6.111 Final Project Presentation

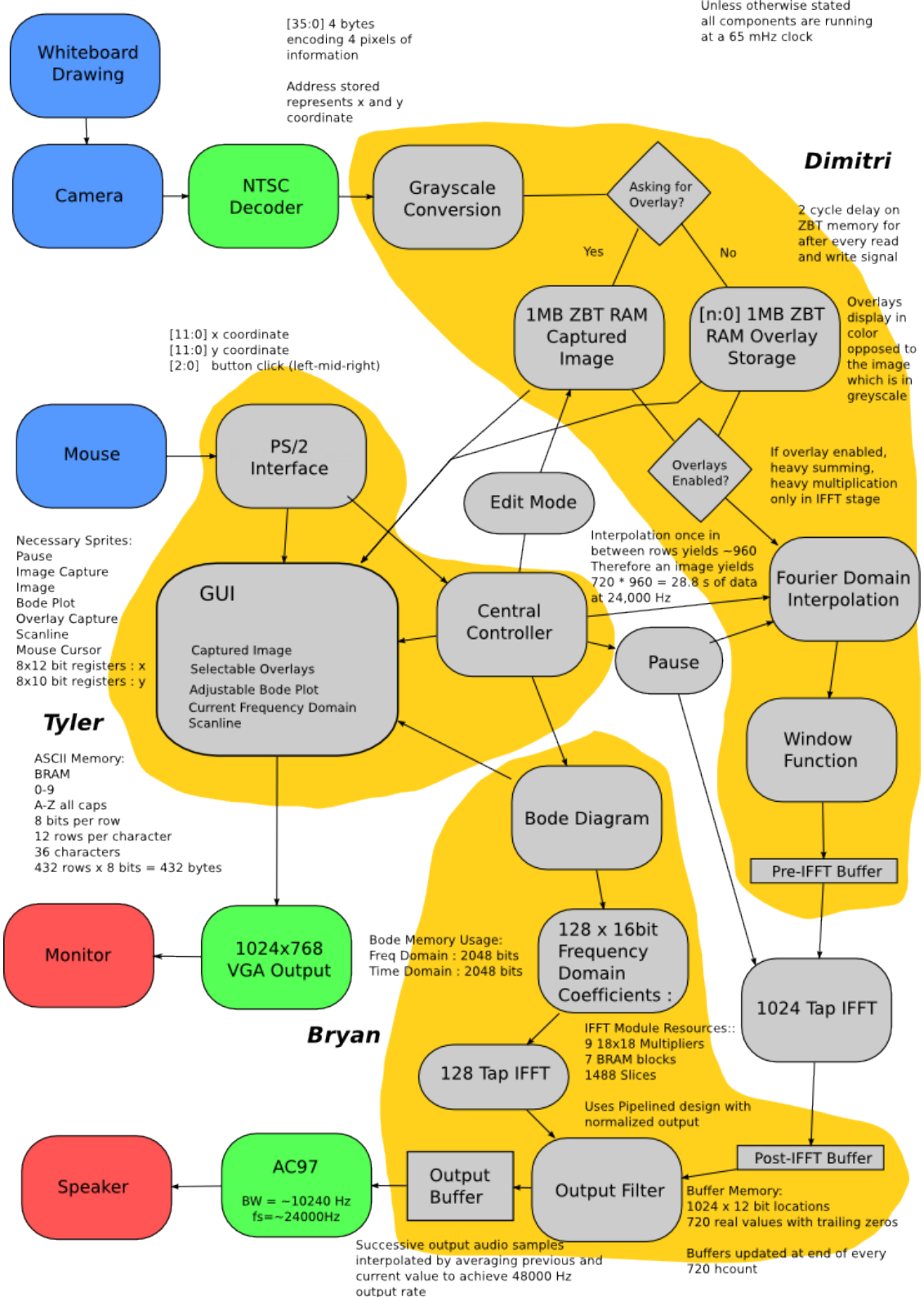
Real-Time Visual Audio Composition

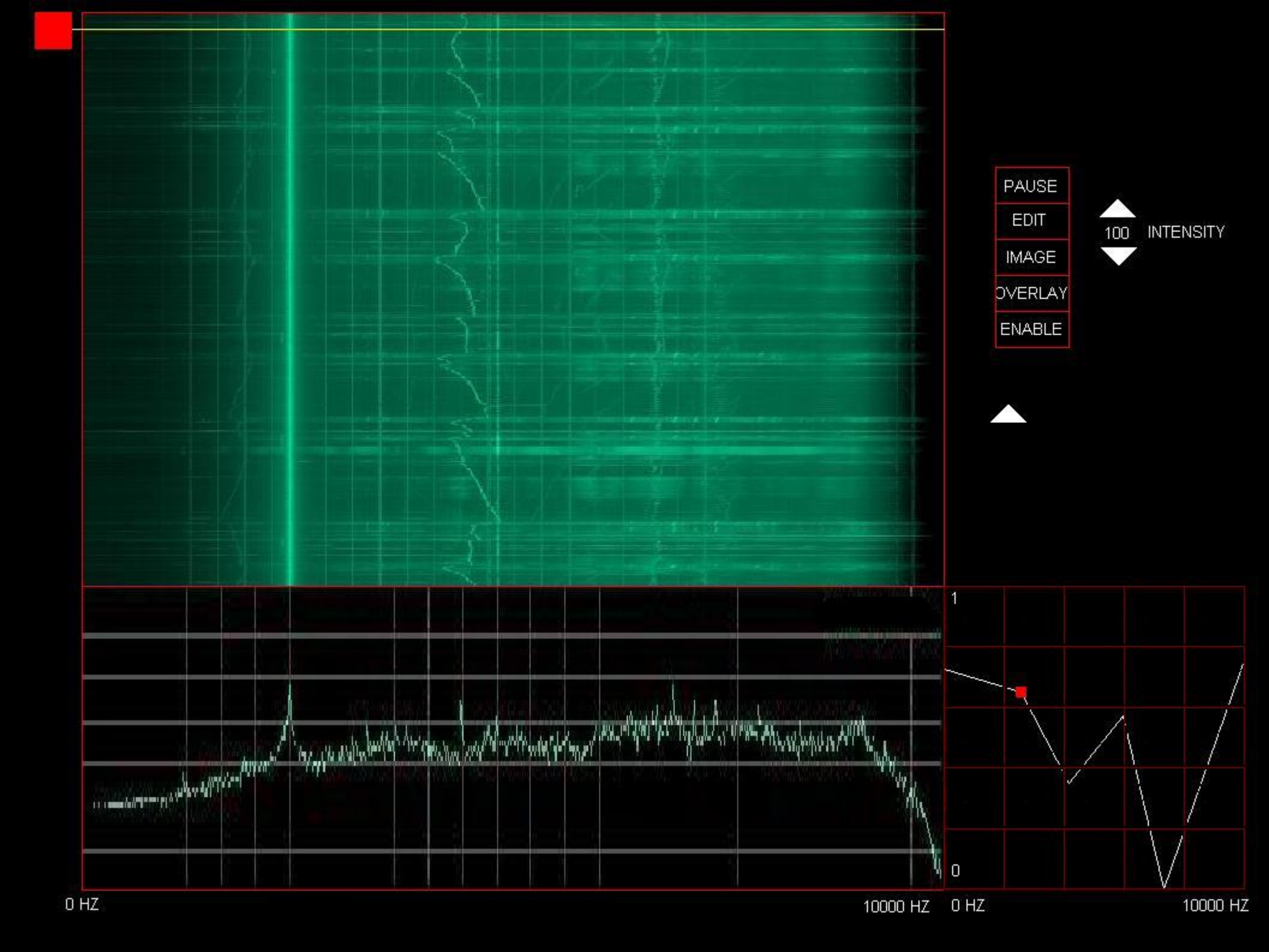
Tyler Hutchison, Dimitri Turbiner, Bryan Newbold

