Full Name: ______________________________

EEL 4750 / EEE 5502 (Fall 2019) – Code #07 **Due Date:** **Dec. 6, 2019**

Coding note: Throughout this assignment, when implementing a filter, use the MATLAB `conv` function. In addition, if you are filtering a signal `x` with filter impulse response `f` to get output `y`, use the code:

```
ytmp = conv(x,f);
y = ytmp(1:length(x));
```

That is, truncate the output to be the same length as the input. If you do not do this, keeping track of all signal lengths becomes very difficult.

Question #1: *(1 pts)* How many hours did you spend on this homework?

Question #2: *(4 pts)* *Filter Bank Analysis and Synthesis*

Provided with this assignment are two functions: `fb_analysis` and `fb_synthesis` which implement the analysis filterbank and synthesis filterbank, respectively. For these functions, the filter variables `h` and `g` are N-by-M matrices, where N is the length of the filters and M is the number of filters. Each function is currently incomplete, with a line stating `% ADD CODE HERE`.

(a) Complete the `fb_analysis` function for an input signal `x` and chosen set of filters `h`. Submit this code.

(b) Run the function below and manually verify that your output is correct (Note: you may use the `upsample` and `downsample` functions provided by MATLAB):

```
fb_analysis([0 1 1 -2 2], (1/sqrt(2))*[1 1; 1 -1])
```

(c) Complete the `fb_synthesis` function for the coefficients `v` and chosen set of filters `g`. Submit this code.

(d) Run the function below and manually verify that your output is correct (Note: the synthesis filters are not the time-reversed version of the previous analysis filters):

```
fb_synthesis([0 2; 1 0], (1/sqrt(2))*[1 1; 1 -1])
```

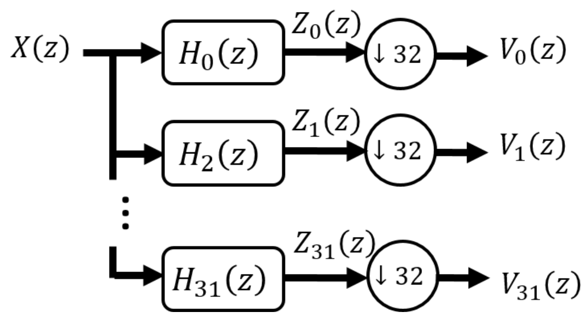
Question #3: *(8 pts)* *Filter Bank Implementation*

Now let's use the two previous functions to implement an orthogonal filter bank. For this implementation, use filter bank coefficients defined by a length-$2N$ modified discrete cosine transform:

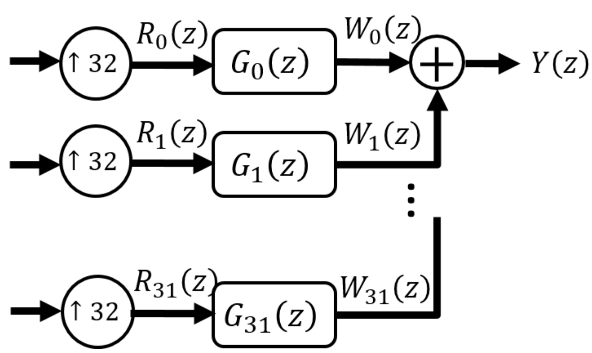
$$g_k[n] = \frac{1}{\sqrt{N}} \cos\left(\frac{\pi}{N}\left(n + \frac{N+1}{2}\right)\left(k + \frac{1}{2}\right)\right)$$

The modified discrete cosine filter bank is widely used as part of modern audio compression schemes (mp4, vorbis, etc.). For this question, load `code07_music.wav` (provided with this assignment) and implement each of the filter banks below with this audio.

(a) Implement the 32-channel analysis filter bank shown below to transform a signal $x[n]$ into the output of 32 different filter coefficient streams $v_m[n]$ (note that each filter will have a length of 64 samples in this case). Use `imagesc` to plot the transpose of the filter coefficients. The result should resemble a STFT. Show this plot.



