

Project Title: Audio visualization device.

Project Team:

Jennifer Mo	992476762
Gary Pong	992339144
Jason Wong	992353421

Date: January 22nd 2007

Project Description:

This project will implement a device which takes an input sound signal and outputs a visualization of that signal onto a VGA monitor. This will be similar to the visualization feature found in popular audio players such as Windows Media Player and WinAmp.

Our hardware implementation of audio visualization will require a MicroBlaze CPU for processing and a rendering module to provide hardware acceleration for drawing graphics. The AC97 audio codec will sample the input sound signal and its result will be processed by the Xilinx-provided Fast Fourier Transform block. Software running on the MicroBlaze will then analyze the frequency components of the signal and generate an appropriate visualization for it. Various different visualizations will be implemented, with their complexity increasing as we near project completion. The ZBT RAM modules will be used as frame buffers to store the pixels to be drawn to the monitor by the VGA controller.

Our main hardware block to be implemented is the rendering module. The rendering module will take commands from the CPU and translate those into pixels to be drawn. This will lessen the load on the CPU and allow the processor to issue commands such as “draw a line” and let the rendering module perform the computations required to do the actual rendering.

The system block diagram is shown in Figure 1 on the following page.

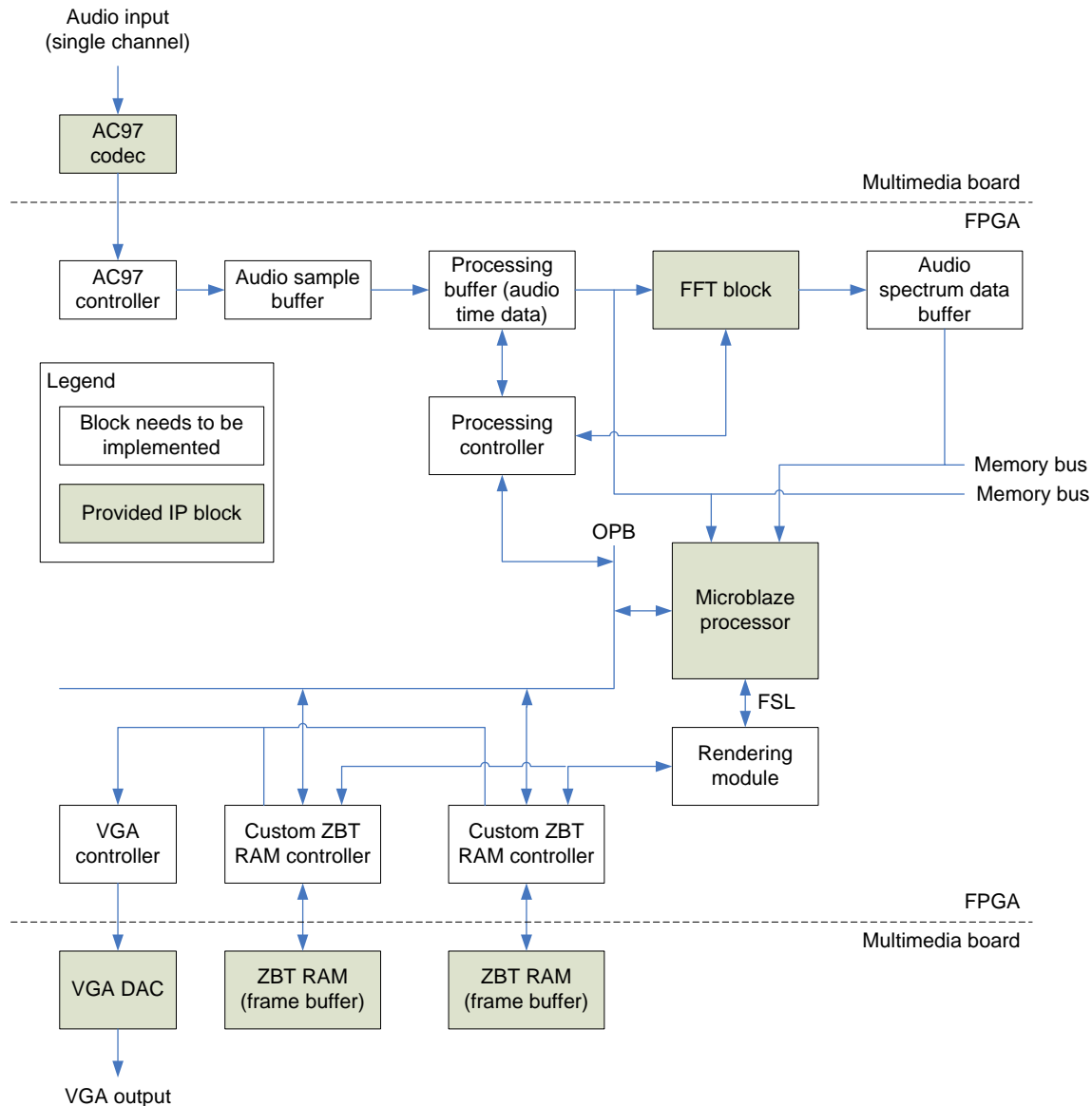


Figure 1 System block diagram.

Milestones:February 14:

- Able to perform FFT on an input sound signal captured from an audio source using the RCA input into the AC97 codec.
- Will demo the above by using two different sound signals of noticeably different pitches.
- The FFT result can be read from memory and displayed to console output.
- Sound with higher pitch should show more energy towards the higher end of frequency spectrum; sound with lower pitch should show more energy towards lower end of frequency spectrum.

February 28:

- Finish implementing VGA controller to draw front buffer to VGA monitor.
- Add basic drawing capabilities to rendering module (drawing straight lines).
- Finish basic visualization effect that shows the frequency spectrum of the input sound signal.

March 7:

- Will have implemented more drawing capabilities to rendering module
 - draw pictures from an off-screen buffer with colour keying (i.e. transparency support).
- Show a basic visualization effect that makes use of this rendering feature.

March 14:

- More drawing capabilities to rendering module (scaling the pictures drawn from an off-screen buffer).
- Show a basic visualization effect that makes use of this rendering feature.

March 21:

- More drawing capabilities to rendering module (rotating the pictures drawn from an off-screen buffer).
- Show a basic visualization effect that makes use of this rendering feature.

March 28:

- Create more elaborate visualization effects using the rendering features that have been implemented so far.
- Add logic to dynamically switch between visualization effects.
- Final project demonstration to show the above.

Resource Requirements:

- Audio source (e.g. mp3 player)
- VGA monitor
- Speakers