

Digital Theremin Synthesizer and Visualizer

6.111 Final Project

November 15, 2007

Alexander Spicer, Daniel Rodgers, Jeffrey Chang

Project Overview

- Functionality of end product:
 - Musical instrument with Theremin-style input
 - Audio effects synthesizer
 - Real-time music visualizer
- Should be fun to use and entertaining to watch!

Project Overview, cont.

■ System inputs:

- Movement of user's hands (while wearing gloves)
- External audio input (e.g., mp3 player)
- Computer keyboard (to select/apply audio effects)

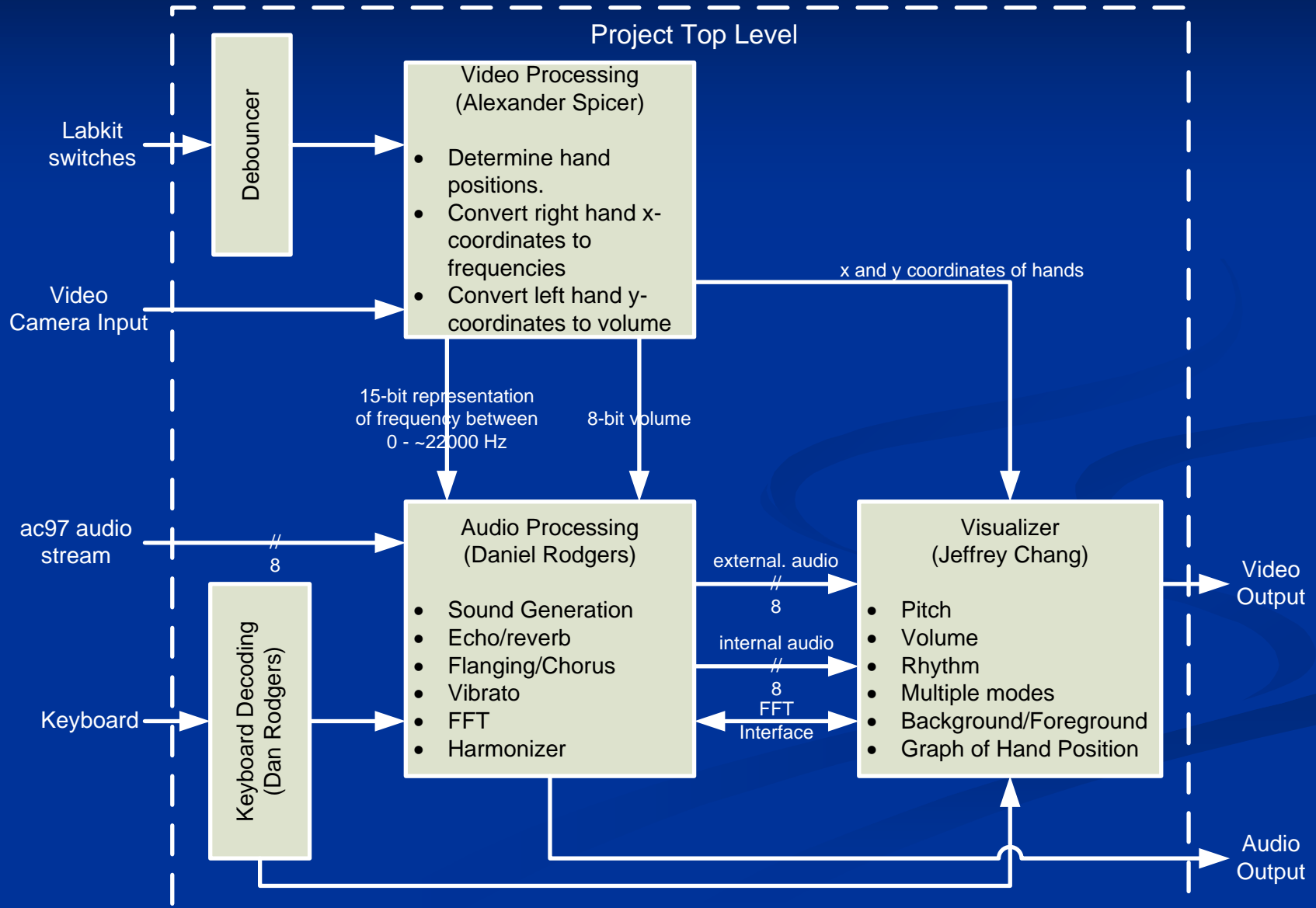
■ System outputs:

- Playback of music from mp3 player, via speakers
- Tones/effects generated from Theremin input
- Real-time music visualization on LCD monitor

Project Overview, cont.

