FILTER COEFFICIENT DESIGN
Filter Coefficient Design

- There are many algorithms to find the coefficients for a digital filter. A DSP course will tell you digital filters can be developed that share characteristics with common analog filters such as:
  - Butterworth
  - Chebyshev
  - Bilinear transformation
  - Elliptic

- Some specify no ripple in the pass band or the stop band since this is often a desirable characteristic
Parks-McClellan Method

• Parks-McClellan method is a popular method for designing digital filters
  – Published in the early 70s
  – Iterative
  – Computationally efficient
  – Works by specifying the 1) length of the filter and 2) frequency/magnitude pairs
  – See Oppenheim & Schafer for a thorough discussion
Filter Specification

- Filter specifications are frequently given in dB as min/max attenuation/ripple over frequency regions
- An example filter specification:
  - Low-pass filter
  - Maximum +/- 4dB ripple in passband
  - Sampling frequency is 100 MHz
  - Passband from DC to 12.5 MHz
  - Minimum attenuation 22dB from 19 MHz to 50 MHz
Attenuation and Ripple

- Key filter specifications
  - Min attenuation in stopband
  - Max attenuation in passband
  - Max ripple

\[ \text{ripple} = \text{max} - \text{min} \]

- Minimum attenuation in the stopband
- Maximum attenuation in the passband
Example Filter

- The same example filter specification getting ready to be entered into matlab:
  - Low-pass
  - Notes:
    - 100 MHz = $2\pi = f_s$
    - 50 MHz = $\pi$
    - 12.5 MHz = 0.25 $\pi$
    - 19 MHz = 0.38 $\pi$
  - frequencies specified as fractions of $\pi$: [0 0.25 0.38 1];
  - corresponding amplitudes: [1 1 0 0];
  - Parks-McClellan ignores every other interval starting with the second one ($0.25 \pi - 0.38 \pi$). But this is ok—in this example, we don’t care about transition band between $0.25 \pi$ and 0.38 $\pi$ anyway
  - Use the `remez()` function in matlab
Example Filter

- 7 coeffs.
Example Filter

- 11 coeffs.
Example Filter

- 21 coeffs.
Example Filter

- 51 coeffs.
Example 21-tap Filter

- \texttt{coeffs = remez(20, [0 0.25 0.30 1], [1 1 0 0]);}
- Notice \texttt{remez} function’s first argument is the number of desired taps minus 1
- \texttt{remez()} for filter design.
  \texttt{>> help remez}
  to get more information on the Matlab function
- To plot the coefficients, use
  \texttt{stem(-10:10, coeffs);}
Example Filter Coefficients

- Coefficients of 21-tap filter
- Note sinc() shape in time domain
- Remember this is a low-pass filter which is a rect() in the frequency domain
Seeing the Frequency Response of Filters
Filter Frequency Response (Method I)

• To see the frequency response of a vector of filter coefficients
• Method 1
  - freqz()
    • Exact frequency response
    • Very fast
Filter Frequency Response (Method II)

- To see frequency response of a filter (method II)
  1. Make flat (white) spectrum input signal
  2. Send signal into filter and look at output spectrum
     - Requires many samples for accurate output (not exact)
     - Much slower
     - Sometimes the only way to see spectrum
       - Ex: an arbitrary signal, not a filter response
       - Ex: hardware rounding
       - Ex: signal saturation

- Example matlab code:
  
  ```matlab
  in = rand(1, 100000) - 0.5;
  out = conv(coeffs, in) + 0.25;   % Hypothetical ¼ LSB bias
  abs(fft(out))
  psd(out)
  spectrum(out)
  ```