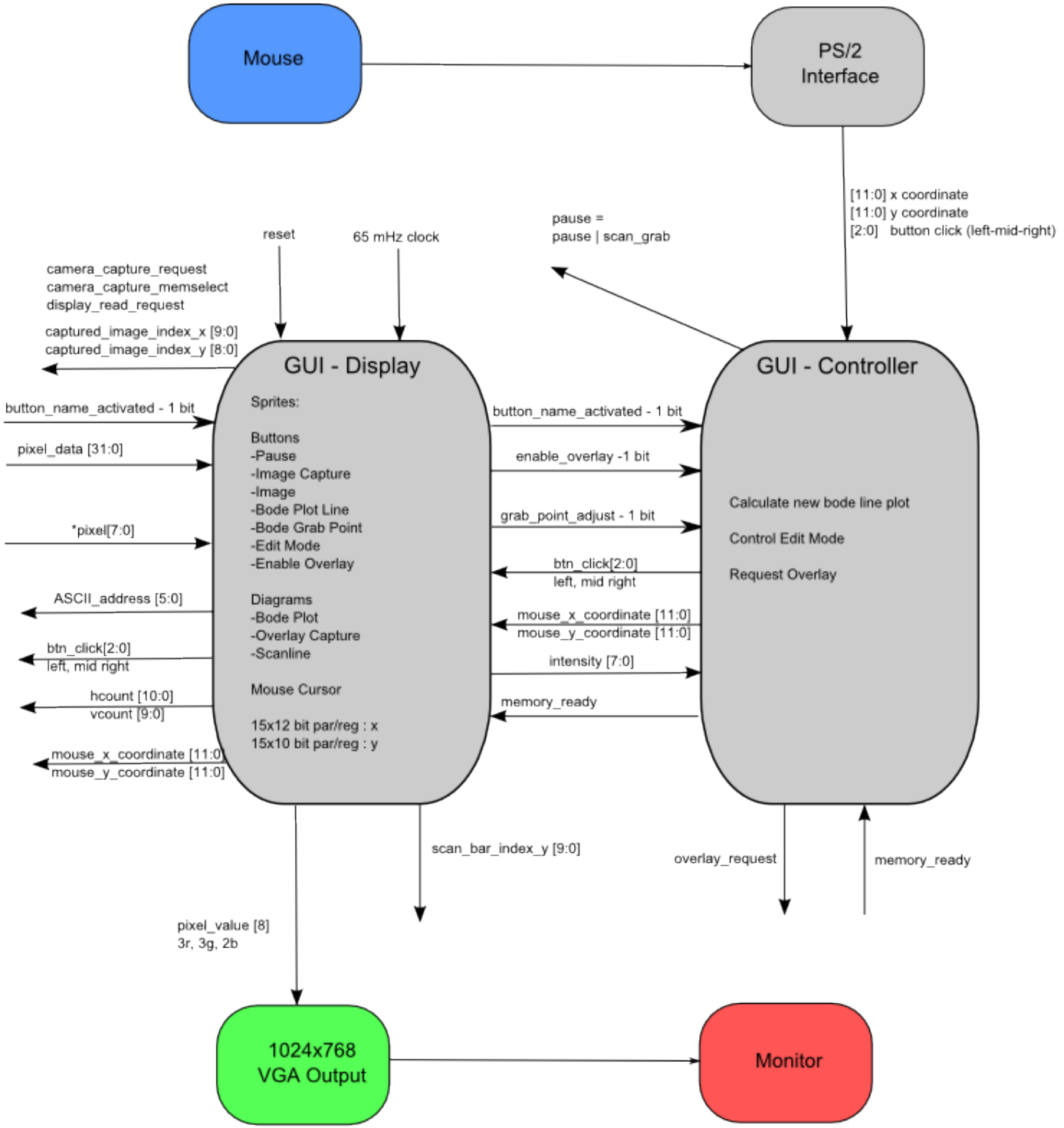
Project Breakdown

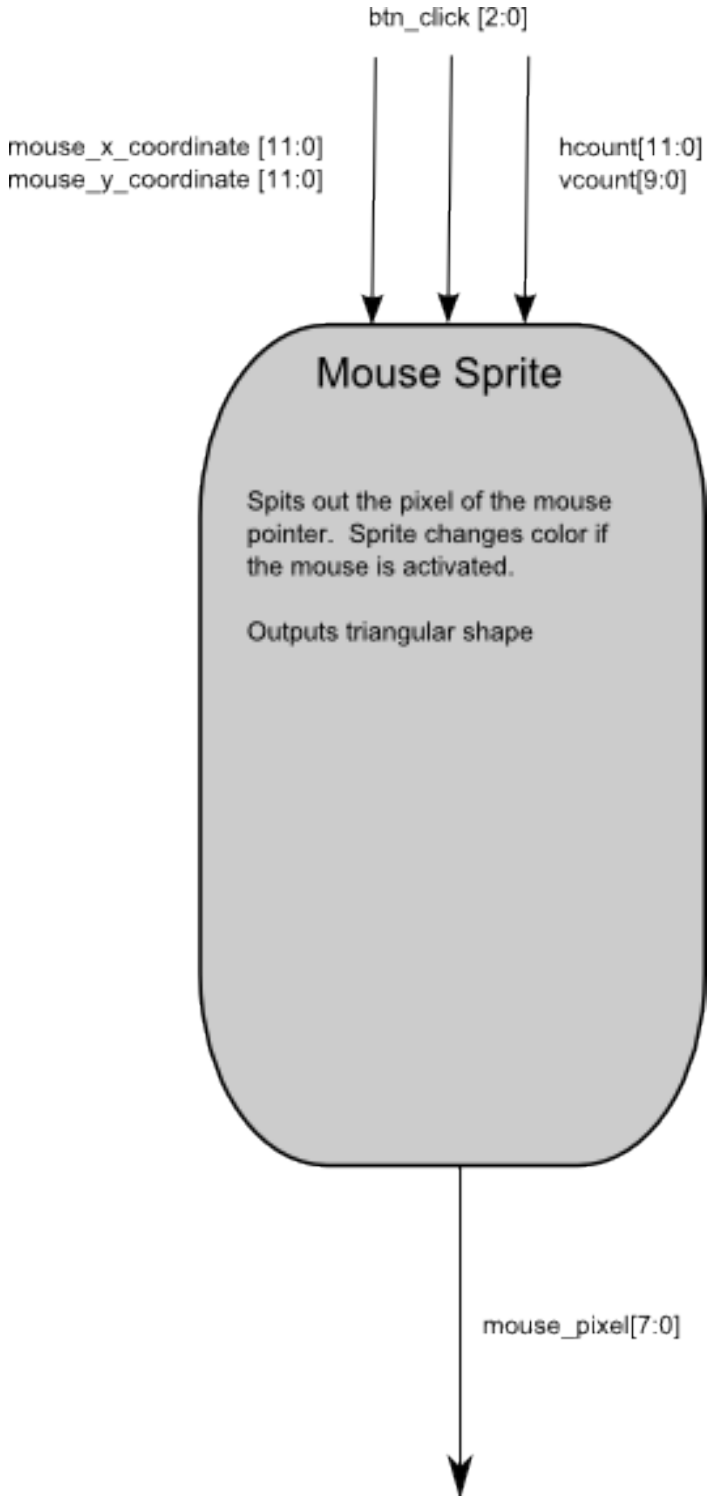
- Camera reads spectrogram, processed with IFFT module, output as looping audio
- **Tyler** is building VGA output GUI, including user editable filter diagram
- **Dimitri** is handling NTSC input and memory control
- **Bryan** is streaming spectral samples through IFFT, applying filter, and outputing audio

