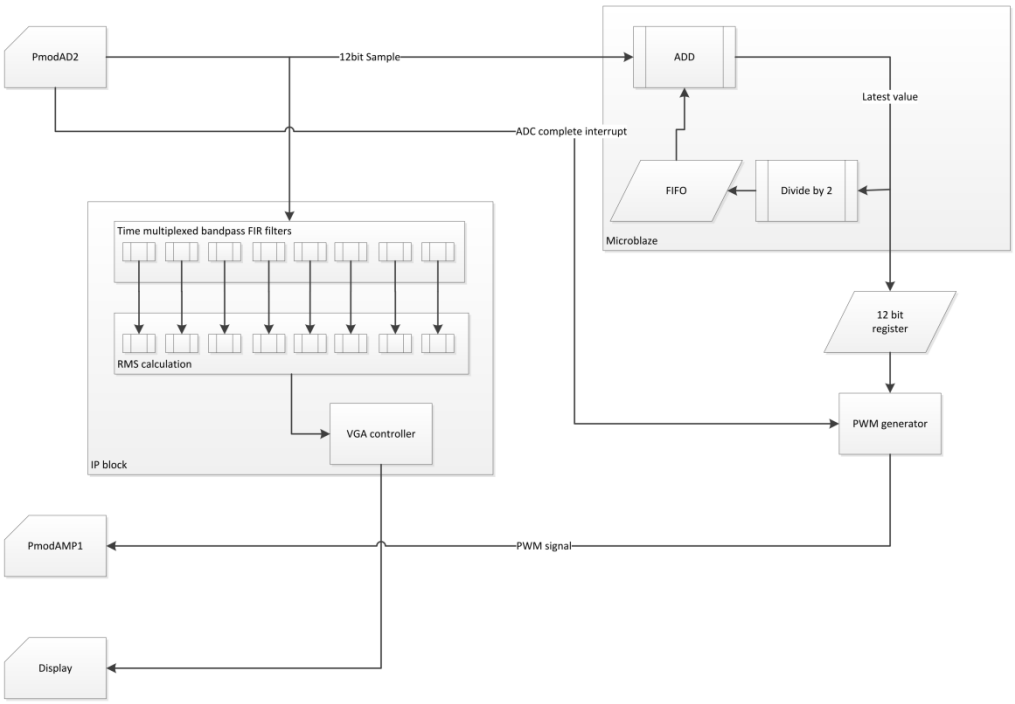

Figure 1: A block diagram of the proposed system.

Software

Echo processing will be the software implemented part of the system. As seen in Figure 1, it will be implemented inside the Microblaze processor and utilize a FIFO queue for buffering of the incoming samples.

Memory constraints

The size of the FIFO queue will denote the delay of the added echo signal. Due to the memory constraint of fast BRAM the maximum delay that can be achieved at a sampling rate of 32kHz is 1.125s.

Until the buffer is not completely filled the adder will operate with zero values coming from the FIFO causing a pass-through of the original signal for at most 1.125s, as mentioned before.

Controlling the Echo

The echo effect will be always running causing the only on and off regulation to be the volume of the PmodAMP1 module. The design intends the echo delay to be static, however it can be made variable

by changing the top pointer of the FIFO stack. Any changes to the pointer would be made by onboard buttons on the Nexys 3.

ADC and PWM

Both the PWM IP block and ADC need to be initialized before the system becomes runnable. This will be done inside the Microblaze by using interrupts and I2C serial communication. The PWM employed will be the same as given in the PmodAMP1's Diligent's demonstration project.

The I2C communication interface for the PmodADC2 will be taken from its corresponding Diligen demonstration project.

Hardware

FIR filter

The nature of the FIR filter IP is a time multiplexing bandpass filtering system. It will consist of eight FIR filter components where each is tuned to only accept a certain range of frequencies. The reason for having eight filters components is due to the decision of displaying eight bars on the visualizer.

Each of the filter components will be able to handle an input of 12bits and will have 32 taps.

Low end	Mid base	Low midrange	Midrange	High midrange	Lower highs	Middle highs	Top end
10-100Hz	100-300Hz	300-600Hz	600-1.2kHz	1.2kHz-2.4kHz	2.4kHz-4.8kHz	4.8kHz-9.6kHz	9.6kHz-20kHz

After FIR filter bandpass the values are relayed to an RMS calculation IP.

