HW #6 : ECE 444/644 : NDSU : Fall, 2020
DUE: Monday, November 16

1. Write your own MATLAB program/function that designs a digital IIR BPF that meets the following specifications:

   - Sampling frequency of $F_\mathrm{s} = 10$ kHz.
   - INPUT: Selectable factor-of-2 filter order K.
   - INPUT: Selectable 2 dB lower passband frequency $= 250 \leq f_{\mathrm{p}_1} \leq 2000$ [in Hz]. The upper 2 dB passband frequency is determined as $f_{\mathrm{p}_2} = \frac{F_\mathrm{s}}{2} - f_{\mathrm{p}_1}$.
   - Family is inverse Chebyshev BPF with maximum stopband gain of $\frac{1}{10}$.
   - Design method is bilinear transform with prewarping that preserves the 2 dB passband frequencies.
   - OUTPUT: Filter coefficients (32- or 64- bit floats) for realization as cascade of second-order DFII filter stages. It is recommended that your MATLAB code automatically generate a header file that contains the filter coefficients for your project. For example:

     ```
     fid = fopen('coef.h','w');
     fprintf(fid,'#define NS %d \n',NS);
     fprintf(fid,'float A[NS][3] = { \n');
     for ns = 1:NS-1,
         % fprintf statement here
     end
     % fprintf for last row of A matrix
     % Repeat technique for B matrix
     fclose(fid);
     ```

   - OUTPUT: Digital filter pole/zero plot that clearly indicates pole-zero groupings for each second-order section as well as section ordering.
   - OUTPUT: Frequency response plot over $-\frac{F_\mathrm{s}}{2} \leq f = \frac{\Omega F_\mathrm{s}}{2\pi} \leq \frac{F_\mathrm{s}}{2}$.

   Verify code operation by designing a 20th-order filter with lower passband frequency of $f_{\mathrm{p}_1} = 1$ kHz.

2. Create a KEIL project to implement an IIR filter on our K22F hardware. Design requirements include:

   - Sampling rate of 10 kHz.
   - Filter realized as a cascade of second-order DFII IIR filter sections. Your structure should accommodate a range of filter orders (determined by a program parameter K).
   - Highest order possible while meeting timing requirements.
   - Use efficient programming techniques and compiler optimizations. You must write your own filter code – it is not acceptable to use predesigned filter code or functions. Any code optimizations should not alter the generality of the implemented filter – that is, by simply changing a coefficient header file, you should be able to change the implemented filter.
   - Run-time filter selection (none plus four different filters) by single-button presses.
   - Since we will be implementing a BPF with a unipolar DAC, add an appropriate dc offset to keep your final output in range (0 to 4095).

   Test your filter code for correct operation using some simple filter (like an all-pass filter). Determine (or make an *educated* estimate) of the highest order that will still correctly operate.

3. Once you determine the highest filter order K that your K22F program can achieve, use your MATLAB code from problem 1 to design four Kth-order filters with lower passband frequencies of 250 Hz, 500 Hz, 1000 Hz, and 2000 Hz, respectively. Integrate these filter designs with your K22F program. When dowloaded, the K22 should operate as a digital wire. When the button is pressed, the system should cyclically switch between filters as:

$$\text{none} \to f_{\mathrm{p}_1} = 250\text{ Hz} \to f_{\mathrm{p}_1} = 500\text{ Hz} \to f_{\mathrm{p}_1} = 1000\text{ Hz} \to f_{\mathrm{p}_1} = 2000\text{ Hz} \to \textit{repeat}$$

   Test your system for proper operation. Use the K22F LEDs to demonstrate timing is being met. What is the maximum order K you can successfully achieve?

Deliverables:

1. Demonstrate correct operation to the course instructor. You will need to demonstrate the frequency responses of your filters, including passband/stopband characteristics, cutoff frequencies, transition bands, and so forth.

2. E-mail (*Roger.Green@ndsu.edu*) a *zipped* MATLAB mat-file that contains your final design coefficients. Name your file using your initials (e.g., `rag.mat`). Assume each section is represented by a difference equation $y[n] + a_1y[n-1] + a_2y[n-2] = b_0x[n] + b_1x[n-1] + b_2x[n-2]$. Your file should include:

   - A 4-by-NS-by-3 matrix `A` where the first dimension indicates filter number (4 filters, one for each of the 4 lower passband frequencies), the second dimension indicates section number, and the third dimension indicates the feedback coefficients $\texttt{[}1, a_1, a_2\texttt{]}$ for that particular section and filter. Notice, the number of sections `NS` should be $K/2$.
   - A 4-by-NS-by-3 matrix `B` where the first dimension indicates filter number (4 filters, one for each of the 4 lower passband frequencies), the second dimension indicates section number, and the third dimension indicates the feed-forward coefficients $\texttt{[}b_0, b_1, b_2\texttt{]}$ for that particular section and filter.

   For both matrices, sections should be ordered according the how you ordered the sections of your filter (first section first), and filters should be ordered with the smallest lower passband frequency filter occurring first.

3. You should prepare a report summarizing your design and implementation. This is a complex design, so it is incumbent upon you, the designer, to provide results in a clear and readable format. I will not waste time trying to make sense of a poor report, even if the underlying design is the best. Attach code and other bulky supporting material as appendices.