- Our system will have 3 main modules:
 - Input module
 - Processes input from the video camera
 - Audio effects module
 - Applies audio filters as specified
 - Visualizer module
 - Displays position of hands, plus visualization of music

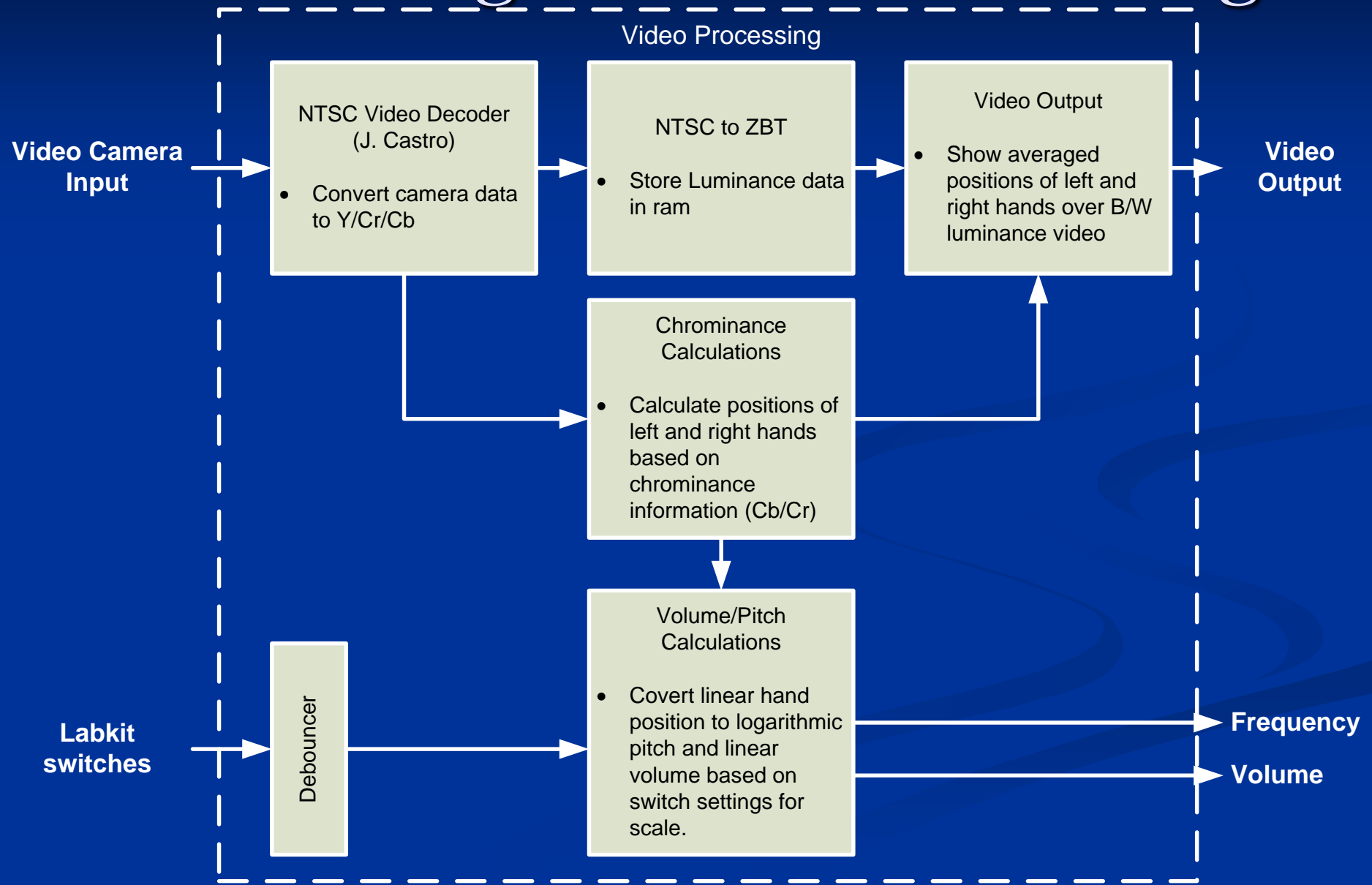
Top-level Block Diagram