(b) Implement a 32-channel synthesis filter bank shown below to transform the filter coefficients $v_m[n]$ into the output signal $y[n]$. Plot both the original input signal $x[n]$ and the reconstructed signal $y[n]$ side-by-side to confirm perfect reconstruction. Show these plots.



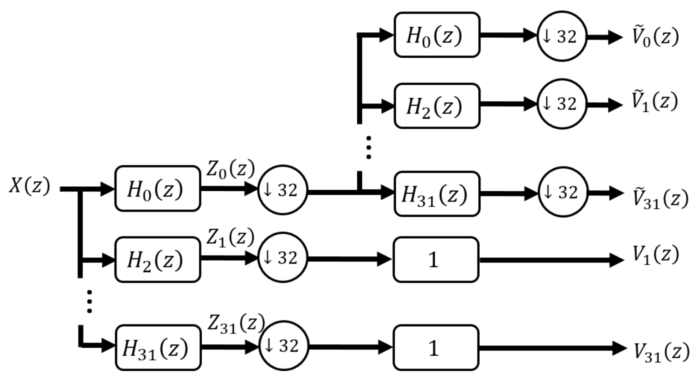
(c) There should be a delay of 63 samples between the input and output since we are using causal filters. Why is this the delay?

(d) Before inputting the filter coefficients $v_m[n]$ into the synthesis filter bank, manipulate them by computing the square of the coefficients. Input those modified coefficients into the synthesis filter bank. Explain how this changes the final output signal.

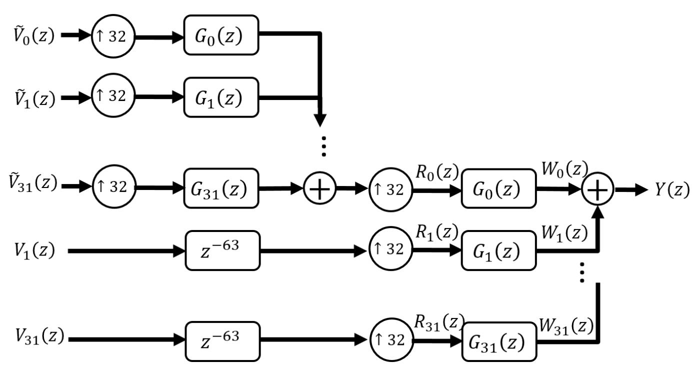
Question #4: *(8 pts)* *Filter Tree Implementation*

There is a lot of information in the lower frequency coefficients that we cannot easily manipulate in the previous problem. Therefore, in this question, we will apply another analysis filter bank to those coefficients to improve its frequency resolution. For this question, load `code07_music.wav` (provided with this assignment) and implement each of the filter banks below with this audio.

(a) Implement a 32-channel analysis filter bank tree, shown below. This is effectively adding another analysis filter bank to the result of Question #3. Use `imagesc` to plot the new $\widehat{v}_m[n]$ filter coefficients with a finer frequency resolution.



(b) Implement a 32-channel synthesis filter bank tree, shown below. This is effectively adding another synthesis filter bank to the result of Question #3. Note that extra delays are necessary to keep all of the outputs remain synchronized. A MATLAB function `delay` is provided to implement these. Plot both the original input signal $x[n]$ and the reconstructed signal $y[n]$ side-by-side to confirm perfect reconstruction. Show these plots.



(c) There should be a delay of $32 \times 63 + 63 = 2079$ samples between the input and output since we are using causal filters. Why is this the delay?

(d) Before inputting the filter coefficients $v_m[n]$ into the synthesis filter banks, manipulate them by computing the square of just the $\widehat{V}(z)$ coefficients. Input those modified coefficients into the synthesis filter banks. Explain how it changes the final output signal. How is this different than the Question #3 example?