6.111 Final Project Proposal

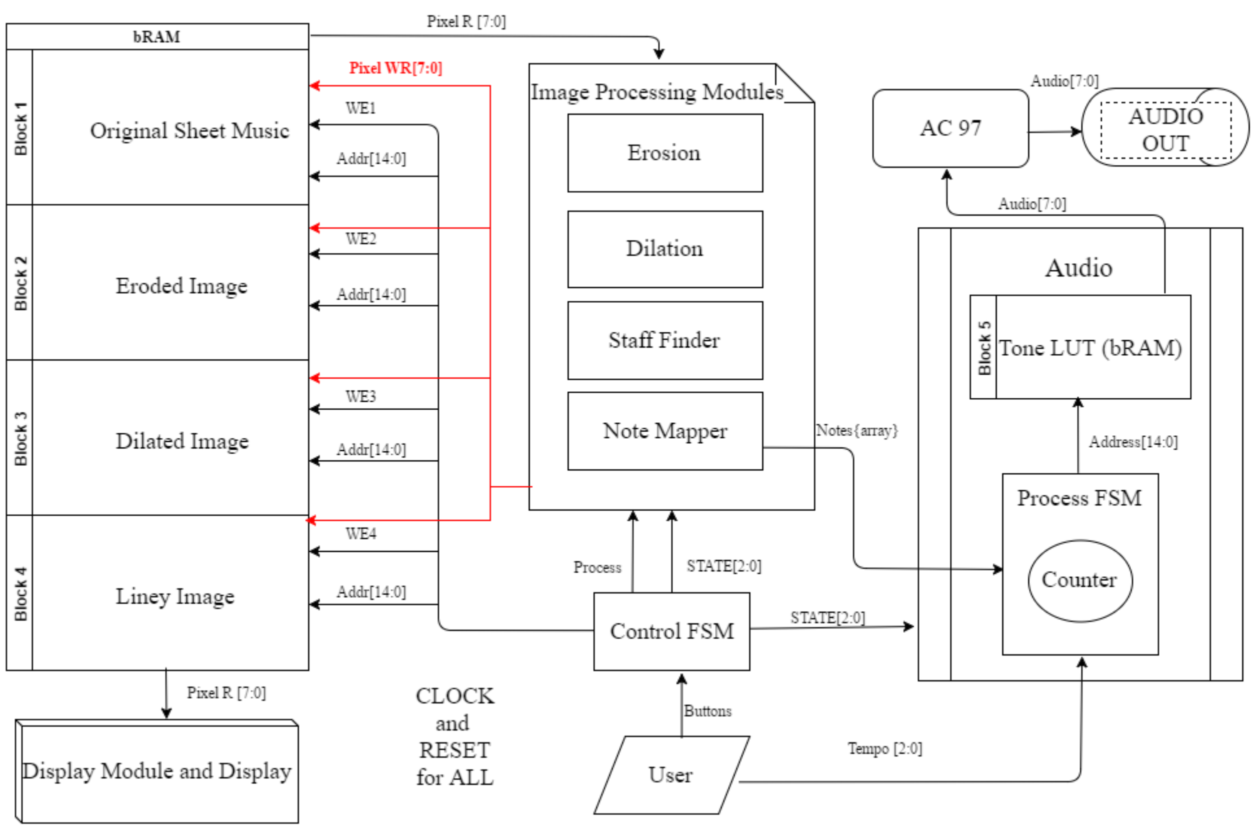
Mark (Yuechen) Yang & Henry Love

1 Summary

Being able to hear what is on sheet music is very helpful to musicians beginning to learn a piece of music. Having auditory input can help people learn notes and rhythm faster and correct current mistakes. To make the transfer of sheet music to sound convenient, we propose a digital music reading machine. This project will take a picture of a sheet music, process the image, and play the notes back to the user. There are two main parts to this project; note recognition and audio playback. This project will start with the reading of simple rhythms, notes and key signatures and if time permits, this project will evolve to read a score, notes that do not lie on the ledger lines, and music with accidentals. A user interface will allow the user to input the tempo of the piece before sight reading occurs.

2 High Level Perspective

2.1 High Level Block Diagram



High level block diagram for the system

2.2 Design Implementation

Our digital sightreader will take an image of music stored in bRAM, process it, and allow the user to listen to a digitized interpretation of the score. We will begin with simple music, consisting only of quarter notes and no sharps or flats. With this in mind, determining the vertical and horizontal position of the note on the staff will be enough to fully characterize the pitch and where that note fits in the music relative to the other notes.

3 User Inputs

3.1 Staff Detection

Staff detection input gives the staff detection module data to use to determine the spacing of the staff lines. The user specifies a column of pixels where nothing but the staff intersects this column. The column is a one (or two) pixel wide line that runs vertically down the display and can be position left or right via the left and right buttons on the lab kit.

3.2 Tempo

Tempo is a 9 bit binary number that ranges from 0 to 246. Tempo is a specification of how much time to allocate to the note value specified by the time signature (most commonly a quarter note). Traditionally, tempo refers to beats per minute (bpm) and thus a tempo set to 100 will result in one quarter note played every 1/100th of a minute if the time signature dictates a quarter note as the note to get the beat. The user will be prohibited from setting the tempo to 0 as this will result in a never ending piece!