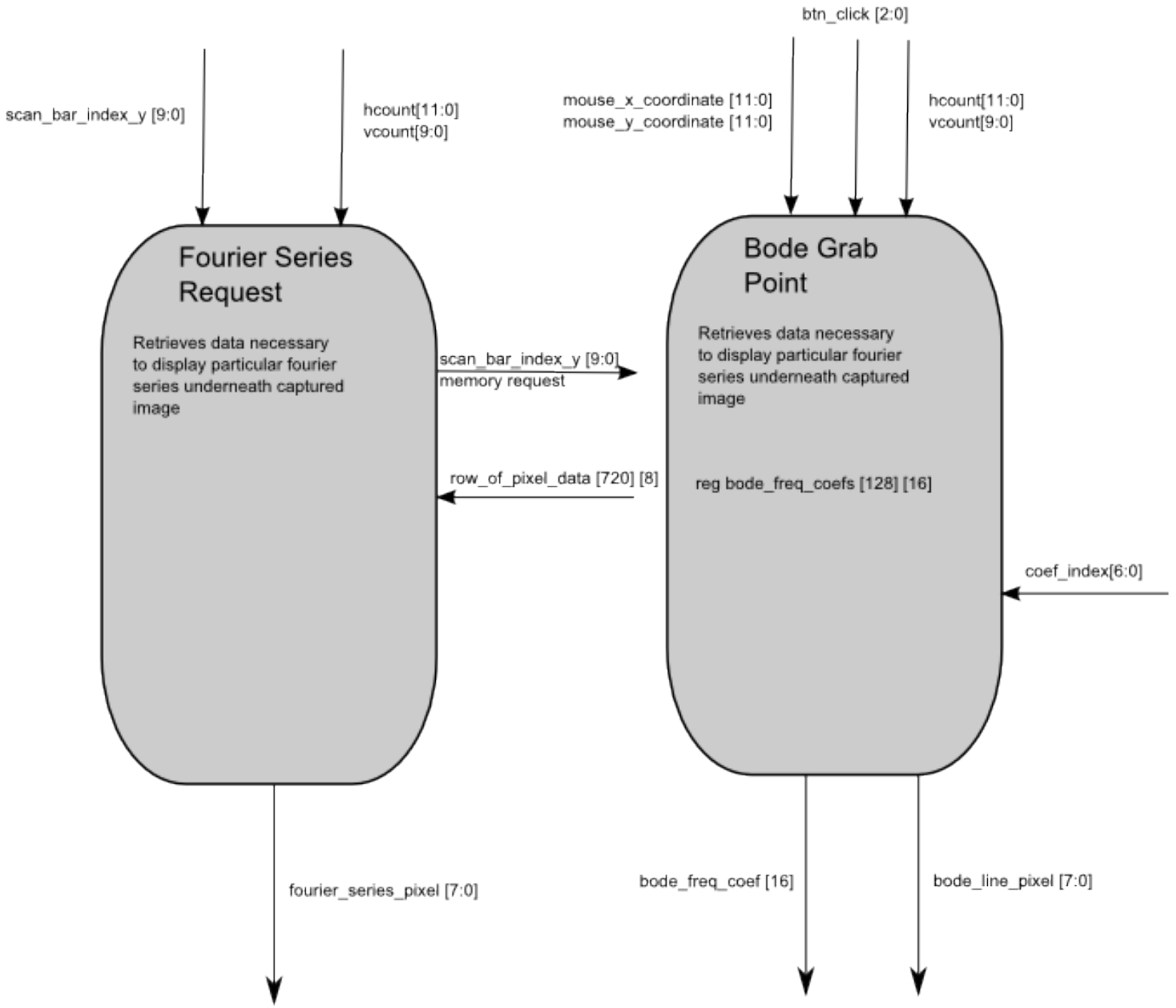
Full System Block Diagram

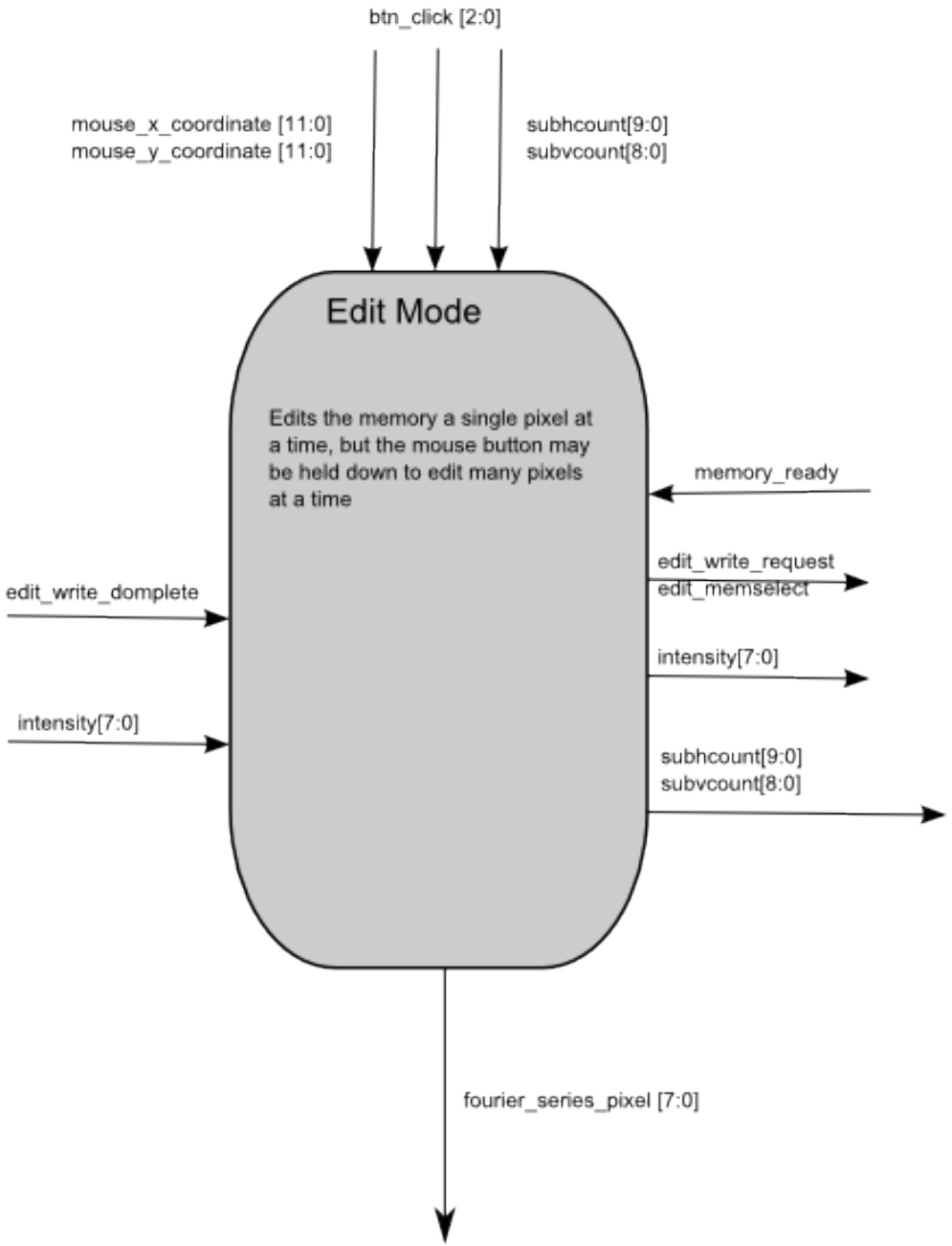


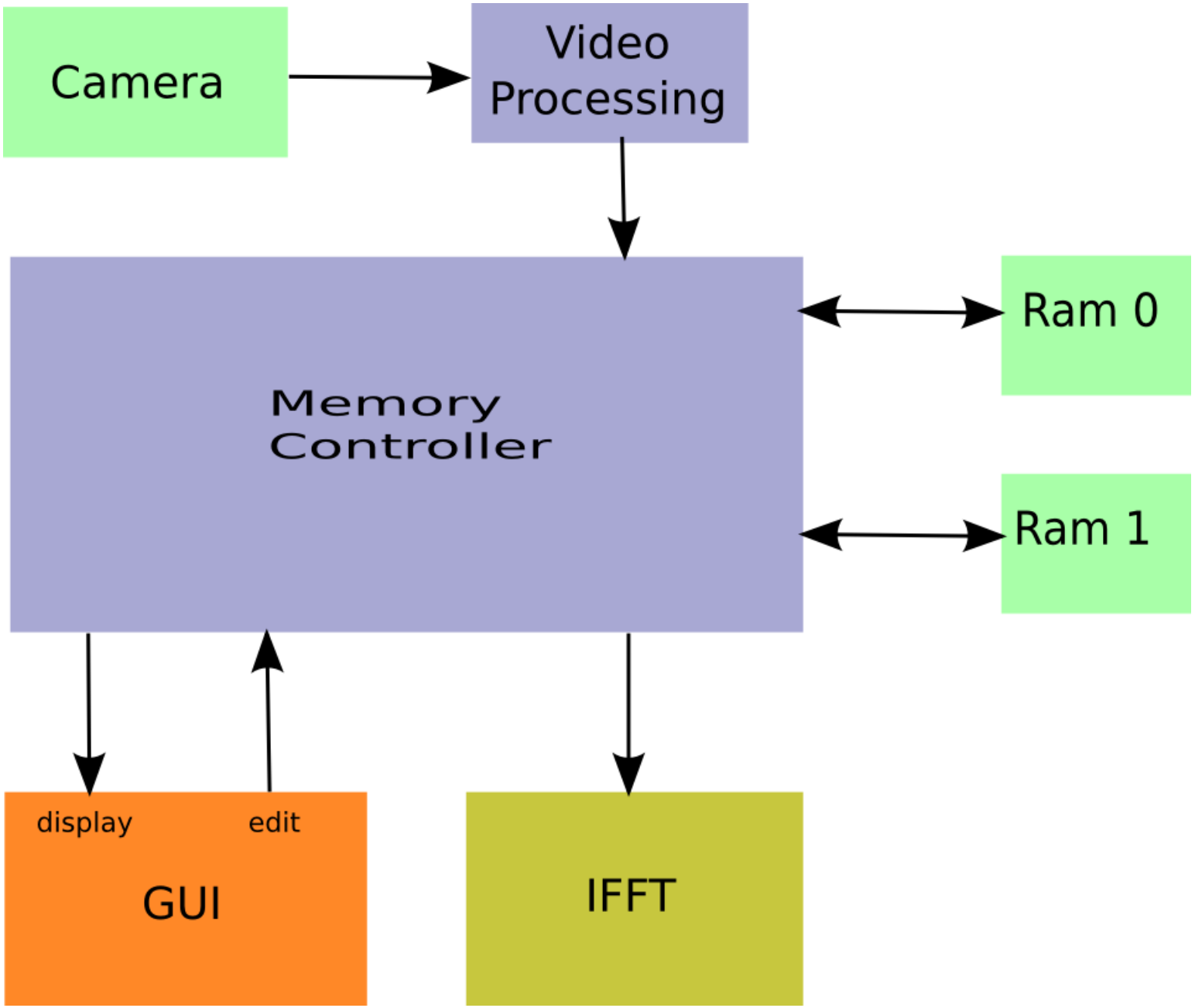

PAUSE
EDIT
IMAGE
OVERLAY
ENABLE
100
INTENSITY
1
0
0 HZ
10000 HZ
0 HZ
10000 HZ

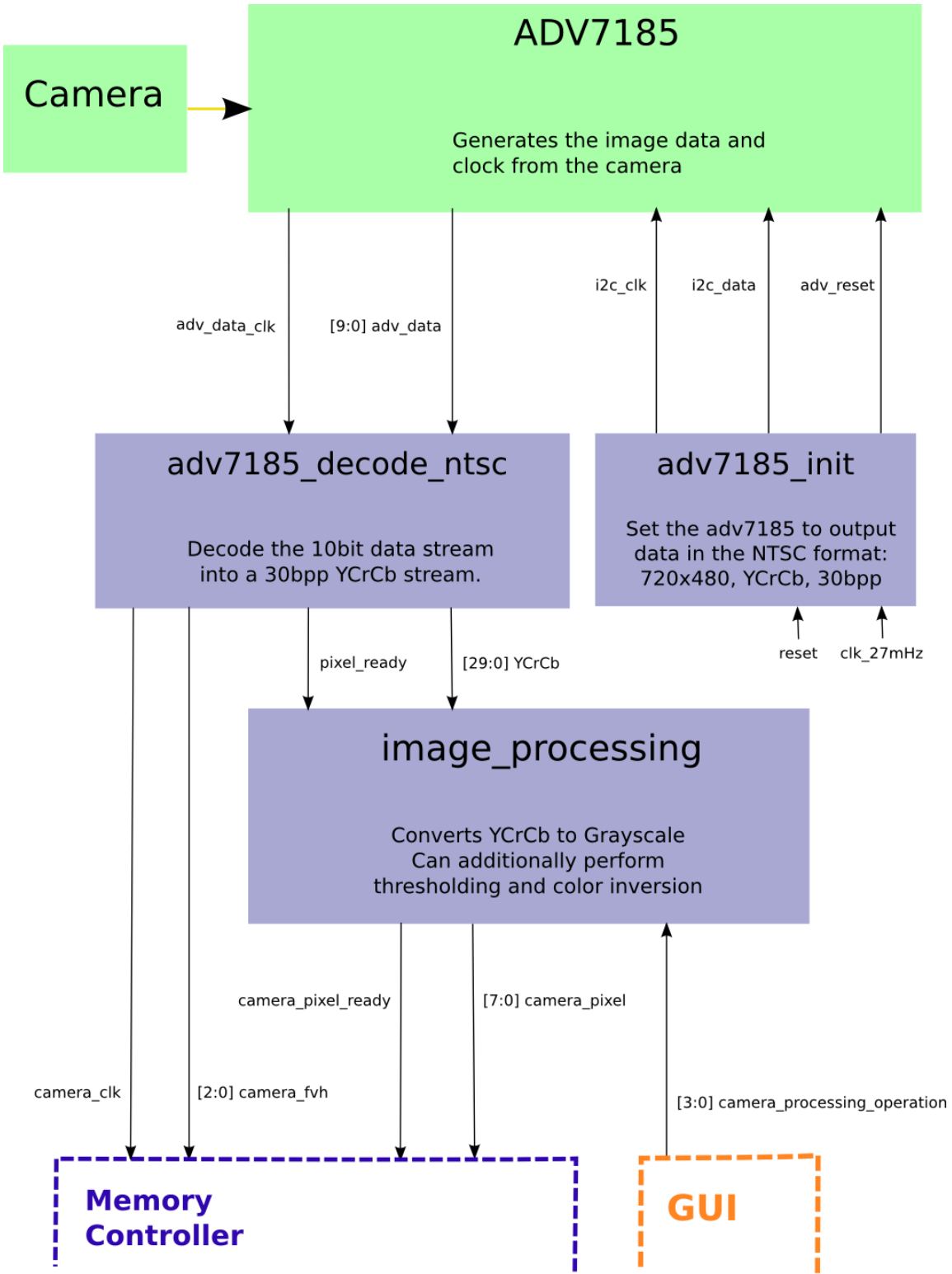

Mouse
PS/2
Interface
[11:0] x coordinate
[11:0] y coordinate
[2:0] button click (left-mid-right)
reset
65 mHz clock
pause =
