Alex Sanchez & Behram Mistree
6.111 Final Project Proposal
11.3.2006
Coordinate TA: Javy

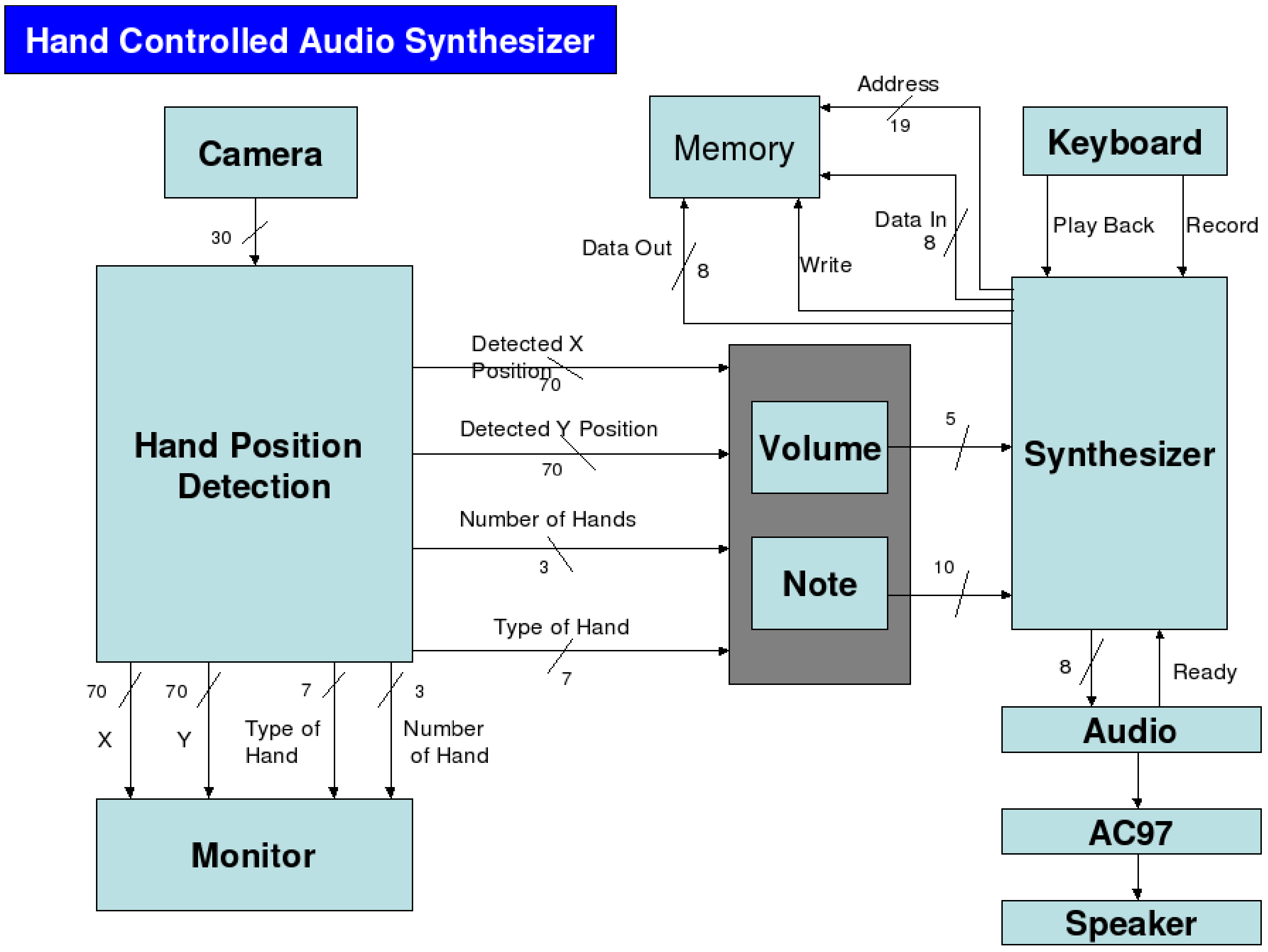
Hand Controlled Audio Synthesizer

1. Proposed Functionality

We propose to build a device that incorporates a video camera that records a person moving with LED's on his/her hands. Our device will process the real-time images received from the video camera and extract the x and y coordinates of each LED in the camera's field of view. Our device will do two things with these coordinates: (1) give feedback to the user by displaying the position of each LED on a computer monitor, and (2) map each LED's x-y coordinate to a musical note and volume.

