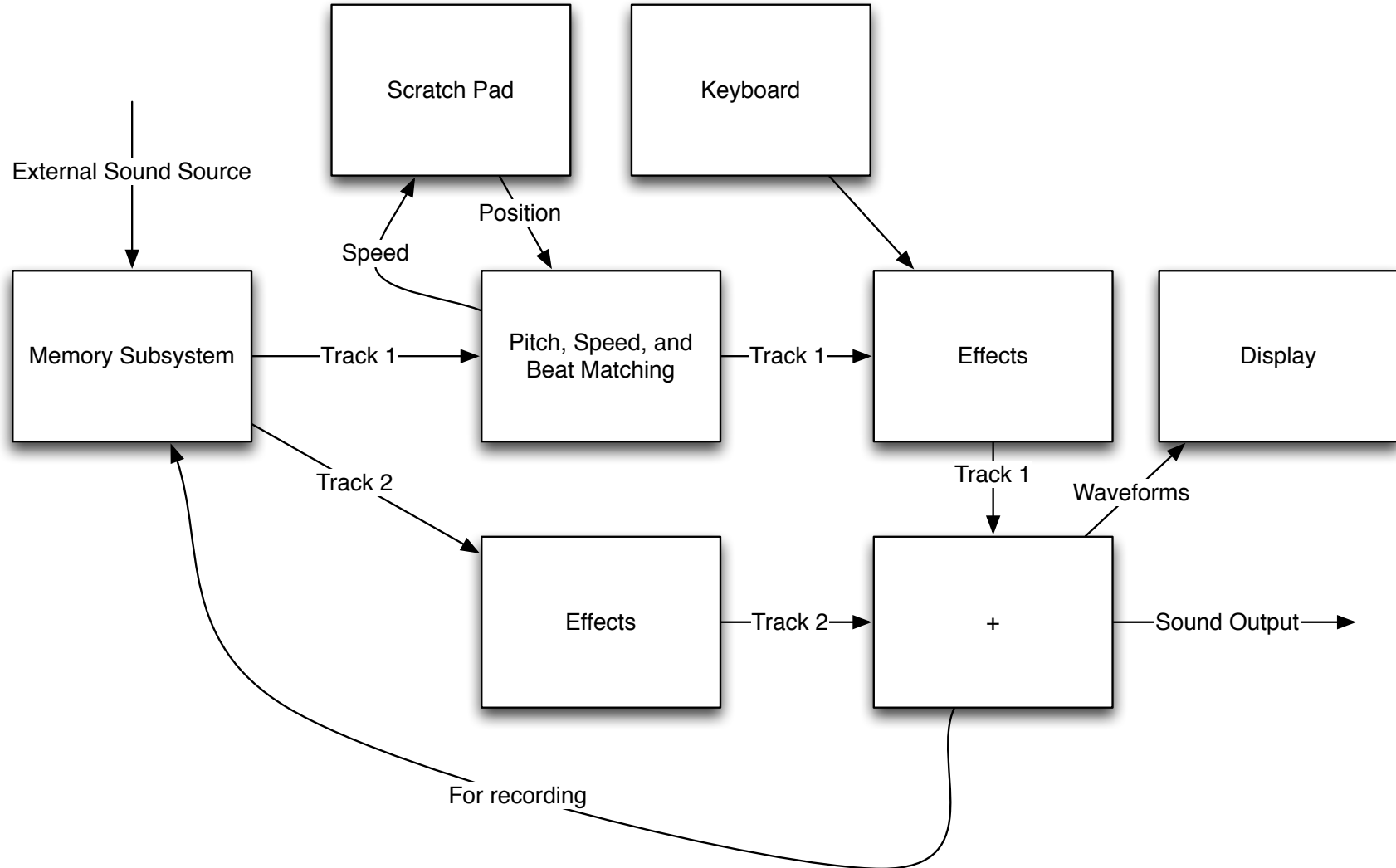


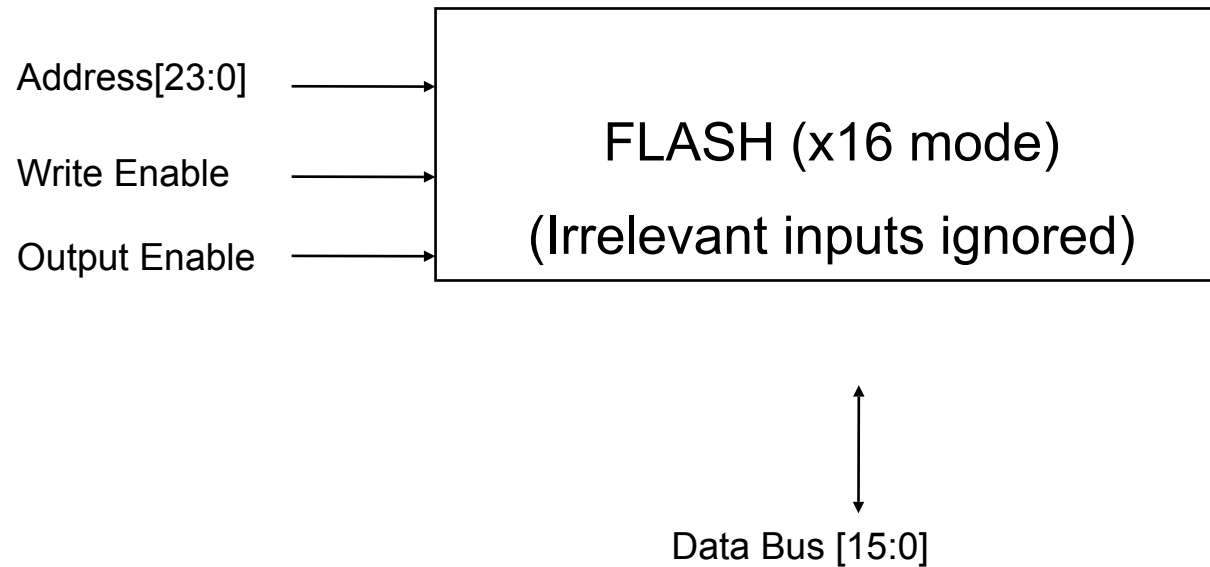
# Mixing Setup

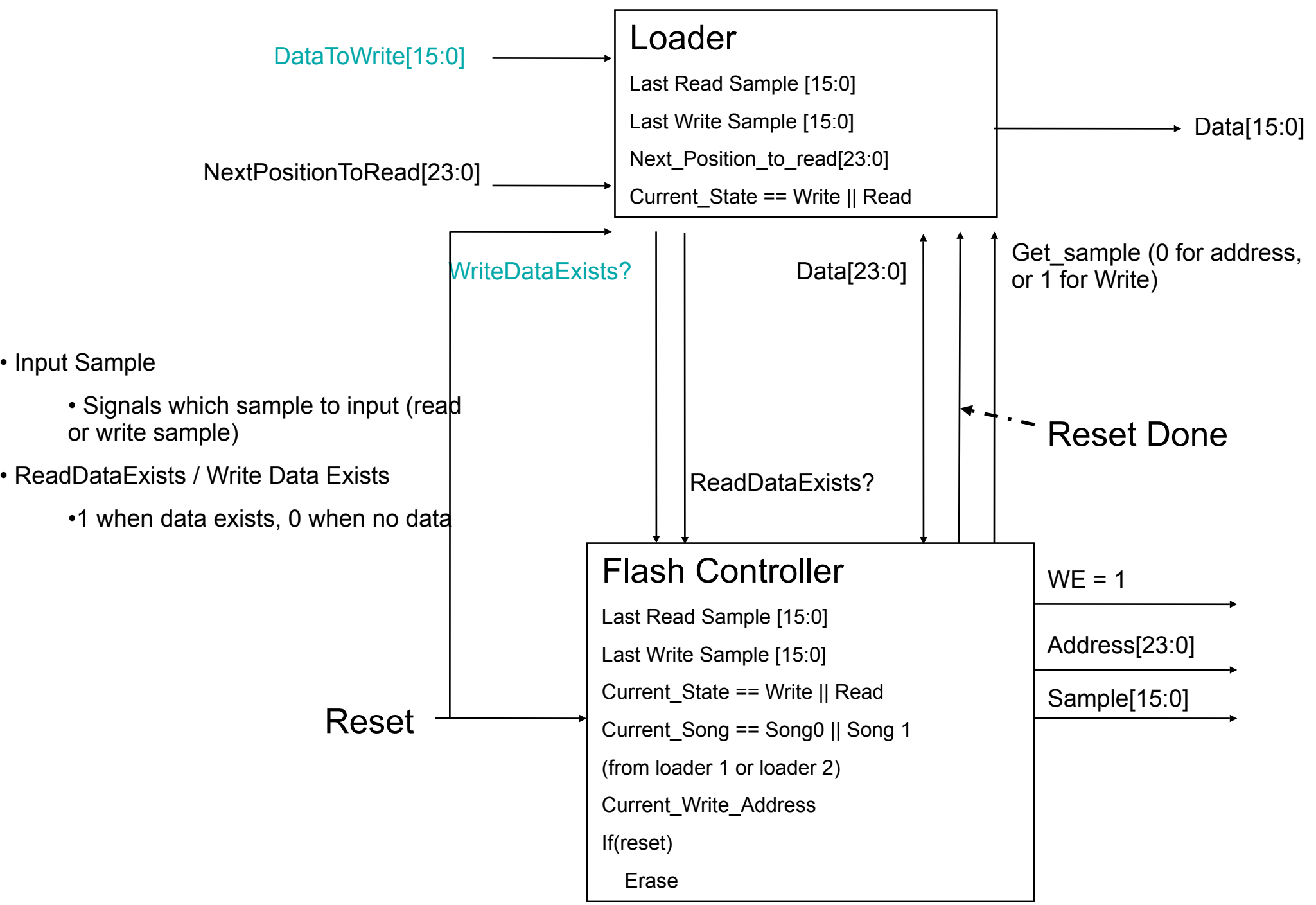
Nathan Artz, Adam Goldstein, Matthew Putnam

# The Big Picture



# Memory Subsystem





DataToWrite[15:0]

### Loader

Last Read Sample [15:0]  
 Last Write Sample [15:0]  
 Next\_Position\_to\_read[23:0]  
 Current\_State == Write || Read

NextPositionToRead[23:0]

Data[15:0]

WriteDataExists?