pause | scan_grab
camera_capture_request
camera_capture_memselect
display_read_request
captured_image_index_x [9:0]
captured_image_index_y [8:0]
GUI - Display
Sprites:
Buttons
-Pause
-Image Capture
-Image
-Bode Plot Line
-Bode Grab Point
-Edit Mode
-Enable Overlay
Diagrams
-Bode Plot
-Overlay Capture
-Scanline
Mouse Cursor
15x12 bit par/reg : x
15x10 bit par/reg : y
GUI - Controller
Calculate new bode line plot
Control Edit Mode
Request Overlay
button_name_activated - 1 bit
pixel_data [31:0]
*pixel[7:0]
ASCII_address [5:0]
btn_click[2:0]
left, mid right
hcount [10:0]
vcount [9:0]
mouse_x_coordinate [11:0]
mouse_y_coordinate [11:0]
button_name_activated - 1 bit
enable_overlay -1 bit
grab_point_adjust - 1 bit
btn_click[2:0]
left, mid right
mouse_x_coordinate [11:0]
mouse_y_coordinate [11:0]
intensity [7:0]
memory_ready
scan_bar_index_y [9:0]
overlay_request
memory_ready
pixel_value [8]
3r, 3g, 2b
1024x768
VGA Output
Monitor

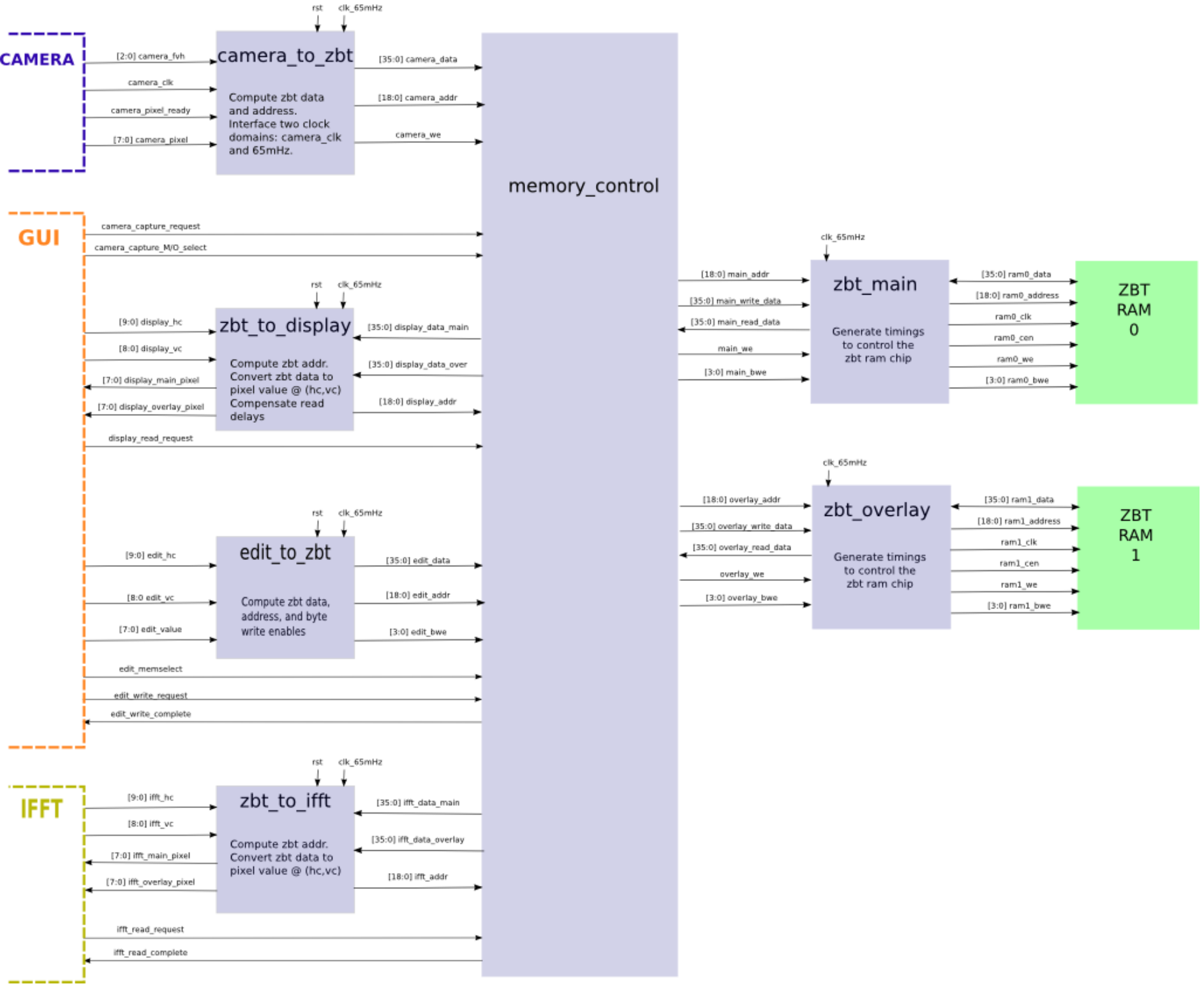

btn_click [2:0]
mouse_x_coordinate [11:0]
mouse_y_coordinate [11:0]
hcount[11:0]
vcount[9:0]
Mouse Sprite
Spits out the pixel of the mouse pointer. Sprite changes color if the mouse is activated.
Outputs triangular shape
mouse_pixel[7:0]


