

Samantha Palazzolo
6.111 Final Project

For my final project, I will be designing a radix-2 FFT which will output a spectrum analyzer on a video screen. In addition, a simple FIR filter system will be set-up, which will influence both the audio and the visual outputs. The FIR filter will multiply the incoming samples by a designated coefficient before releasing the changed samples as the outputs.

For the FFT, many factors have to be taken into account. First is the number of samples, N , the FFT will use. The AC97 A/D converter samples audio signals at 8kHz. This gives 8k samples per second which will fall into the -4000 to +4000 Hz range. Since the positive and negative frequencies will mirror each other, the only output used in the display will be the positive frequencies. With help from Prof. Terman, N was chosen to be 1024 samples. This value gives 512 positive buckets with about 8hz in each bucket, referring to each bar in the visual output.

With N decided upon, next comes the actual FFT implementation. The basic component of the FFT is the butterfly, depicted in Figure 1.

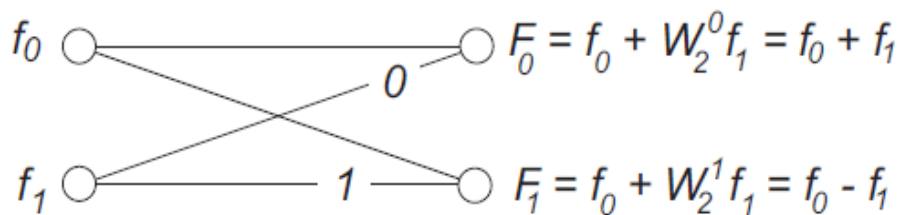


Figure 1: Example butterfly with complex inputs and outputs.
<http://ocw.mit.edu - 2.161>

Each sample will be placed in a circular buffer similar to the one in Lab 4. This buffer will only be written to by the input samples and not changed by any other aspect of this system. There will be a ROM where the twiddle factors, seen in Equation 1, for $n=0, \dots, 1023$ are stored.

$$F_m = \sum_{n=0}^{N-1} f_n e^{-j \frac{2\pi mn}{N}} = \sum_{n=0}^{N-1} f_n W_N^{mn}, \quad m = 0, \dots, N-1$$

where $W_N = e^{-j2\pi/N}$. We also note the following periodic and symmetry properties of W_N

- $W_N^{k(N-n)} = W_N^{-kn} = \overline{W_N^{kn}}$ (complex conjugate symmetry),
- $W_N^{kn} = W_N^{k(n+N)} = W_N^{n(k+N)}$ (periodicity in n and k),
- $W_N^n = -W_N^{n-N/2}$ for $n \geq N/2$.

Equation 1: F_m is the DFT equation. W_N appears in the DFT equation. It's periodic and symmetry properties are why the FFT works.

<http://ocw.mit.edu - 2.161>

W_N is a complex number and will be stored in a format that keeps track of the real and imaginary parts. When the FFT is initiated, each butterfly pair, designated using bit reversal order, will be put in the butterfly module (one pair at a time) and the outputs (which are complex) are saved in the specified location in a new buffer. For this FFT, there are 10 levels of butterflies. Each level has $N/2$, or 512, complex multiplications. The final output will then be given to the spectrum analyzer module.

Due to the complexity of the FFT butterfly system, it is necessary to give it a more in-depth look.

Each sample coming from the A/D converter through the FIR module is considered complex. The samples will be saved as an 8 bit complex number without any imaginary part since the audio input has no imaginary parts. The offset will point to the 1023rd sample, seen in Figure 2, since the order is important.