Data[23:0]

Get\_sample (0 for address,  
 or 1 for Write)

Reset Done

ReadDataExists?

### Flash Controller

Last Read Sample [15:0]  
 Last Write Sample [15:0]  
 Current\_State == Write || Read  
 Current\_Song == Song0 || Song 1  
 (from loader 1 or loader 2)  
 Current\_Write\_Address  
 If(reset)  
 Erase

WE = 1

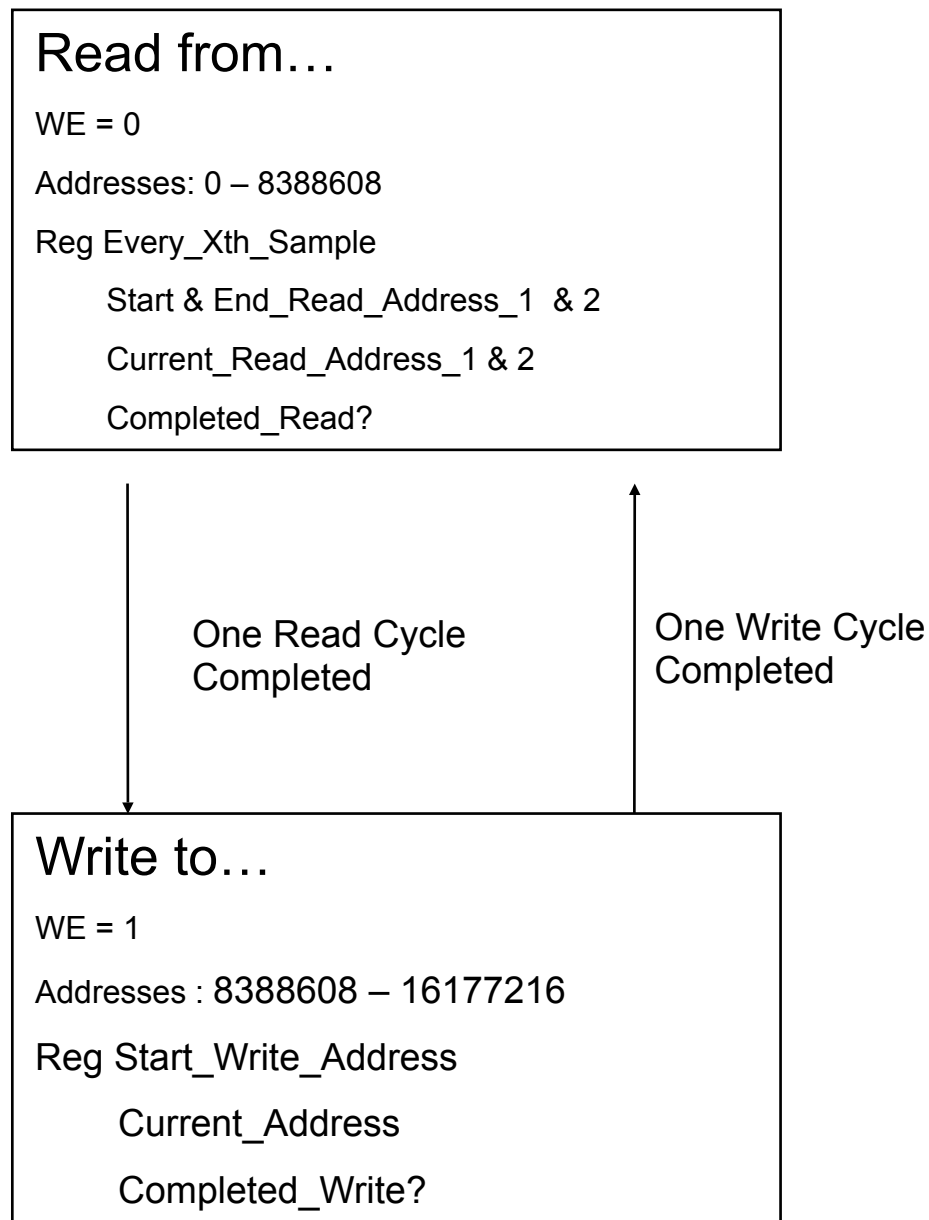
Address[23:0]

Sample[15:0]

Reset

- Input Sample
  - Signals which sample to input (read or write sample)
- ReadDataExists / Write Data Exists
  - 1 when data exists, 0 when no data

