projective_transform: processes a stream of incoming pixels, skewing, rotation, and scaling the image by generating new \((x, y)\) coordinates for each pixel corresponding to the four corners of the frame. (Logan)

- Correctly calculates distances and iterator incrementors, using the \texttt{sqrt} and \texttt{divide} submodules.
- Sends a signal to LPF to request new data when initial frame calculations have been done.
- Generate one new set of coordinates per clock cycle and transmit to \texttt{memory_interface}.
- Pipelines square root and division calculations so that there is no delay for each new line.
- Pauses appropriately when \texttt{memory_interface} cannot handle new data.
- Can handle “unexpected” new frame events.

object_recognition: average the \((x, y)\) tuples for each pixel that matches one of four Cr/Cb regions of interest. (Logan)

- Sums the coordinates of each color that it receives.
- Correctly averages each coordinate.
- Outputs the list of coordinates and a flag immediately after \texttt{ntsc_capture} has finished processing a frame and the \texttt{divide} submodules have finished their averaging operations.
- Output “fake” downsampling coefficients based on linear estimates of distance.
- (Time permitting:) Generate and output \(M_x\) and \(M_y\) downsampling coefficients after a frame has been captured.

memory_interface: efficiently interfaces with the memory and all of the modules that have to write to and read from ZBT memory. (José)

- Writes to memory data from \texttt{ntsc_capture}.
- Reads from memory an image to \texttt{vga_display}.
- Outputs to and captures data from \texttt{LPF}.
- Captures data from \texttt{projective_transform}.
- Shifts data locations when \texttt{ntsc_capture} starts providing a new image.
- (Time permitting:) Reads an image from flash memory and stores it in RAM for use as the transformed image.

LPF: applies lowpass filters, vertically and horizontally, on the image that is to be warped, in order to prevent aliasing at the output. (José)
(Out of time:) Just fetches pixels from memory and feeds them to `projective_transform`. LPF does not filtering.

- Loads appropriate filter coefficients based on the coefficients $M_x$ and $M_y$ from `object_recognition`.
- Reads data from memory vertically and horizontally, and has the necessary data for the calculation of each output pixel in its buffers.
- Mirrors the data appropriately in its buffers when processing near an edge.
- Outputs to `memory_interface` a pair of pixels that correspond to the convolution sum of the corresponding data.

`ntsc_capture`: process the incoming video stream and send pixels in sets of two to `memory_interface` (Logan)

- Capable of reading the incoming video stream from the video ADC.
- Can transmit pixels to `vga_display` for immediate display.
- Saves full color data.
- Lumps pixels into groups of two to transmit to `memory_interface`.
- Recognizes pixels matching specific regions of the Cr/Cb plane, and transmits that information to `object_recognition`.

`vga_display`: fetches data from memory and displays it on the screen. (José)

- Displays a predefined pattern on the screen.
- Requests a pixel one video clock cycle before it is needed.
- Reads an image from memory, through `memory_interface`, and correctly displays it on the screen.