btn_click [2:0]
scan_bar_index_y [9:0]
hcount[11:0]
vcount[9:0]
mouse_x_coordinate [11:0]
mouse_y_coordinate [11:0]
hcount[11:0]
vcount[9:0]
Fourier Series Request
Retrieves data necessary to display particular fourier series underneath captured image
Bode Grab Point
Retrieves data necessary to display particular fourier series underneath captured image
scan_bar_index_y [9:0]
memory request
row_of_pixel_data [720] [8]
reg bode_freq_coefs [128] [16]
coef_index[6:0]
fourier_series_pixel [7:0]
bode_freq_coef [16]
bode_line_pixel [7:0]

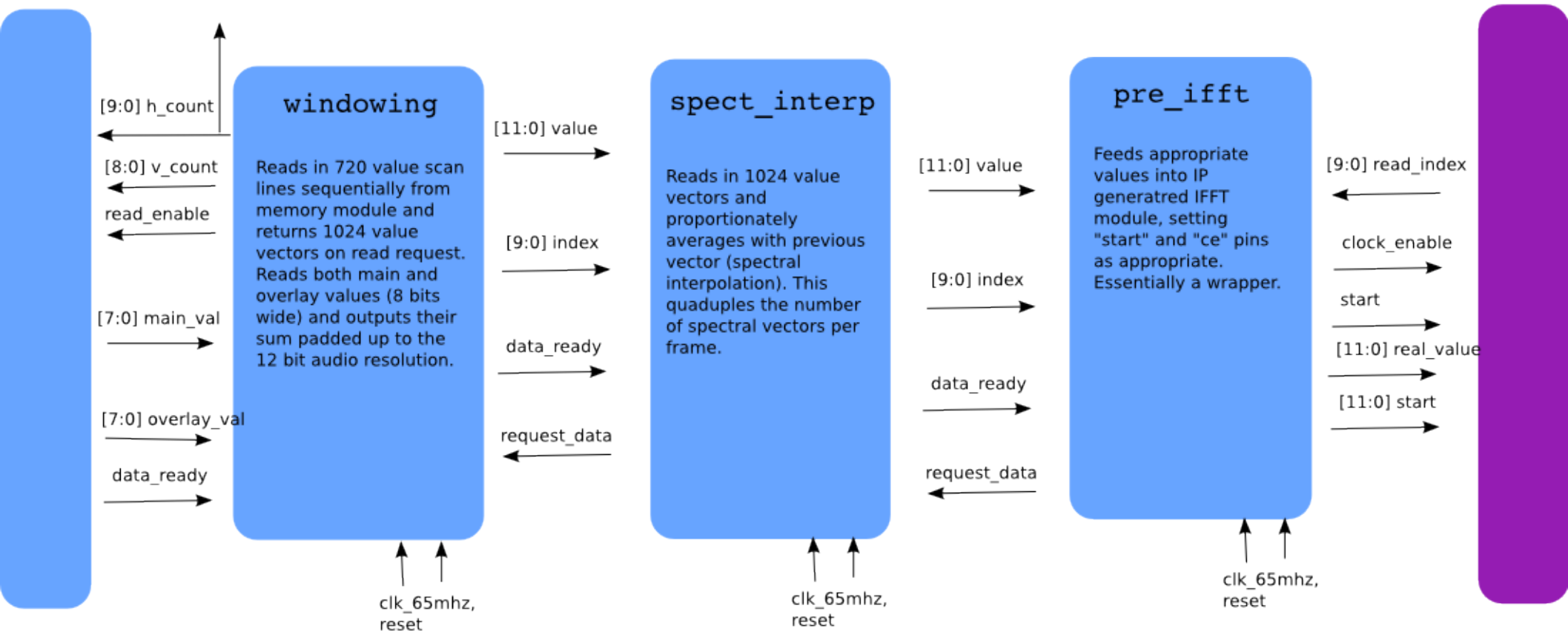

btn_click [2:0]
mouse_x_coordinate [11:0]
mouse_y_coordinate [11:0]
subhcount[9:0]
subvcount[8:0]
Edit Mode
Edits the memory a single pixel at a time, but the mouse button may be held down to edit many pixels at a time
memory_ready
edit_write_request
edit_memselect
edit_write_domplete
intensity[7:0]
intensity[7:0]
subhcount[9:0]
subvcount[8:0]
fourier_series_pixel [7:0]


Camera
Video Processing
Memory Controller
Ram 0
Ram 1
display
edit
GUI
IFFT


Camera
ADV7185
Generates the image data and clock from the camera
adv_data_clk
[9:0] adv_data
i2c_clk
i2c_data
adv_reset
adv7185_decode_ntsc
Decode the 10bit data stream into a 30bpp YCrCb stream.
adv7185_init
Set the adv7185 to output data in the NTSC format: 720x480, YCrCb, 30bpp
reset
clk_27mHz
pixel_ready
[29:0] YCrCb
image_processing
Converts YCrCb to Grayscale
Can additionally perform thresholding and color inversion
camera_pixel_ready
[7:0] camera_pixel
camera_clk
[2:0] camera_fvh
[3:0] camera_processing_operation
Memory Controller
GUI