# Flash Controller: MIXING

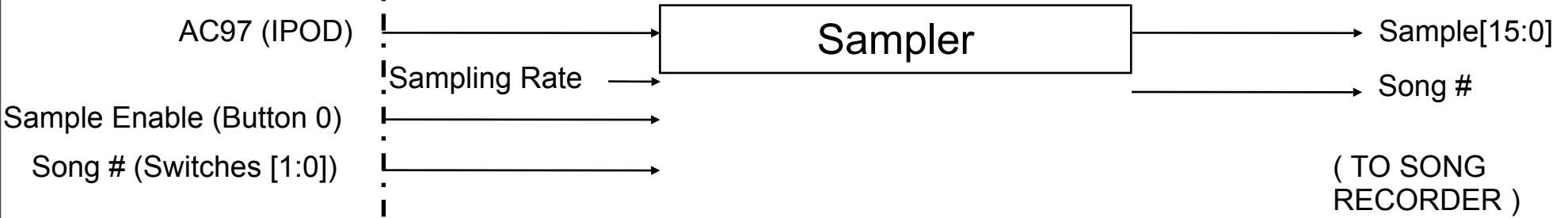


## Spec

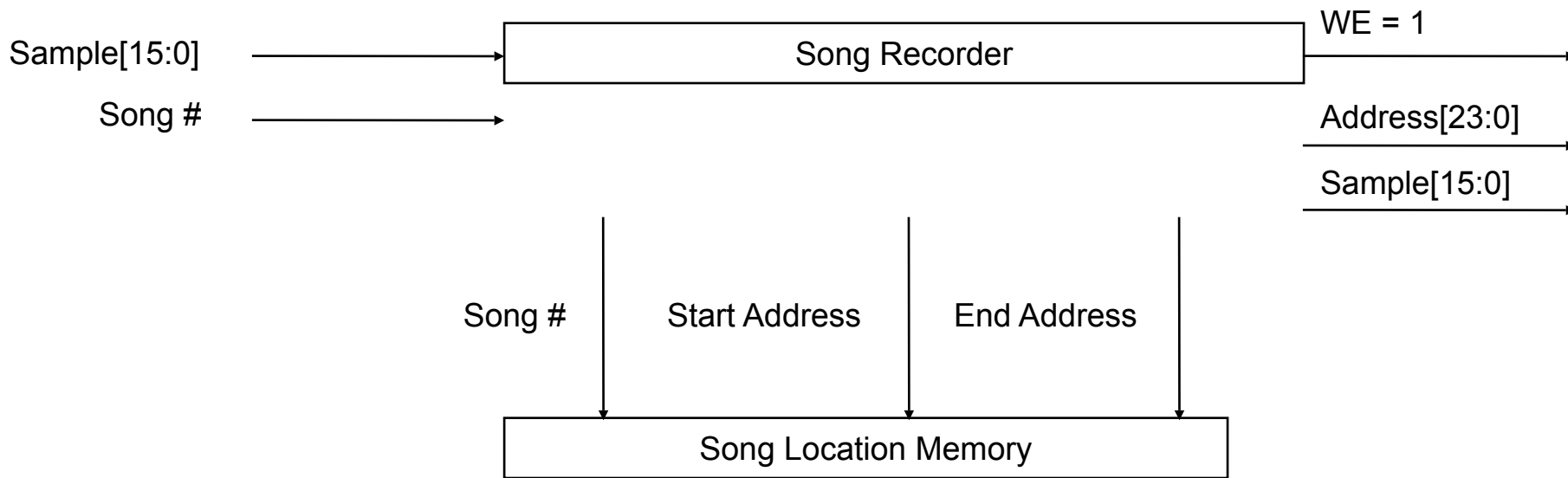
- Read (one cycle)
  - CurrentState == Read
  - Read Data Exists?
    - Get\_sample Signal(0)
      - Read Address
    - Send back to Loader
  - Loader output (next clock cycle)
- IF(WriteDataExists)
  - CurrentState -> Write
    - Get\_sample signal (1)
    - Write Data (100 ns)
- Timings (for one second)
  - 24 Khz Sampling means 24,000 clock cycles needed per sample
    - Sample every  $4.16 \times 10^{-5}$  seconds
  - Read = 25 ns (< 100 ns)
  - Write < 100 ns
- Per cycle
  - Read X 2 (loader 1 and 2)
  - Write Sample (other loader)
  - About 300 ns

EXTERNAL

INTERNAL



Write until full, then simply loop over and overwrite over old.



## Song Recorder Specs

- Flash Controller will be ERASED before songs are loaded
- Songs are written starting at the beginning of the allotted possible address space for that Specified Address Field
- Specified Address Fields (in decimal):
  - Song 0: 0 - 5242880
  - Song 1: 5242880 – 10,485,760
  - RECORDING SPACE IS 10,485,760 – 16,777,216

# The Keyboard



- Given a start pitch and an end pitch, outputs a ratio for frequency shifting



# The Scratch Pad

# The Scratch Pad

- The strategy: Notice *substantial* changes in speed (outside a few % of the current speed)