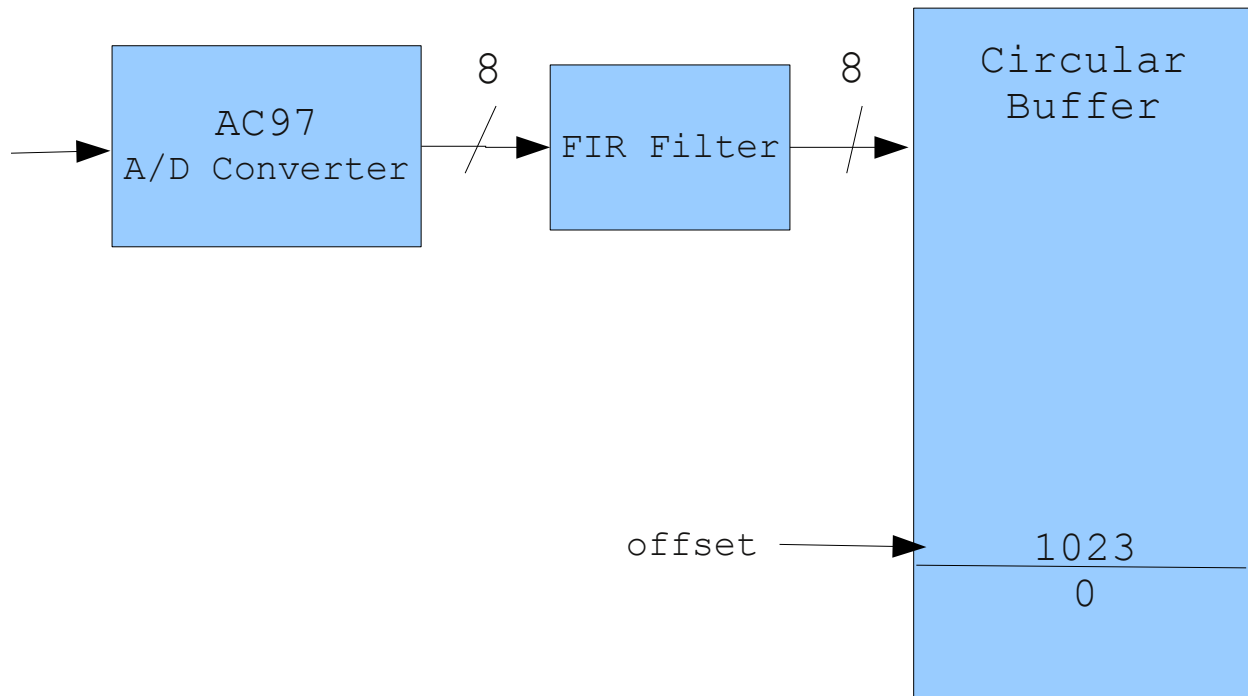


Figure 2: Circular buffer inputs

When the buffer is full and no longer being written to, the FFT will begin. The first level of butterflies will use the values from the circular buffer, while the following levels will use one of two new buffers (*buffer1*, *buffer2*) which will have two places for each sample due to the complex values. Using the bit reversal order previously mentioned, 512 butterfly operations will be performed similar to the depiction in Figure 3.

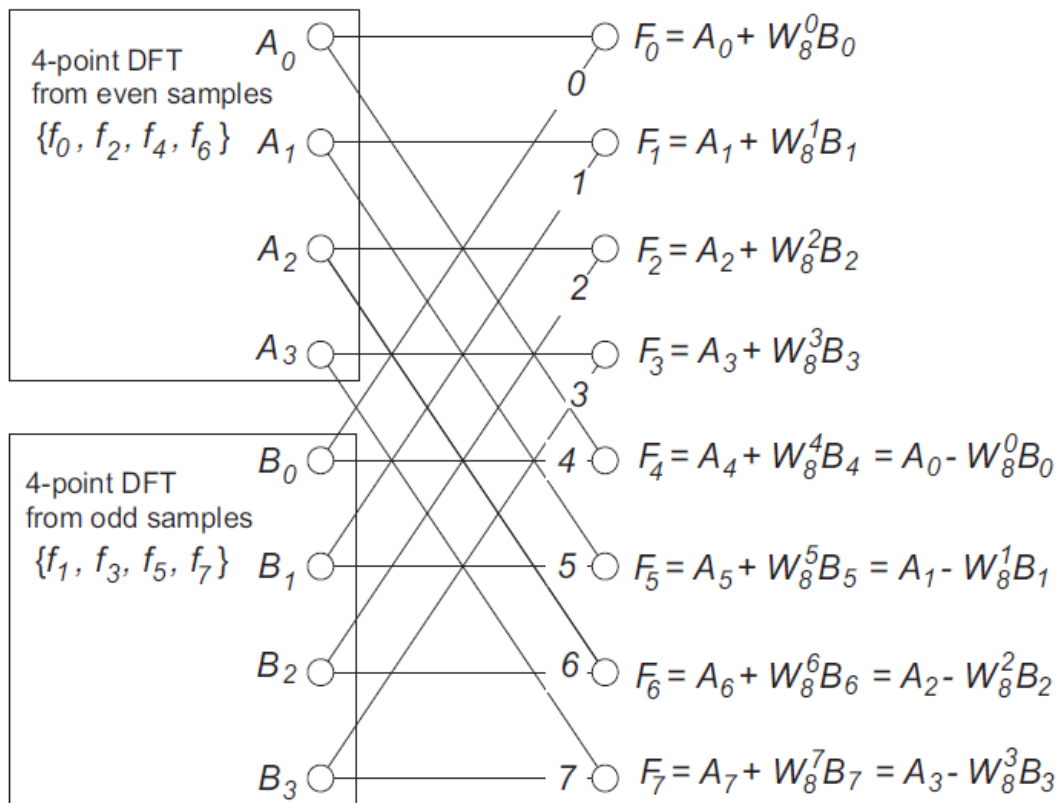


Figure 3: One level of an 8 input FFT butterfly system.

<http://ocw.mit.edu> – 2.161

The outputs of these butterflies will be stored in another buffer than the input samples (i.e. if the inputs came from *buffer1*, then the outputs are saved to *buffer2*). The reason for having two buffers, that will alternate, is because every butterfly cycle there must be two reads and two writes, and the buffers on the 6.111 labkit only have two ports. The values will be cycled through the butterfly 10 times, saving over previous (no longer used) values.

The timing itself is much more complex. The input audio has 125ms between samples. With a 50kHz clock (which will be the one used for this project), this gives 6250 cycles in between each sample. My preliminary FFT design called for 512 butterfly cycles 10 times with one clock cycle for each making the butterfly cycles ~5120 cycles, giving more than enough time for the FFT to run between each sample. The design discussed in this proposal, however, requires more cycles per butterfly.

For this design, every read and write (for the two values for each butterfly) takes 2 clock cycles and every complex multiplication takes two clock cycles as well. With these adjustments, the FFT will not be run every time a new sample comes in. Although not optimal, this will work sufficiently for the visual output which runs at 60Hz, a much slower rate than the rest of the system. The FFT will have more than one new samples every time it runs.

This hopefully gives a good description of the main components in my final project. Things not discussed in the proposal, and to be decided upon at a later time:

- scaling schedule to deal with the 18x18 bit multiplier
- implementation of complex multiplication and storage of complex value bits
- exact inputs to the visual output module (i.e. format, complex value?, etc.)
- Exact timing of the system.