An audio synthesizer will take these note and volume values and generates the audio waveform corresponding to a piano's playing those notes at those volumes. This waveform will be sent to the AC 97 Codec on the 6.111 Labkit which in turn will output the sound to a set of speakers.

In addition, the device will allow the user to record and playback his/her generated waveforms, thereby allowing for the superposition of multiple sounds as an ultimate output.



2. External Devices and Interfaces

1. Camera: Acquired from the 6.111 parts room. Hardware interfaces directly with board, but we will need to develop a module to request and process the inputs of the camera.
2. LED's (including power supply): We will likely acquire LED's from the 6.111 Lab or Radio Shack. Likely powered by 1.5 V Triple-A batteries. Each LED circuit will contain an on/off switch and a resistor to regulate current.
3. Speakers: We will likely use a pair of inexpensive computer speakers plugged directly into the headphone jack of the 6.111 Labkit.
4. Keyboard and Monitor: We will use available monitors and keyboards from lab.

3. Description of Individual Modules

3.1 HAND POSITION DETECTION Module

The camera returns a 30 bit number that represents a horizontal line's YCrCb values. After converting this line to RGB values, we will scan the line looking for a long stretch of red values (corresponding to a red LED) or a long stretch of green values (corresponding to a green LED). If we encounter such a stretch, we find and store the x-y value that corresponds to the center of that stretch. If we encounter future stretches of red or green colored LED's, we only record their center x-y values if those values are substantially different from those already recorded (for instance, at least 40 pixels away from the x center and 40 pixels away from the y center).

The HAND POSITION DETECTION module will send a three bit signal called NUMBER OF HANDS DETECTED to the NOTE and VOLUME modules. This signal will represent the number of different LED's detected by the camera (ranging from a minimum of zero to a maximum number of seven LED's detected).

As mentioned above, the x-y coordinates of each LED will be recorded. Each of the LED's can have an X position that ranges from 0 to 1023 (a number that will likely be reduced by the view of the camera). Therefore we will be able to encode each X position of each LED with a 10 bit number. Because we want to be able to track up to seven distinct LED's, we must pass a 70 bit number that corresponds to X positions from the HAND POSITION DETECTION module on to the NOTE and VOLUME modules. A similar sized number indicates the Y values.

The TYPE OF HAND signal shown in the block diagram will distinguish whether the x-y values passed are from a red LED or a green LED.

3.2 VOLUME Module

The VOLUME module converts the x-y position of a hand into a number representing the volume of the note being played. The volume of the note can take on 32 values from 0-31 where 0 is completely silent and 31 the loudest. The volume level is computed based on the value of the y coordinate of the hand. The minimum value of the y coordinate will be zero (the bottom of the screen) and the maximum value will be at the top of the screen. The volume level is proportional to the value of the y-coordinate. The highest volume levels are associated with hand positions located at the top of the screen and the lowest volume levels are associated with hand positions located at the bottom of the screen.

3.3 NOTE Module

The NOTE module converts the x-y position of a hand into a number representing a musical note. There will be 26 possible notes (2 full octaves). The x-axis of the monitor represents the note being played and the y-axis represents the octave the note is in. The left half of the monitor is the lower octave and the right half is the higher octave. The notes will be arranged in chromatic order with the note A being at the bottom of the monitor and G sharp at the top of the monitor.

3.4 KEYBOARD Module

The KEYBOARD module generates control signals for the synthesizer. The KEYBOARD module takes as input keystrokes from a computer keyboard. The control signals generated from this module are playback and record.

3.5 MEMORY Module

The MEMORY module is used for storing audio signals generated by the synthesizer. The MEMORY module will be implemented using the 6.111 Labkit's ZBT memories. Each ZBT is 512Kx36. Noting that the 64Kx8 BRAM from Lab 4 stored 11 seconds of data, we calculate that each ZBT is capable of storing 6 minutes 40 seconds of data.