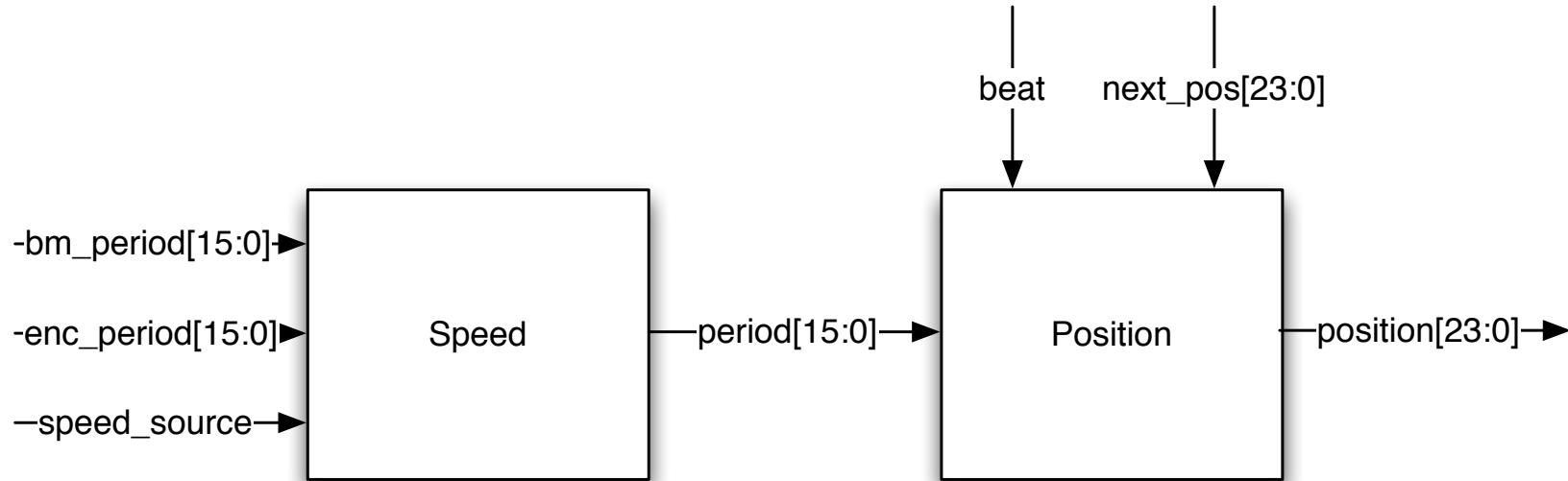
# The Scratch Pad

- The strategy: Notice *substantial* changes in speed (outside a few % of the current speed)
- Convert the speed into a position in memory

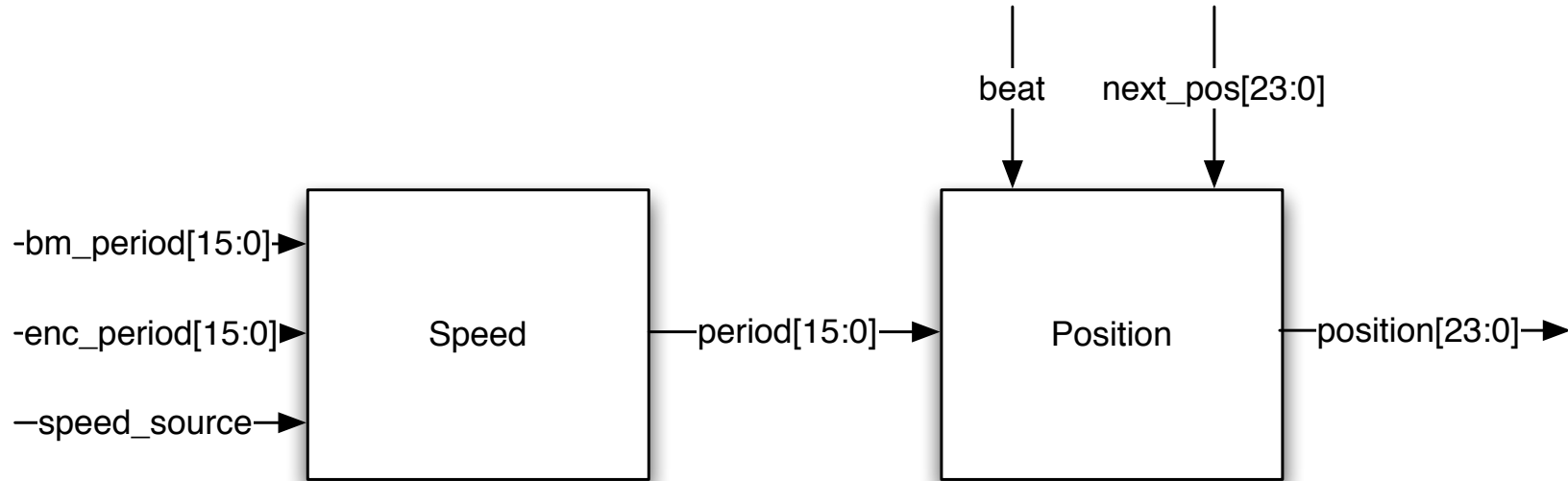
# The Scratch Pad

- The strategy: Notice *substantial* changes in speed (outside a few % of the current speed)
- Convert the speed into a position in memory
- If the speed is being provided externally (e.g. from the beat matcher), change the speed the motor is turning