3.3 Buttons

5 buttons allow to the user to cycle through the control FSM and control other inputs, such as the staff detection input. Buttons left and right move the staff detection line in the respective direction. Enter on the first press begins processing of the image loaded into bRAM memory and enter on the second press begins music playback. Button 1 on the labkit acts as system reset, and captures another image of music to be processed.

4 Module Descriptions

4.1 Staff Finder

The purpose of this module is to determine where the lines of the staff lie on the image of the sheet music and to determine the spacing between each adjacent line. The staff finder module takes the staff detection user input data (vertical column of pixels) and determines the vertical spacing of the staff lines by calculating the number of white pixels between black pixels.

4.2 Erosion

The erosion module will be responsible for eliminating the staff lines of the sheet music. Erosion is one of two fundamental operations in morphological image processing (dilation being the other) and assigns the color white to any pixel not entirely surrounded by black pixels. Preliminary note detection will happen when the staff lines have been eroded (or removed) and before note heads have been entirely removed.

4.3 Dilation
The dilation module functions the opposite way as the erosion module. This thickens the lines and the notes. The main usage for this is to reconstruct the notes after erasing the lines.

4.4 Note Mapper
In music, horizontal location correlates to time and vertical position correlates to pitch. Note mapper takes the eroded image provided by the erosion module and determines the intersection between the note heads and the staff lines specified by the staff detection module. Knowing the vertical position of each note head provides information on what tone to play. Knowing which note is to the right of the note just processed provides information on when to play each note. It is the note mapper's job to take each blob of black pixel from the erosion module and map it to a tone with a characteristic pitch and duration. For proof of concept, each note will be assumed and treated as a quarter note. If time permits, a better note detection algorithm will be implemented to detect different types of notes to allow for more complex music playback (see section 7).

4.5 High Level Control FSM
This FSM will interact with all functional blocks and contain states that set up the correct parameters for the image processing modules, the audio interface, the saved frames, and the display. Upon receiving input from the user, the FSM will send out signals to the audio module to set up the correct tempo. It will also send the current STATE information to the processing modules so image processing can be conducted. Another function of the control FSM is to provide write enable signals to the bRAM blocks holding different frames, so that they can be updated from the processing module output.

4.6 bRAM containing saved images (Frames)
There will be 4 different frames saved in the bRAM that can be toggled by the switches on the lab kit. The original image is first saved, this is kept as a reference for all the processing work. Then the eroded output and the dilated output each will be saved to their own frames. Note that by design, we allow processing of any of the saved frames. For example. we can dilate an eroded module to regain the size of the note heads.

5 Performance or hardware limitations if any

The current note detection algorithm will process a slanted image incorrectly as staff detection is performed using one measurement of the staff and from there, assuming staff lines are perfectly horizontal. In addition, if a camera is used to capture the image from a physical piece of sheet music (as opposed to a screen shot off a computer screen), noise will be introduced to the image. There will have to be some method in place to account and compensate for this noise. Camera detection could also lead to variable scale for different images. If the distance of the sheet music from the camera is allowed to be variable, rather than fixed at a specified distance, an additional module will have to be created to normalize the size of the image such that the notes are a recognizable size.

6 Schedule

	10/31	11/7	11/14	11/21	11/28
Bring online all provided modules					
Matlab proof of concept functional					
Implement memory structure in Verilog					
Implement erosion & dilation in Verilog					
Implement staff finder in Verilog					
Implement note mapper in Verilog					
Implement control FSM in Verilog					
Bring first version online					
Final Debugging					

Allow for different key signatures and cleff					
Implement camera vision					
Implement better note detection algorithm and account for different time signatures					

	Essential for proof of concept
	If time permits

7 Stretch Goals and Future Improvements

7.1 More user inputs

Future improvements include more user inputs for a more complicated note detection algorithm and reading of more complex music. Additional inputs include key signature, cleff, and time signature. Cleff dictates where the staff is centered and how the vertical position of the notes on the staff are mapped to pitch. Depending on the cleff, the note mapper module will pull pitches from a lookup table corresponding to the cleff. Key signature determines which notes are raised or lowered by a half step on the staff. Allowing the user to input key signature will allow our digital sight reading system to play a variety of music that is not just in the key of C major (no sharps or flats). Finally, time signature dictates how many beats are in a measure and which note is assigned to that beat. Time signature as an input could potentially lead to a better note reading algorithm as the rhythm in each bar could be checked against the time signature to ensure that all beats have been accounted for. If rhythm in the bar does not match the specified time signature, the note with the weakest correlation would be removed and replaced such that the time signature is satisfied (please see below).

7.2 Better note detection algorithm

Currently, notes will be detected as a clump of black pixels. This does not allow for the detection of different rhythms and thus any clump of pixels on the processed image is assigned the value of a quarter note. A proposed, and more complicated note detection algorithm is one that assigns a weight/numerical value to each note. A vertical sweep will occur over each line, integrating the number of black pixels that fall in each pixel column. Graphing the number of black pixels in a column versus horizontal position will produce a graph that has peaks where the black pixel count per column is high and dips where the black pixel count per column is low. Determining the characteristic pixel profile of each note and matching this to the processed/collected data could potentially allow the FPGA to determine which note is where and what time value to assign to it. Summing these profiles together may also lead to the characterization of chord structures and the ability to recognize such features, a very interested problem that could be addressed in the distant future!