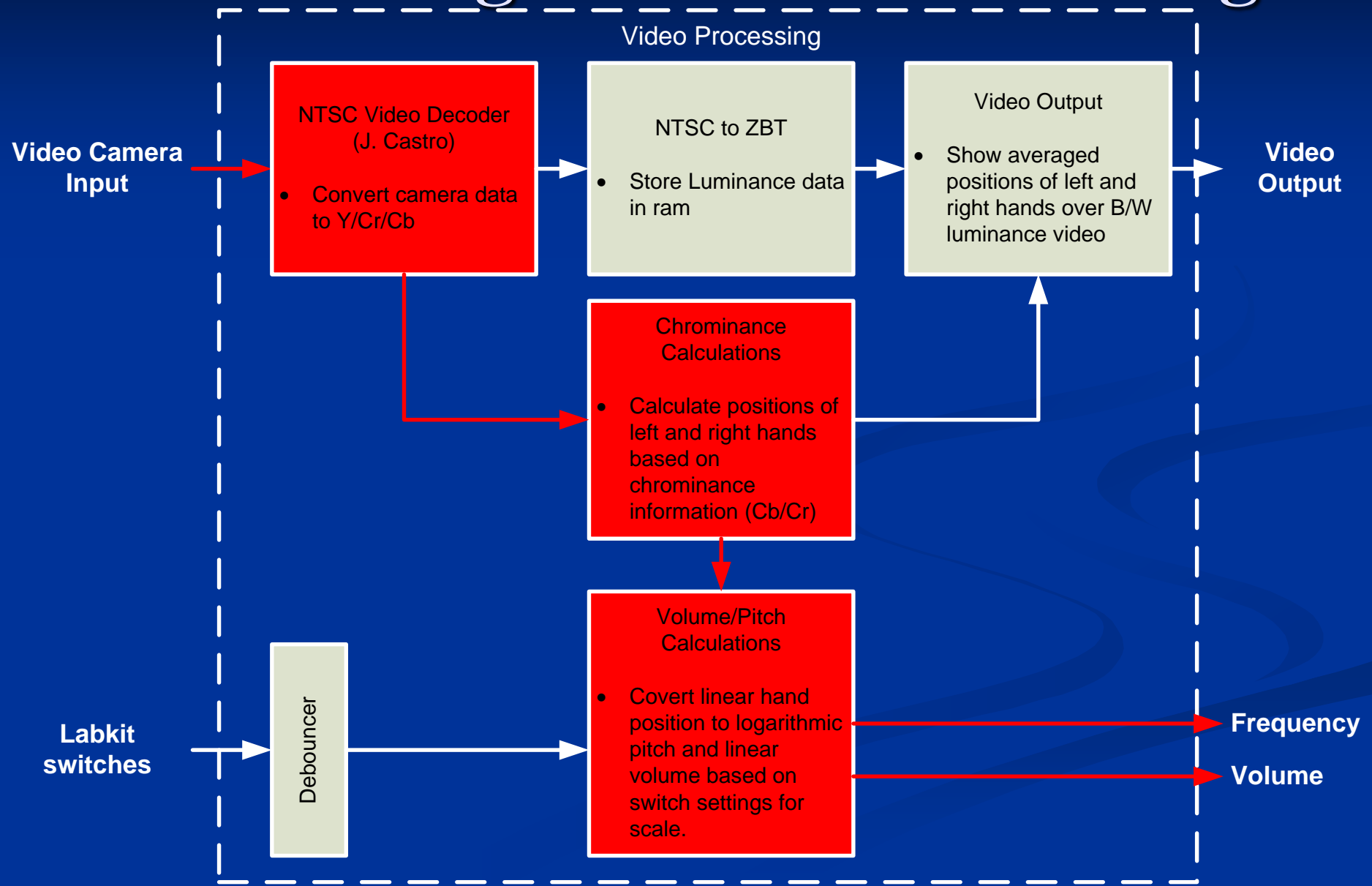
Video Processing Module

- Gloves on Hands
 - Red on Right, Blue on Left
- Control pitch range with input switches
- Calculate linear scale for pitch
- Harmonizer system
 - Rounds note to closest note that matches the harmonics of background music. (FFT)