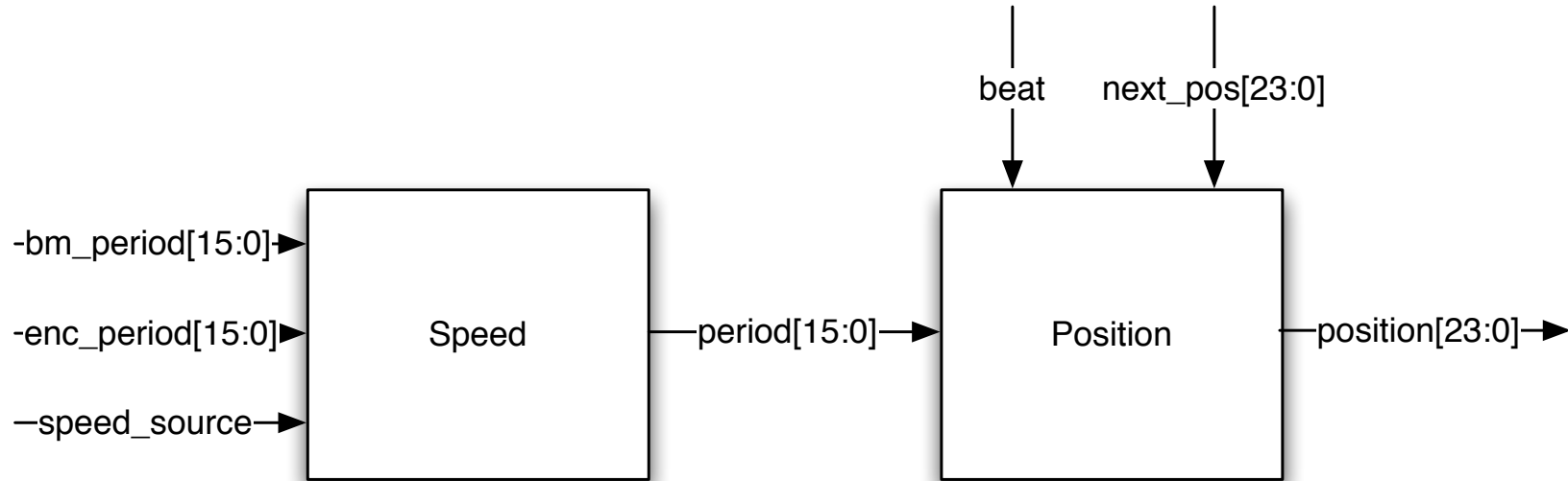
# Speed Changing



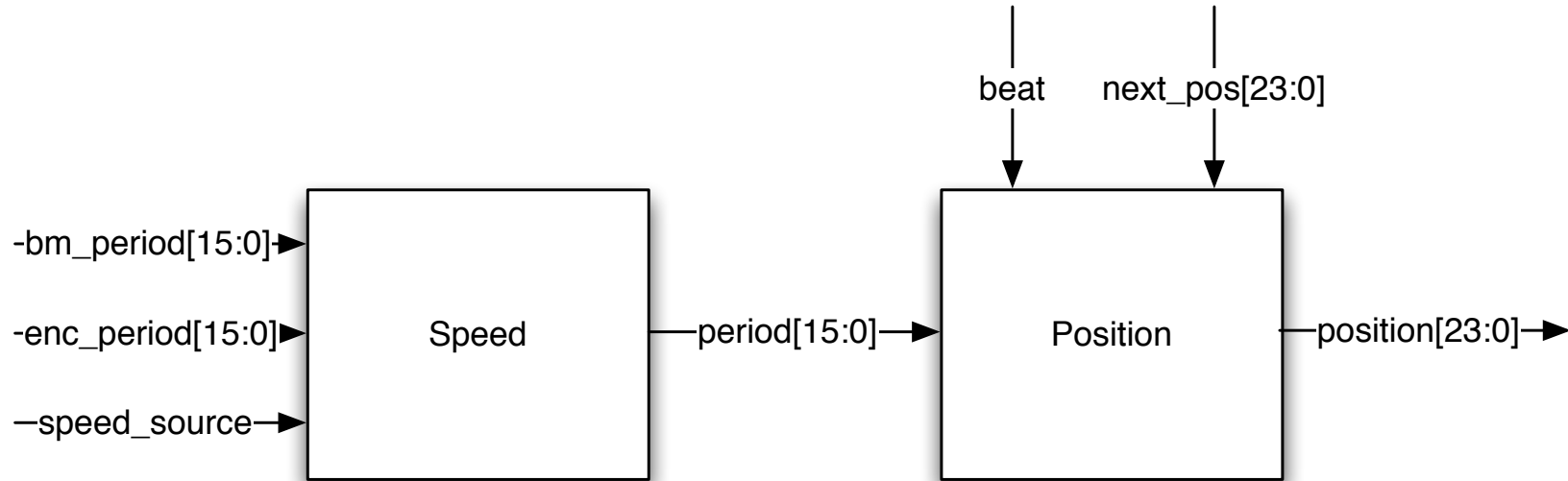
# Speed Changing



# Speed Changing

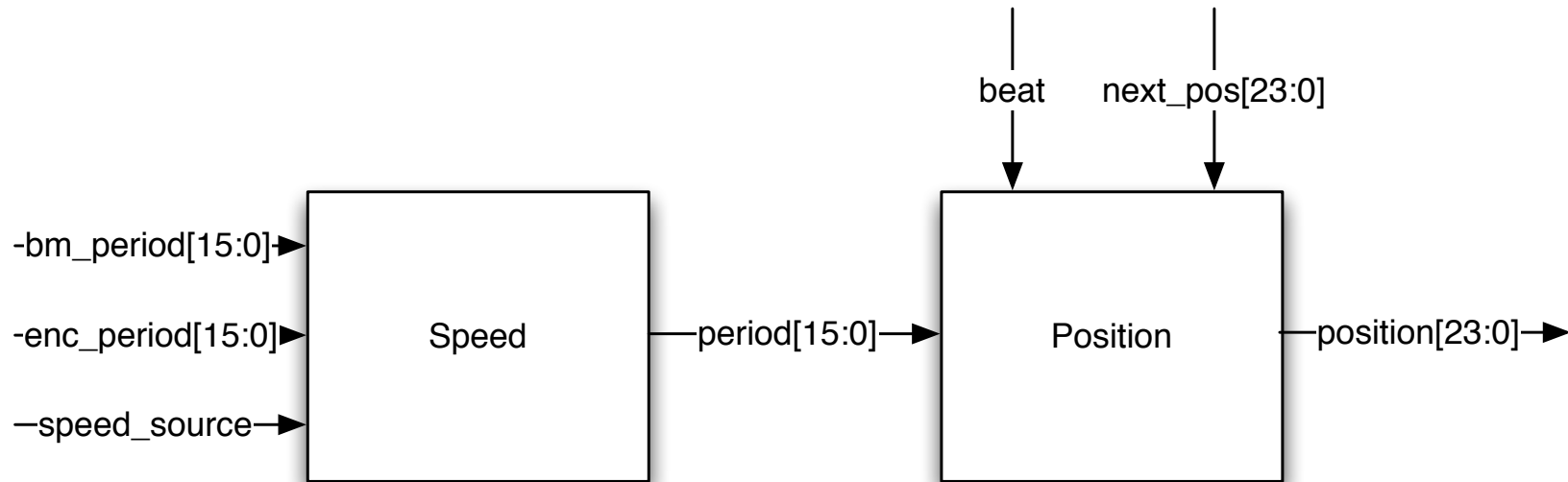


