6.111 Final Project Proposal

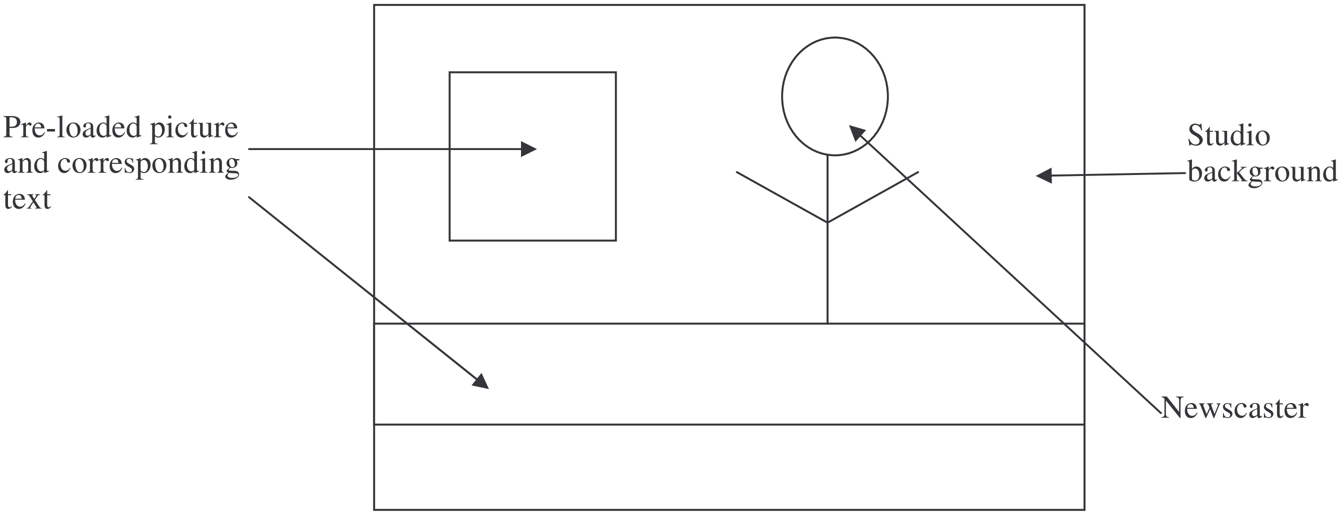
Obinna Anyanwu and Jyotsna Venkataramanan

Overview

Our final project involves a do-it-yourself newscast. It requires a main feed that has a person standing in front of a distinguishable screen and preloaded pictures and corresponding captions. The user is also allowed basic control of the system including changing between different picture and captions and also making the picture become fullscreen to include more detail. This system allows anybody to easily create and manipulate a newscast.

Project Description

Our final project involves building a virtual newscast. This allows a person to insert several pictures and corresponding captions into the system, have a live feed which features a newscaster, insert a studio background and be able to display all of this on the screen in a news format.



The user can change between the preloaded pictures and caption in real time by using the buttons on the labkit. The user is also allowed the option of expanding the preloaded picture in the box to become fullscreen. An expansion to the project that we hope to do is allow a live feed when the box expand to a fullscreen view.

Block Description

This project can be separated into four main modules: the input, the video processing, the control and the output. The video processing blocks puts all the parts of the visible output together. This involves the separation of the newscaster and the background and then putting the picture and caption together with it. The control module uses the preloaded picture and caption as an input. It allows the user to control which picture and caption is loaded and allows them to change them at real time. It allows the picture or second live feed to expand and become full screen. The output module outputs all of this onto a screen.

Input

The video acquisition and processing block will consist of several blocks. The Video Acquisition block will receive data from the camera and handle the interface between the ADV7185 NTSC Decoder chip, which will output 24-bit YCrCb video data. This module includes submodules for the generation of chip initialization signals as well as for the generation of vertical and horizontal synchronization signals.

Video processing

The Frame Buffering block will be responsible for the storage of a single video frame. This frame is necessary to calibrate the video system for chroma keying. The size of the frame buffer will be dependent on the resolution of the incoming video signal, but it is unlikely that the BRAM on the FPGA will be sufficient for this purpose. As a result, this module will interface with the labkit ZBT SRAM, and will be responsible for those control signals. Depending on the set of control signals received from the control block, the module will either store a frame and send it to the Chroma Keying Module or simply send the video signal directly from the Video Acquisition block.

The Chroma Keying module is where the bulk of the video processing will take place. Chroma keying is a process where two images are blended and is commonly known as blue screening. The module will consist of submodules for the initial system calibration and the image overlay. The calibration will be done by finding the average YCrCb values across an entire frame (using data taken from the frame buffer). When a person (or object) is placed in front of the camera, differences between the camera YCrCb values and the averaged values will indicate where image data should/should not be displayed.

Next is the Image Composition module, which will be responsible for the generation of the composite video signal to be output. This includes not only the chroma keyed signal, but the text headlines as well as the images for the news headlines. Again, depending on control signal values, it will be possible to change the images and text displayed on the screen.

Control

The control module includes two submodules. The first submodule stores the preloaded picture and caption and relates them to each other. The second submodule includes the FSM that allows the user to change the picture and caption or make the picture or second video screen become fullscreen. There is a possible third submodule that is the GUI that the user can easily interact with. If not, buttons on the labkit can be used to carry out the same function.

The first submodule is the preload submodule. The picture and corresponding caption have to be stored in a way in which they can relate to each other. The text can just be done in ASCII encoding but the picture is a little more complex to load into the FPGA. The picture has to be converted with Matlab into a .coe file and then inserted into the memory of the FPGA.

