

6.111 Final Project

Status Report

Logan P. Williams
José E. Cruz Serrallés

21 November 2011

1 `projective_transform` (Logan)

1.1 Status

This module is completely written and almost completely tested. Tested functionality includes correctly calculating distance and iterator incrementors, requesting pixels at a rate of one per clock cycle, and correctly outputting coordinates for skewed pixels. A test bench still needs to be written that tests the modules ability to respond to delays in `memory_interface` and to “unexpected” new frame events.

1.2 Possible Issues

There might be some timing issues with the dividers and square rooters, as each has a long latency (20 and 24 clock cycles respectively), so that they are pipelined correctly at the end of each line.

2 `memory_interface` (José)

2.1 Status

The first draft of this module is nearly complete. The following functionality has been included: location shifting, handling of conflicting requests, and interfacing with memory. The only functionality left to implement is the buffering of read requests due to the two-cycle delay of the ZBT RAM.

2.2 Possible Issues

There might be some difficulty in handling multiple requests. Code has been written that should handle these requests appropriately. If this code does not work conceptually due to previously unconsidered properties of the ZBT RAM, the clock frequency of the RAM could be elevated slightly, so as to give the RAM more cycles to handle the different streams of memory accesses.

3 `object_recognition` (Logan)

3.1 Status

This module has not been started yet, although its implementation should be straightforward. `object_recognition` will just have to sum several values, and then use a divider module that has already been tested in `projective_transform`. To calculate downsampling coefficients, it will use the same `sqrt` module as `projective_transform` uses.

3.2 Possible Issues

No issues are foreseen with this module, as both of the tricky submodules, `divider` and `sqrt` have already been tested in `projective_transform`.

4 LPF (José)

4.1 Status

This module has not been started yet, although its implementation should be straightforward. LPF will simply pass along pixels to `projective_transform` until we will use the LPF.

4.2 Possible Issues

The only foreseeable issue with LPF is that we might not have time to complete it. Because it is not a requirement for functionality, work will only be started on it if time permits.

5 ntsc_capture (Logan)

5.1 Status

An initial draft of this module has been written, but is untested. This draft implements all functionality that was proposed, including storing color pixels, and recognizing specific colors. The next step is to write a test bench that will test this functionality.

5.2 Possible Issues

There may be some issues with correctly identifying the markers on the corners of the picture frame. This will likely take some experimenting to determine the best threshold values to use for color identification.

6 vga_display (José)

6.1 Status

This module has been adapted from the Lab 2 Pong code and awaits testing in lab.

6.2 Possible Issues

There are no real foreseeable issues with this module.

7 Timeline

Nov. 21 - ~~First draft of `projective_transform` completed. (Logan)~~

Nov. 22 - First draft of `memory_interface` completed. (José)

Nov. 22 - ~~Start of debugging of `projective_transform`. (Logan)~~

Nov. 23 - Start of debugging of `memory_interface`. (José)

Nov. 25 - ~~First draft of `ntsc_capture` written. (Logan)~~

Nov. 25 - `memory_interface` will be fully functional. (José)

Nov. 25 - `projective_transform` fully tested and functional. (Logan)

Nov. 26 - (Optional.) First draft of LPF completed. (Optional.) Start of debugging of LPF. (José)

Nov. 26 - First draft of `object_recognition` written. (Logan)

Nov. 27 - `vga_display` tested in lab without `memory_access`. (José)

Nov. 28 - (Optional.) LPF fully tested. (José)

Nov. 28 - `object_recognition` and `ntsc_capture` fully tested and functional. (Logan)

Nov. 30 - Partial integration of `ntsc_capture`, `memory_interface`, and `vga_display` begins.

Dec. 01 - Partial integration complete.

Dec. 04 - Full integration started. Start of debugging of integration.

Dec. 07 - Full integration finalized.

Dec. 08 - Project demonstration and videotaping.

Dec. 12 - Final report written.