# Speed Changing



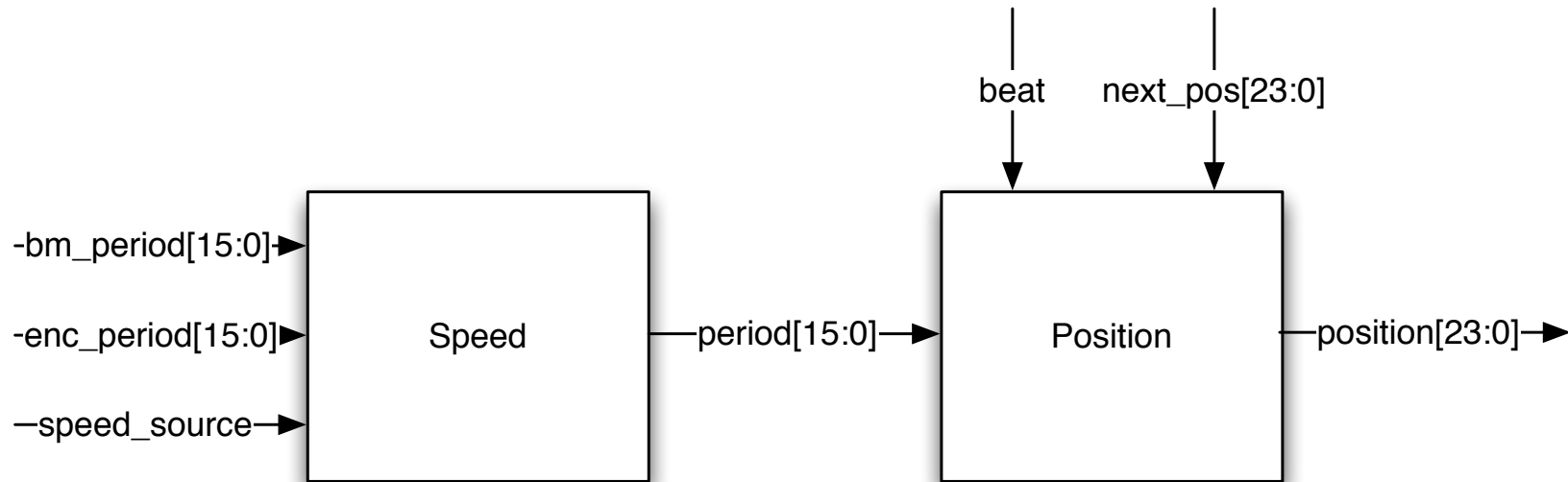


# Speed Changing



- Immediately after starting to beat-match, hold *position* constant until the next beat so the beats are synchronized