The second submodule is the real time control module that allows the user to change the picture and corresponding caption and also make the picture or the second live feed become fullscreen. This will be an FSM that takes in inputs of buttons and switches on the labkit. It will output addresses to the Image composition block.

Output

The final module, Video Output, will be responsible for generating the necessary synchronization, blanking, and initialization signals necessary to operate the ADV7125 Video DAC. The module will also convert the YCrCb data into RGB format before it is sent to the DAC, and the result will then be displayed on a VGA monitor.

Testing

Video Processing

The modules in the block have been designed so that they can be tested independently of each other. Since it will be necessary to see the video outputs of subsequent modules, the testing of the input and output blocks should come first. In parallel with this, the image composition module can be tested relatively simply by overlaying images and text over a known signal (such as color bars). The ZBT interface (Frame Buffer module) can be tested by creating a memory tester module. The Chroma-Keying module can be tested initially by setting an arbitrary YCrCb value to overlay the image/keep the camera input. Again, a known signal can be used as the overlay to ensure correct operation.

Control

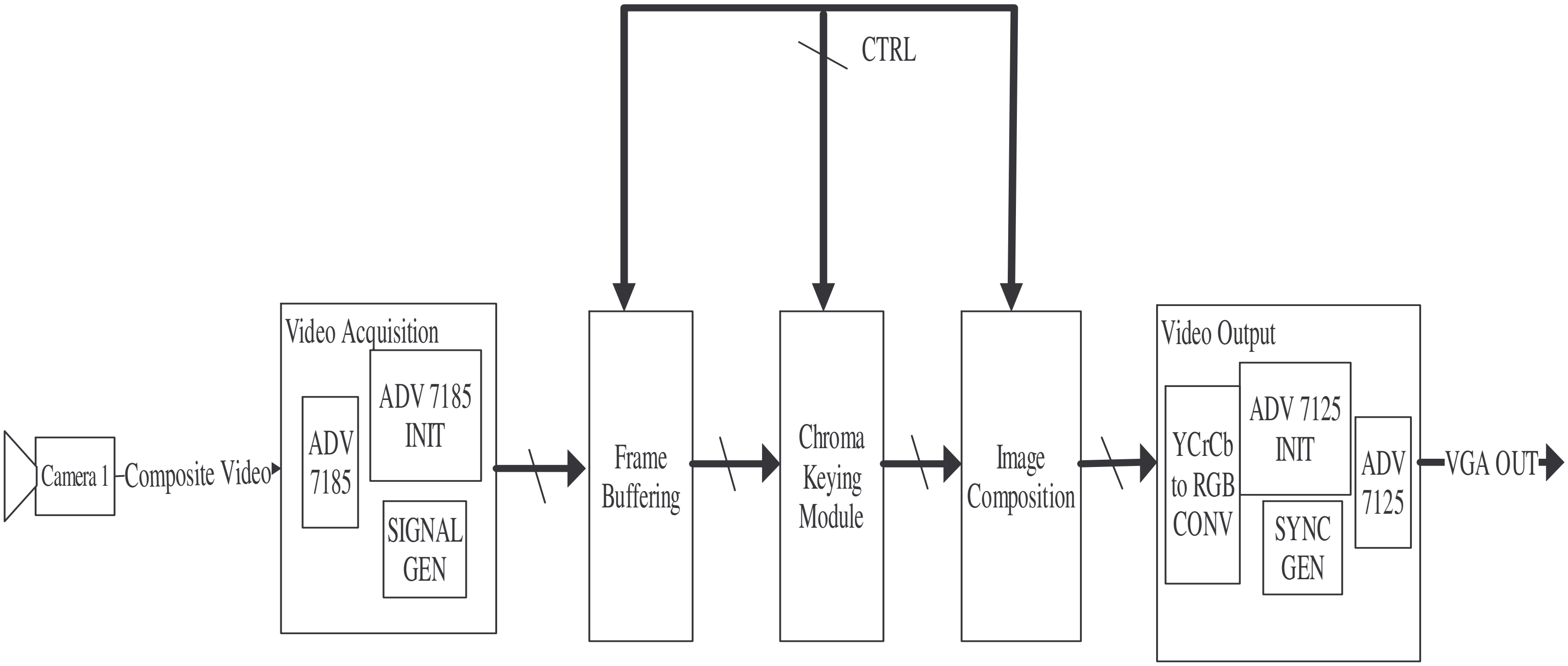
The control block can be tested incrementally. The first part will be getting the pictures and text preloaded into the memory. This involves the use of Matlab and converting all of this to the appropriate file format. This can be tested visually. The real-time signals that control the look of the screen are easily tested because they are implemented in an FSM. Each individual input and output of the FSM can be tested.

Incremental Improvements

Our initial goal in our implementation is to build a system with the following basic functionality and to improve upon features as the project progresses. This project has many places into which we can expand depending on the time available. These improvements include the addition of a second camera to facilitate a real-time feed. Since this will require the use of a second labkit, there will be issues with communication and timing, so a simple communication protocol will need to be developed to counter those effects. In addition, we plan on adding multiple display control options to the Image Composition module. Of course, this will all be done after we have a working model and as time permits. A third possible improvement is adding a graphical interface and allowing the user to interact with the system much more easily.

Division of Labor

Jyotsna will work on the control and output modules, while Obinna will handle the video input and processing.



Preliminary Video Processing Block Diagram