3.6 SYNTHESIZER Module

The SYNTHESIZER module generates audio signals representing musical notes based on the x-y coordinate of the hand. After the VOLUME and NOTE module convert the x-y position of a hand to a volume-note pairing, the SYNTHESIZER creates an audio signal representing that how that note would be played on a musical instrument, in this case a piano. The SYNTHESIZER module uses subtraction synthesis to generate the audio signal. Subtraction synthesis consists of generating two oscillator signal that contain harmonics, changing the tone of the signals via pulse modulation , combining the oscillators at certain volumes, passing the combined signal through a voltage controlled amplifier that is connected to an ADSR envelope, and finally passing the signal through a low pass filter. ADSR represents the action, delay, sustain and release of a note played on an instrument and the envelope is used to produce a waveform that is similar to the expected waveforms produced by an instrument. When the user presses the record button, SYNTHESIZER will store any generated signals in MEMORY until the users presses the record button a second time to stop recording. Pressing the play button will play back the stored audio. The SYNTHESIZER can still create and record audio while stored audio is being played so it is possible to create more complicated loops of music.

4. Testing

To test the camera module, we will move LED's in front of the camera and visually inspect whether their positions are correctly recorded to the monitor.

We will do the majority of testing for the NOTE and VOLUME modules using Modelsim.

In order to reduce testing and debugging headaches, all of the equations that govern the SYNTHESIZER module will first be written and tested in MATLAB. The validity of these equations will be determined in two ways. The first method for determining validity is to superimpose the MATLAB generated signal on top of an actual recording of a piano. The second method for determining validity is to listen to the MATLAB generated signal and an actual recording of a piano. If the two sounds are indistinguishable and the squared difference between the two waveforms is small, we will declare the equations successes and program them directly into the FPGA with Verilog.

5. Schedule

We propose to the following schedule:

11.3.2006:	Several LED hand devices built.
11.11.2006:	HAND POSITION_DETECTION module completed (Camera capable of extracting x-y coordinates of LED's).
11.11.2006:	MONITOR module completed (x, y coordinates can be displayed to monitor).
11.13.2006:	Code for KEYBOARD module (found online) is incorporated into project. Code for interfacing to AC 97 codec (also found online) is incorporated into project.
11.15.2006:	Decision on methodology for synthesizing notes finalized.
11.17.2006:	NOTE and VOLUME modules completed and tested.
11.22.2006:	Interface to ZBT for playback and recording completed and tested.
11.29.2006:	SYNTHESIZER module completed and tested.

****Debug, Debug, Debug****

12.7.2006:	Work on possible extensions. (See final section of proposal.)
12.11.2006:	Report completely written.

6. Division of Labor

Behram:

1. Build LED hand devices;
2. Build HAND POSITION DETECTION module;
3. Build VOLUME module; and
4. Incorporate KEYBOARD.

Alex:

1. Build MONITOR module;
2. Build NOTE module;
3. Incorporate AUDIO CONTROL; and
4. Build SYNTHESIZER module from equations.

Work together on:

1. MATLABBING of synthesizer;
2. Extensions (see final section of proposal); and
3. Final report write-up.

ZBT interface will be coded by whoever has more free time.

7. Potential Extensions

The schedule calls for us to be finished with all the desired functionality before the final project is due. *If* we are on schedule, we hope to add on additional functionality. Potential ideas for this additional functionality are listed below:

1. <u>Sound Effects:</u> In researching how to synthesize sounds, we observed that there were several other effects that we could incorporate into our project. Examples include an echo and slowing or speeding up the sound during playback.
2. <u>LED Control Tower:</u> Our synthesizer can be controlled by any LED movement. Therefore, if we build a device that turns sequences of LED's on and off, the synthesizer should play notes that correspond to those LED sequences. If we make this LED device controllable by circuitry, we should be able to play complicated music precisely.
3. <u>Voice In:</u> Our synthesizer allows users to make music using body movements. It would be nice to add a voice element
4. <u>More Instruments:</u> Our lab report calls for only synthesizing a piano. If our efforts are successful, we may be able to add on other instruments such as guitar, saxophone, and bass guitar.
5. <u>Display:</u> Any musical instrument is difficult to learn. If our synthesizer proves particularly non-intuitive, we may make a more sophisticated display that makes our synthesizer easier to play.