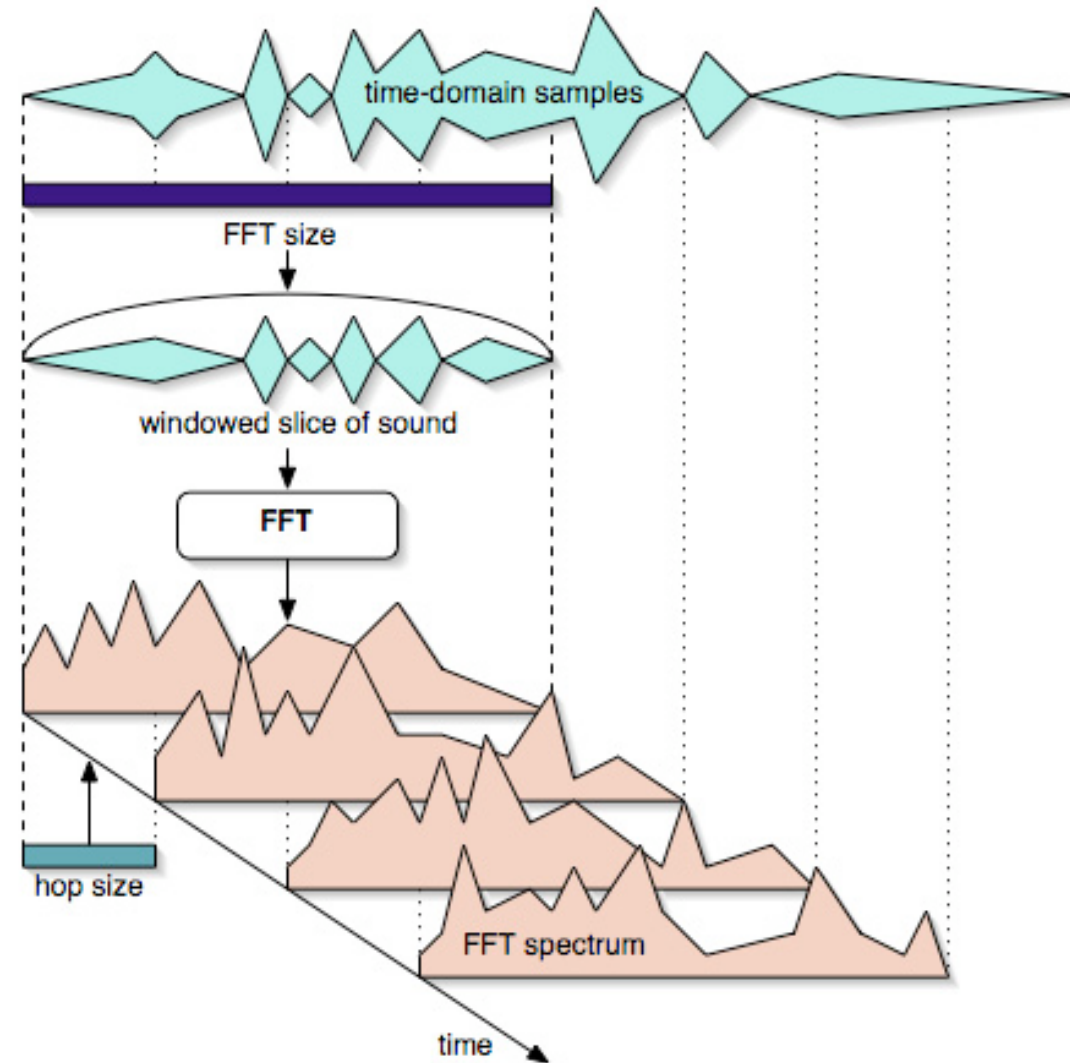
# Speed Changing



- Immediately after starting to beat-match, hold *position* constant until the next beat so the beats are synchronized
- Nathan's sampling stuff

# Frequency Changing

# Frequency Changing



# Frequency Changing

- For 512-sample windows, need  $\sim 1024$  cycles for each window to STFT and ISTFT
- Ideally have 4 overlapping windows; can do with 2

# Other Effects

# Other Effects

- How they work: Echo, Reverb, Bandpass, Etc.

# Other Effects

- How they work: Echo, Reverb, Bandpass, Etc.
- Visual Outputs: Speed, Position, Waveform, Etc.



# Schedule



# Schedule

November 18

Nathan: Record/Output a Single Speed

Matt: Convert Keyboard Signals to Pitches

Adam: Interfacing with Scratch Pad



# Schedule

November 18

Nathan: Record/Output a Single Speed

Matt: Convert Keyboard Signals to Pitches

Adam: Interfacing with Scratch Pad

November 21

Nathan: Up/Downsampling, Outputting at Multiple Speeds

Matt: Echo, Reverb

Adam: Working Vocoder



# Schedule

November 18

Nathan: Record/Output a Single Speed

Matt: Convert Keyboard Signals to Pitches

Adam: Interfacing with Scratch Pad

November 21

Nathan: Up/Downsampling, Outputting at Multiple Speeds

Matt: Echo, Reverb

Adam: Working Vocoder

November 26

Adam: Beat Matching

Matt: Visual Outputs

Nathan: Integration



# Schedule

November 18

Nathan: Record/Output a Single Speed

Matt: Convert Keyboard Signals to Pitches

Adam: Interfacing with Scratch Pad

November 21

Nathan: Up/Downsampling, Outputting at Multiple Speeds

Matt: Echo, Reverb

Adam: Working Vocoder

November 26

Adam: Beat Matching

Matt: Visual Outputs

Nathan: Integration

December 4

All: Integration, Visual Outputs