FPGAAAutotune

Elaine Ng + Kika Arias
What is Autotune?
Methods for Autotune

<table>
<thead>
<tr>
<th>Time Domain Method (TD-PSOLA)</th>
<th>Frequency Domain Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Divide signal into chunks</td>
<td>1. Find STFT</td>
</tr>
<tr>
<td>2. Change frequency by shifting chunks:</td>
<td>2. Find frequency corresponding to each note</td>
</tr>
<tr>
<td>a. closer together = higher freq</td>
<td>3. Shift by convolving original signal with two deltas at correct frequency</td>
</tr>
<tr>
<td>b. farther apart = lower freq</td>
<td></td>
</tr>
<tr>
<td>3. Reconstruct signal by adding chunks</td>
<td></td>
</tr>
</tbody>
</table>
We're doing this in the FREQUENCY DOMAIN

- Most pitch correction implementations (including actual Auto-Tune) uses TD-PSOLA because it's simpler and computationally less intensive...
- So...why are we not doing this?
Overview
Short Time Fourier Transform (STFT)

1. Shift in 1024 samples
2. Apply Hann window
3. Calculate FFT using FFT Core
4. Calculate squared magnitude
5. Store squared magnitude in STFT BRAM
6. Go back a few addresses in audio BRAM
7. Shift in new 1024 samples
8. Repeat (step 2-6)
**Visualization**

1. Get the squared magnitudes from FFT BRAM
2. Map squared magnitude to colors (RGB)
3. Plot a spectrogram for the whole signal like →
Peak Detection

1. Find the location of the notes
2. Find the freq that best represents that chunk
3. Use binary search on LUT to find closest “natural” frequency
Pitch Shifting

1. Construct a big filter that is: two deltas at desired frequency for each harmonic
2. Convolve filter with actual signal!
3. Take the IFFT and output it
Foreseeable Issues

- Memory (STFT, peak detection, pitch shifting, actual audio)
- Tuning sampling rates
- Tuning window sizes
- Timing (efficiency, modules synced properly)
- Restoring original durations
Questions?