Real-Time Visual Audio Composition
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 outputting audio
Full System Block Diagram

Whiteboard Drawing

Camera

NTSC Decoder

Grayscale Conversion

Asking for Overlay?

[11:0] x coordinate
[11:0] y coordinate
[2:0] button click (left-mid-right)

1MB ZBT RAM Captured Image

[0:0] 1MB ZBT RAM Overlay Storage

Overlays display in color if enabled and black otherwise
If overlay enabled, heavy summing, heavy multiplication only in IFFT stage

PS/2 Interface

GUI

Central Controller

Mouse

Necessary Sprites:
- Pause
- Image Capture
- Image Overlay Capture
- Scanline
- Mouse Cursor

[8x12 bit registers x]
[8x10 bit registers y]

Tyler

ASCII Memory:
- BRAM
- 0-9
- A-Z all caps
- 8 bits per row
- 12 rows per character
- 36 characters
- 432 rows x 8 bits = 3456 bytes

Monitor

1024x768 VGA Output

Bryan

AC97

Speaker

128 Tap IFFT

128 Tap IFFT

128 x 16bit Frequency Domain Coefficients

Output Buffer

Output Filter

AC97

BW = 10240 Hz
fsw = 24000Hz

Full System Block Diagram

Unless otherwise stated all components are running at a 65 MHz clock.
Mouse Sprite

Spits out the pixel of the mouse pointer. Sprite changes color if the mouse is activated.

Outputs triangular shape

btn_click [2:0]
mouse_x_coordinate [11:0]
mouse_y_coordinate [11:0]
hcount[11:0]
vcount[9:0]
mouse_pixel[7:0]
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
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.

- btn_click [2:0]
- mouse_x_coordinate [11:0]
- mouse_y_coordinate [11:0]
- subhcount[9:0]
- subvcount[8:0]
- memory_ready
- edit_write_request
- edit_memselect
- intensity[7:0]
- subhcount[9:0]
- subvcount[8:0]
- fourier_series_pixel [7:0]
ADV7185

Generates the image data and clock from the camera

Camera

adv_data_clk  [9:0] adv_data

adv7185_decode_ntsc

Decode the 10bit data stream into a 30bpp YCrCb stream.

pixel_ready  [29:0] YCrCb

adv7185_init

Set the adv7185 to output data in the NTSC format: 720x480, YCrCb, 30bpp

reset  clk_27mHz

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
**Ram to IFFT**

**Windowing**
- Reads in 720 value scan lines sequentially from memory module and returns 1024 value vectors on read request.
- Reads both main and overlay values (8 bits wide) and outputs their sum padded up to the 12 bit audio resolution.

**Spect_interp**
- Reads in 1024 value vectors and proportionately averages with previous vector (spectral interpolation).
- This quadruples the number of spectral vectors per frame.

**Pre_ifft**
- Feeds appropriate values into IP generated IFFT module, setting "start" and "ce" pins as appropriate.
- Essentially a wrapper.

**RAM Module**
- [9:0] h_count
- [8:0] v_count
- read_enable
- [7:0] main_val
- [7:0] overlay_val
- data_ready

**IFFT**
- [9:0] read_index
- clock_enable
- start
- [11:0] real_value
- [11:0] start

**Clk_65mhz, reset**

**Clk_65mhz, reset**
Bryan: 
IFFT through audio output

Bode Module

```plaintext
[11:0] value
[6:0] index
write_enable
```

tap_manager
Stores tap values from GUI bode diagram module and feeds them to audio filter

clk_65mhz
reset

[6:0] index
[11:0] value

post_ifft
Reads values out of IP generated IFFT module and takes the magnitude of real/imag value pair. Controls clock enable of IFFT module.

[11:0] imag_in
[11:0] real_in
[9:0] index
data_ready
request_data

clk_65mhz, reset

[bode_filter]
Maintains a cyclic buffer of time domain audio samples mixed with tap values to implement an arbitrary spectral filter, a la lab4

[11:0] value
[9:0] index
data_ready
request_data

clk_65mhz, reset

audio_buffer
Buffers several audio samples FIFO style to ensure a continuous stream of values output to the AC97 codec chip.
Maintains volume state and passes it to codec chip.

[11:0] value
[9:0] index
data_ready
request_data

vol_up, vol_down, mute
clk_65mhz, reset

[18:0] value

ac97_ready
```
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