CAMERA
[2:0] camera_fvh
camera_clk
camera_pixel_ready
[7:0] camera_pixel
rst
clk_65mHz
camera_to_zbt
Compute zbt data and address. Interface two clock domains: camera_clk and 65mHz.
[35:0] camera_data
[18:0] camera_addr
camera_we
memory_control
GUI
camera_capture_request
camera_capture_M/O_select
rst
clk_65mHz
[9:0] display_hc
[8:0] display_vc
[7:0] display_main_pixel
[7:0] display_overlay_pixel
zbt_to_display
Compute zbt addr. Convert zbt data to pixel value @ (hc,vc) Compensate read delays
[35:0] display_data_main
[35:0] display_data_over
[18:0] display_addr
display_read_request
rst
clk_65mHz
[9:0] edit_hc
[8:0 edit_vc
[7:0] edit_value
edit_to_zbt
Compute zbt data, address, and byte write enables
[35:0] edit_data
[18:0] edit_addr
[3:0] edit_bwe
edit_memselect
edit_write_request
edit_write_complete
IFFT
rst
clk_65mHz
[9:0] ifft_hc
[8:0] ifft_vc
[7:0] ifft_main_pixel
[7:0] ifft_overlay_pixel
zbt_to_ifft
Compute zbt addr. Convert zbt data to pixel value @ (hc,vc)
[35:0] ifft_data_main
[35:0] ifft_data_overlay
[18:0] ifft_addr
ifft_read_request
ifft_read_complete
clk_65mHz
[18:0] main_addr
[35:0] main_write_data
[35:0] main_read_data
main_we
[3:0] main_bwe
zbt_main
Generate timings to control the zbt ram chip
[35:0] ram0_data
[18:0] ram0_address
ram0_clk
ram0_cen
ram0_we
[3:0] ram0_bwe
ZBT RAM 0
clk_65mHz
[18:0] overlay_addr
[35:0] overlay_write_data
[35:0] overlay_read_data
overlay_we
[3:0] overlay_bwe
zbt_overlay
Generate timings to control the zbt ram chip
[35:0] ram1_data
[18:0] ram1_address
ram1_clk
ram1_cen
ram1_we
[3:0] ram1_bwe
ZBT RAM 1

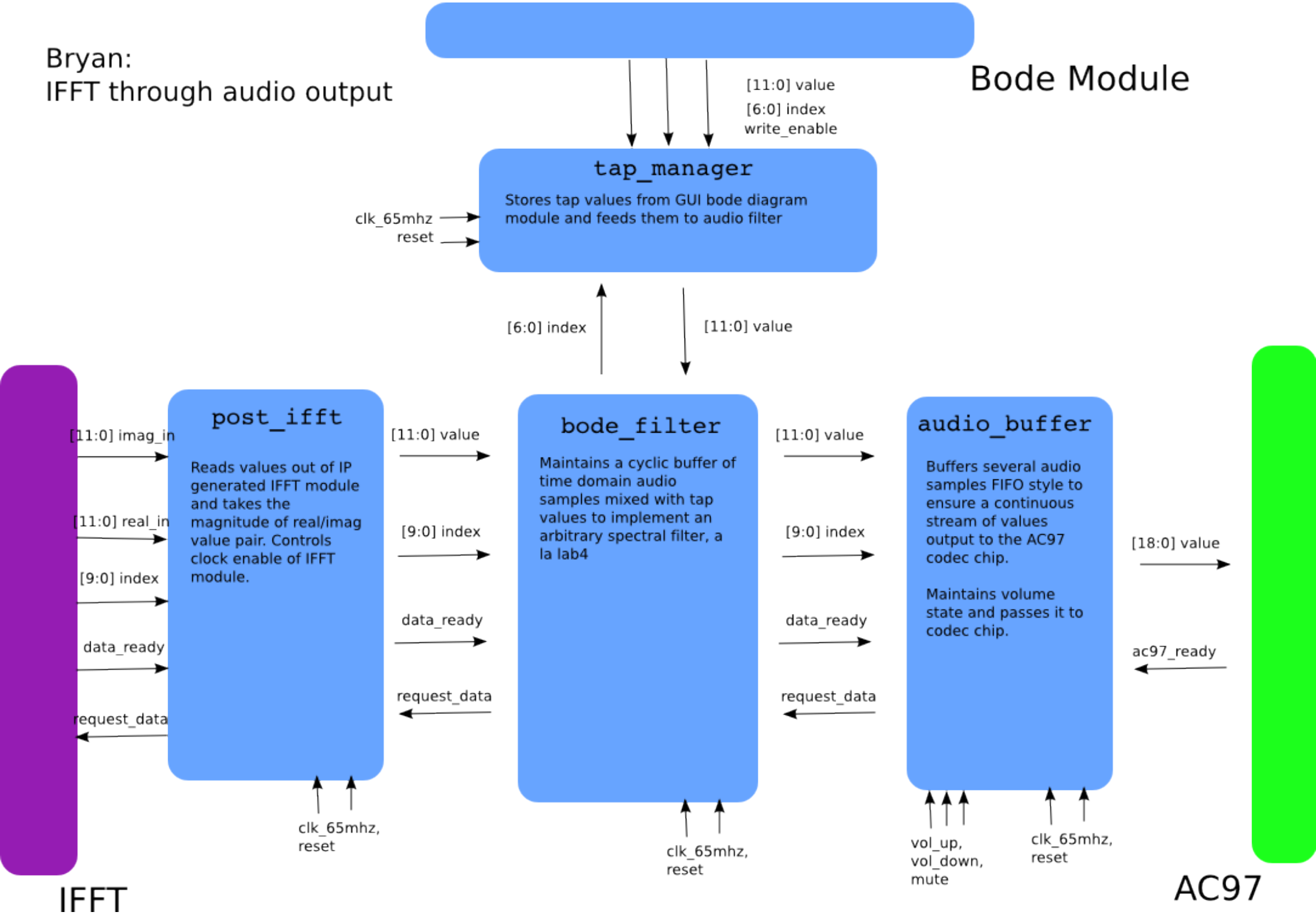
Bryan:
RAM to IFFT



RAM Module

IFFT

Bryan:
IFFT through audio output



Rough Timeline

- ***Week of 11/17:*** Basic Module Functionality on the bench
- ***Week of 11/24:*** Polish and Integrate Modules
- ***Week of 12/01:*** Characterization and testing; extra features.
- ***Week of 12/08:*** Presentations and report due