FFT and Filters for Audio
• System which takes 8kHz samples and returns:
  – Filtered audio
    • Using an FIR Filter
  – Spectrum Analyzer Output
    • Using a FFT
• Circular Buffer stores the newest input in the 1023 offset

• The two buffer system allows for two reads and two writes to take place in a smaller number of cycles

• All memory will hold a 32 bit complex number

Buffer System
Basic Butterfly

\[ W_N^n = (e^{-j\frac{\pi}{N}})^n \]

\[ F_n = a + W_N^n * b \]

\[ F_{n+N} = a - W_N^n * b \]
Complex Multiplication in Butterfly

\[ (a_r + ja_i) + (b_r + jb_i)(w_r + jw_i) \]

\[ (a_r + ja_i) - (b_r + jb_i)(w_r + jw_i) \]

4 multiplications
10 Levels * 512 different inputs = 5120 cycles of Butterfly Module for one full FFT
Timeline

- Nov. 22 – Completed Testing for Butterfly Implementation in Python
- Nov. 24 – First Implementation of Butterfly in Verilog
- Dec. 1 – Completed FFT Module
- Dec. 3 – Visual Output Implementation Complete
- Dec. 5 – FIR Filters implemented