

Assignment #5

Due April 26, 2013

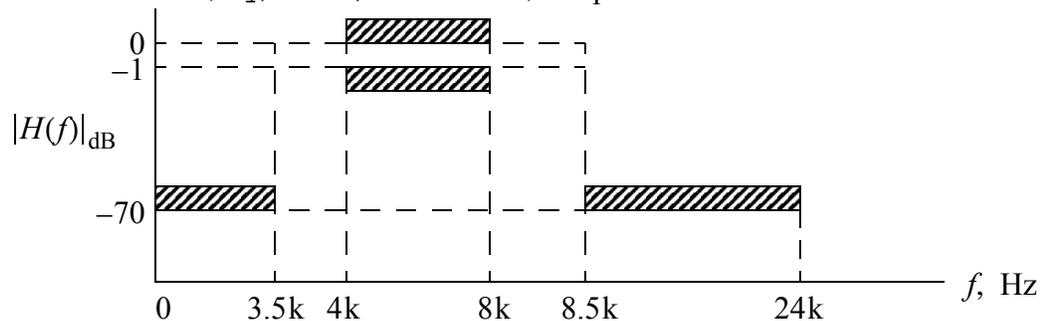
Make Note of the Following:

- This assignment will also be written up more like a lab report
- Include the observations made from the scope etc., also include a block diagram of how the test equipment is configured around the OMAP-L138 and AIC3106, etc.

Problems: Real-Time IIR Digital Filters

1. Cascade of Biquads IIR Using Floating-Point Coefficients

- a.) Using a 48 kHz sampling rate on the OMAP-L138 using the AIC3106, design an **elliptic** bandpass filter using `FDATool` that satisfies the amplitude response specifications shown below. The Web site contains the code ZIP package `sos_iir_float_AIC3106.zip`. The coefficients transferred to the workspace are in `double` format, but will be used as `float` once you create the C header file via the m-code function `sos_C_header(Hq, mode, filename)`. Explain how the double to float conversion performed in this function.



- b.) Provide MATLAB design information, including magnitude and phase response plots. Your plots should use a digital frequency axis scaled to the actual sampling frequency.
- c.) Verify the filter real-time frequency response by placing the coefficients in a working cascade of biquads IIR filtering program such as `ISRs_sos_iir_float.c`
- d.) For verification all you need to obtain is the frequency response magnitude in dB. The best approach is probably to use the vector network analyzer, but optionally you can drive the filter routine with the noise generator code used in Homework 3, and collect output signal samples on the PC using Goldwave.
- e.) Time your code with no optimization and with `-O3` optimization using the GPIO IRQ timing pulse. Estimate the maximum sampling rate possible without loss of real-time performance?

2. Via a simple MATLAB GUI slider control and the C8X_DEBUG interface, implement a variable center frequency notch filter of the form

$$H(e^{j\omega}) = \frac{1 - 2\cos\omega_0 z^{-1} + z^{-2}}{1 - 2r\cos\omega_0 z^{-1} + r^2 z^{-2}}, \quad 0 < \omega_0 < \pi$$

with $r = 0.9$. The parameter the MATLAB GUI adjusts will be ω_0 or have some relation to it. Use fixed or floating point for your design. Test the filter with a sampling frequency of 48 kHz, and verify using the network analyzer the ability to tune the notch around the interval $f_0 \in (0, f_s/2)$ Hz. What is the notch depth in dB when tuned to about 1 kHz? I will expect a sound demo of this in the lab. A nice test of this system is to sum an audio music source together with a single tone *jammer* from a function generator, and then see how well the tone can be suppressed from the music without coloration. With a second slider you can use r to adjust the filter bandwidth. As an audio source you can use an MP3 player/iPod, portable CD player, or Web radio.

3. Extend the peaking filter with MATLAB GUI example on page 8-44 to have at least four bands and equalization. It is suggested that you fix code the center frequencies, and tie together the filter Q across the all four peaking filters. The gain of each peaking filter needs to be an independent slider. I will expect a listening test with music.
4. Transfer the design of Problem 2 to the VC5505 or C5515, and use a lowly CCS 5.1 GEL file for tuning the notch center frequency. You will have to be very careful in how you deal with the fixed-point arithmetic in the single IIR biquad section.