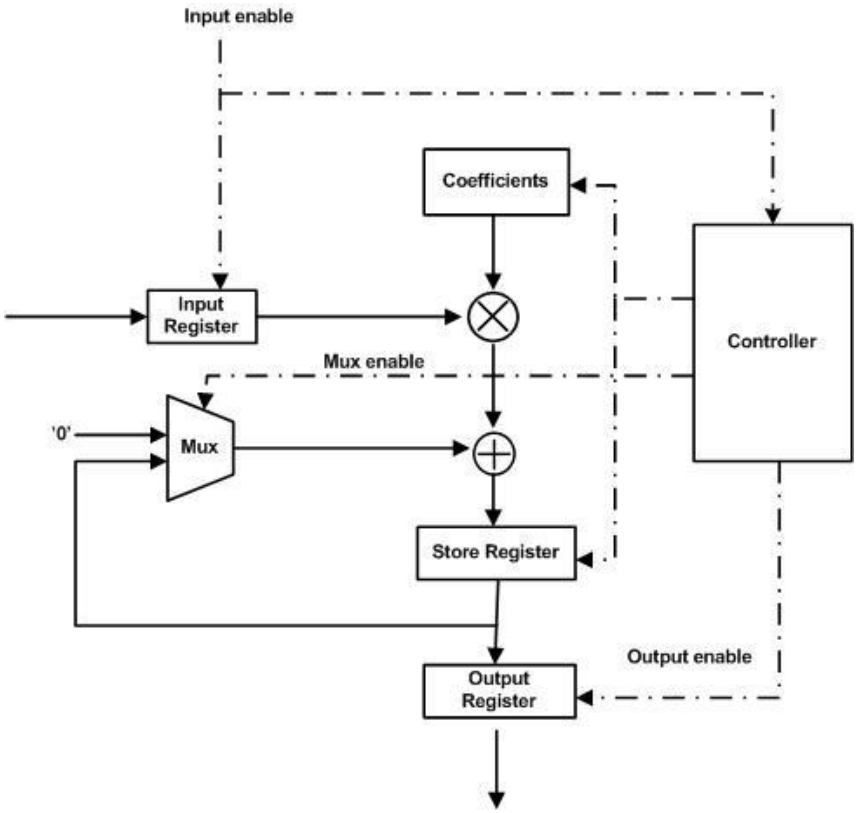

Figure 2: Diagram of one 12-bit time multiplexed, band-pass, 32-tap FIR component.

RMS

This component will essentially be the same as the one in Figure 2, with the difference being the absence of the multiplier and coefficients.

It will pass through an adder pipeline consequently adding 512 samples and storing them in an RMS value register. The stored data is then used to send out information about the height of the bars on the VGA display.

If the visualizer will look as presented in Figure 3 then 8-bits will be taken from the RMS register.

VGA driver

The display will operate at a frequency of 60Hz and a resolution of 640x480. Any rendered data will not require any RAM because all data will be made available by hardware in real-time.

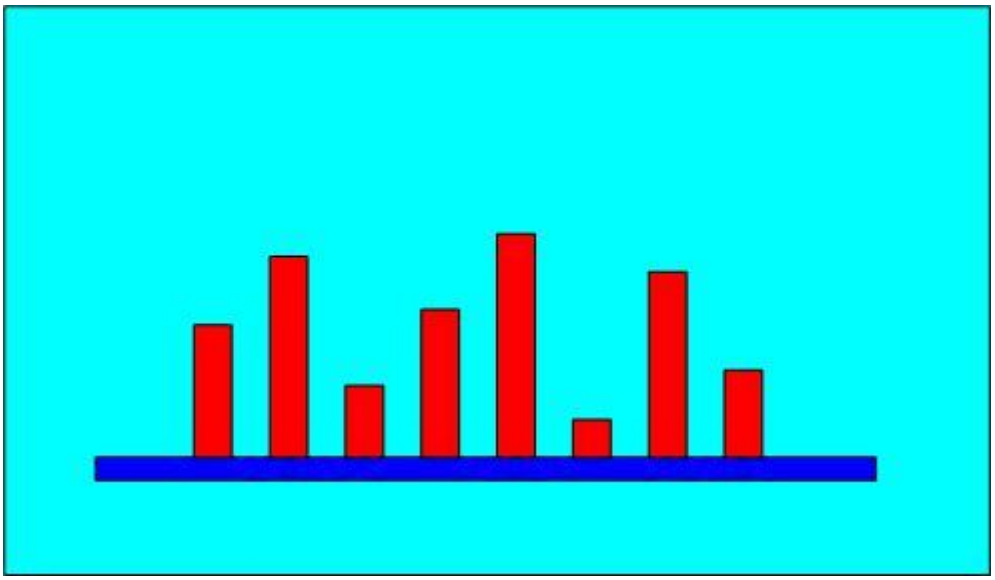

Figure 3: A drawing of how the displays bars may look like.