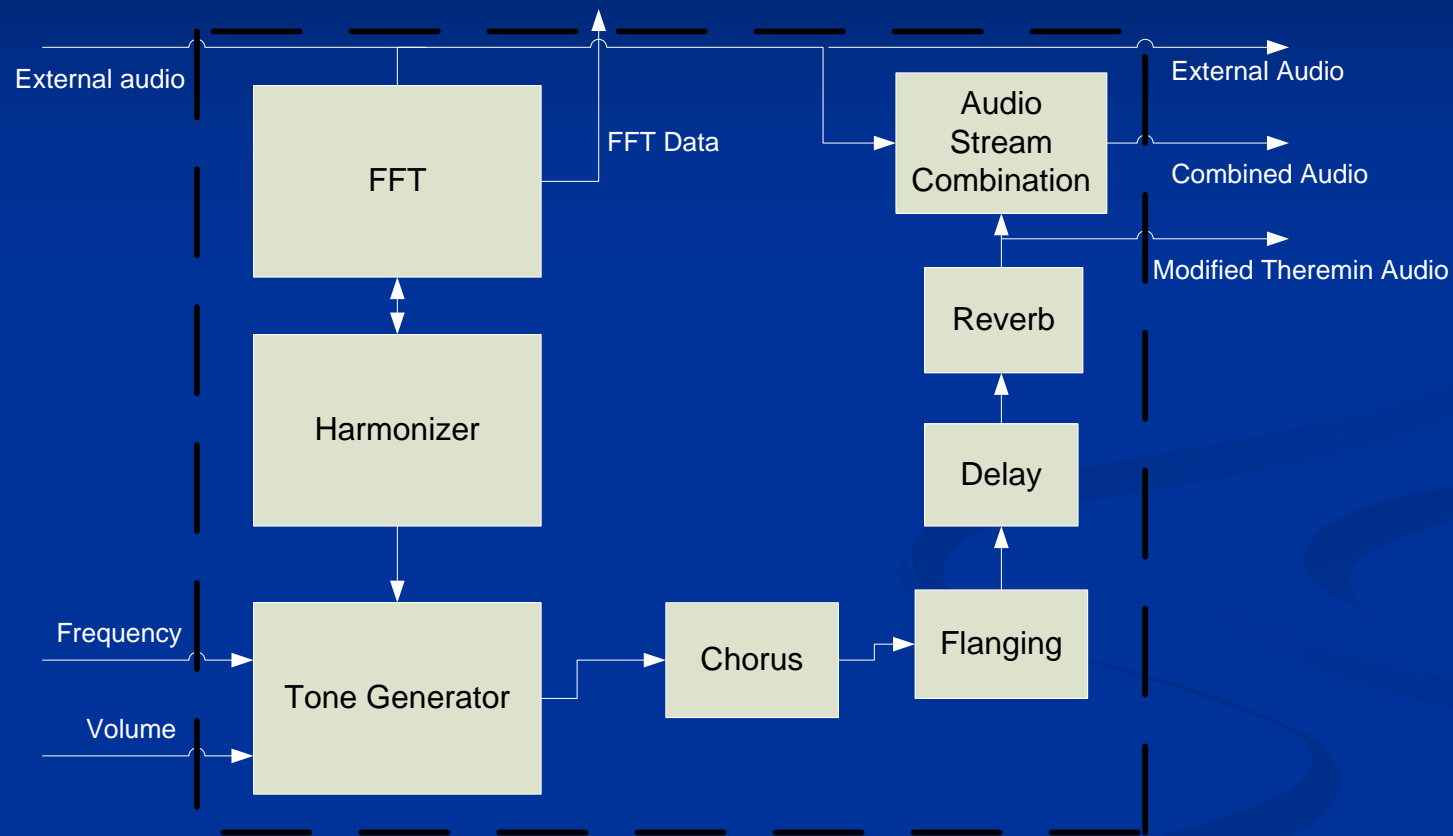
Block Diagram: Video Processing



Block Diagram: Video Processing



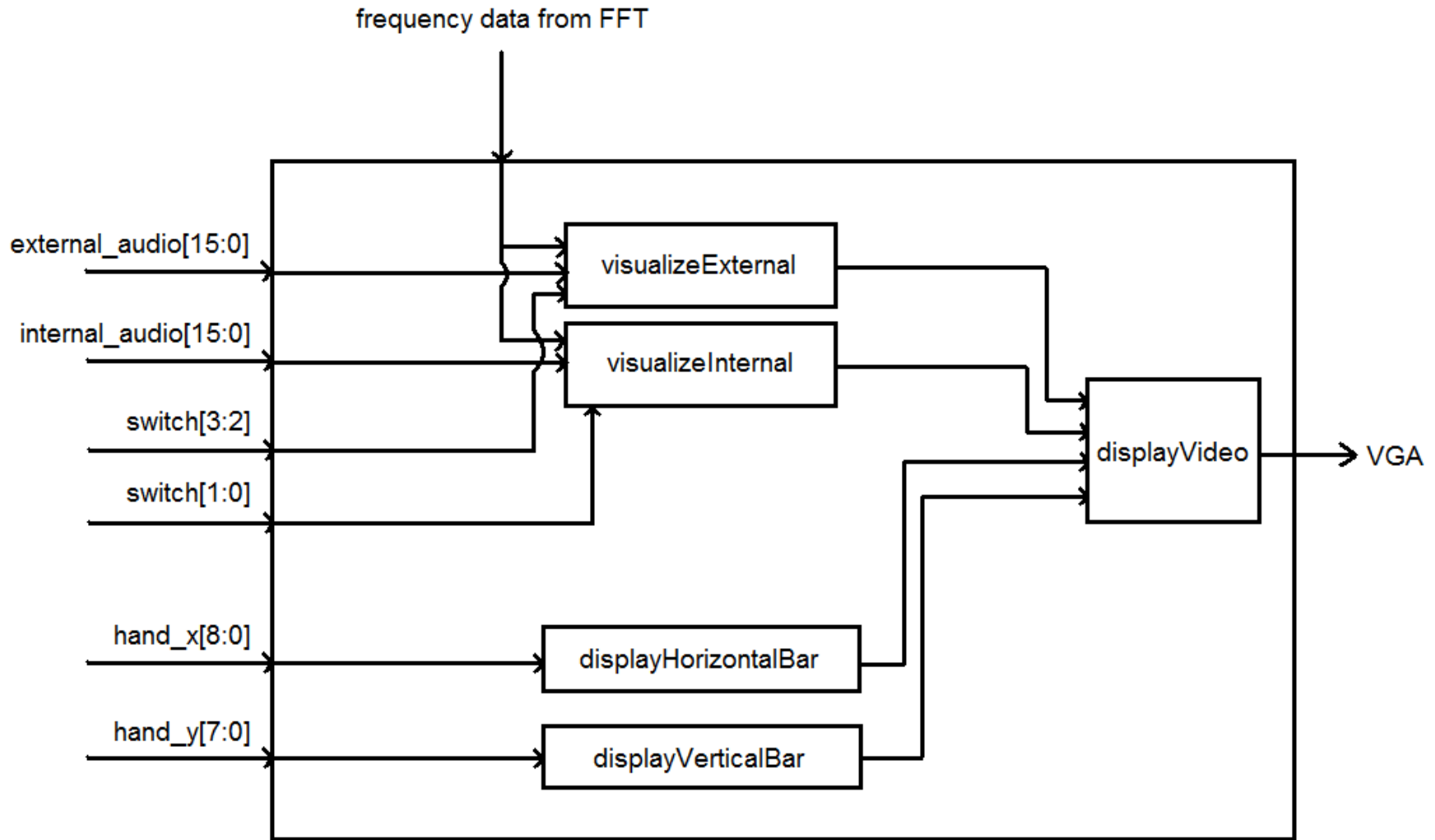
Block Diagram: Audio Effects



Audio Effects Module

- Responsible for generating audio given frequency and volume from Theremin
- Four different effects can be added to Theremin audio – Chorus, Flange, Delay, and Reverb
- Computes FFT of external audio and stores result in BRAM
- Outputs combined audio stream to AC97
- Advanced Theremin tone generation
 - More realistic tones through additive synthesis.

Block Diagram: Visualizer



Visualizer Module

- Hand positions will be displayed as vertical and horizontal bars on left and bottom of screen, respectively
- The visualizations will take into account the source of the audio (external vs. Theremin), and will also utilize stereo data (left vs. right speaker)
- Extra feature, if time permits: Save screenshot into BRAM for later retrieval

Visualizer Module, cont.

- Three different modes that the user can select from:
 - Mode 1: “Fire”
 - Uses frequency data from the FFT, pitch varies horizontally, intensity varies vertically, has pixels that gradually drop from the peaks
 - Mode 2: “Kaleidoscope”
 - Primarily depends on volume, rhythmically jumps from one configuration of shapes to another, symmetrical around the center
 - Mode 3: “Rainbow Ladder”
 - Primarily depends on pitches detected, vertical position changes with pitch, color changes with time (duration of note)

Timeline and Milestones

<u>Feature</u>	<u>Owner</u>	<u>Date</u>
Bare-minimum working system	All	11/18
Basic audio effects (e.g. echo)	drodgers	11/25
Displaying hand position on screen	jchang1	11/25
Color detection, frequency scaling/centering	aspicer	11/25
Having a functional FFT	drodgers	12/2
One mode with simple visualization	jchang1	12/2
More complex audio filters, or sampling	aspicer	12/2
Additional audio effects (e.g. instruments)	drodgers	12/9
Multiple modes, more complex visualizations	jchang1	12/9
Harmonizer (discretizing input to match)	aspicer	12/9
Final report and presentation	All	12/12

Thank you!



Questions?