

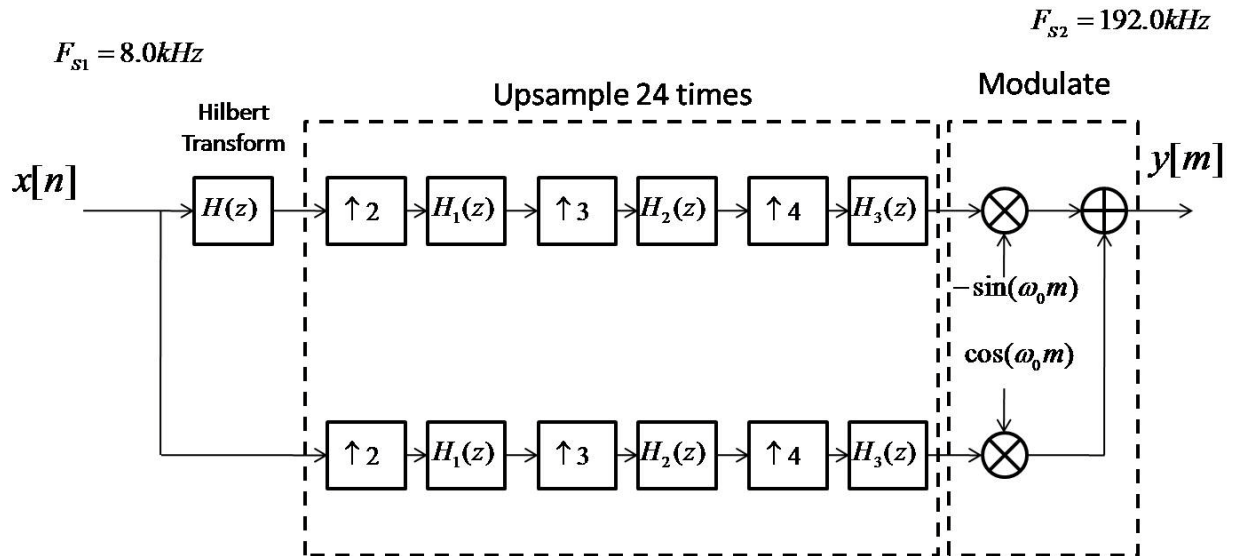
EC3400 Computer Project: Software Defined Radio

The aim of this project is to show how we can apply filter design techniques and multirate signal processing to design and implement a Software Defined Radio (SRD). In this approach the whole modulation and demodulation is implemented in software, thus making it easier to adapt to different schemes and standards. It builds on a previous project on Single Side Band modulation and the Hilbert Transform.

Introduction. An important technique in communications is *Amplitude Modulation (AM)*, where we shift the frequency spectrum of a signal, so that we can send it through a channel. The channel is defined by the frequency band associated to it. In the last project we have seen how to design and implement a SSB modulator, in a case where we do not have to resample the signal

In this project we extend the SSB project to the case where the carrier frequency is much higher than the sampling rate of the given signal. This is typical of a standard modem where the carrier can be of the order of (say) GHz, and the baseband signal sampled on the order of MHz.

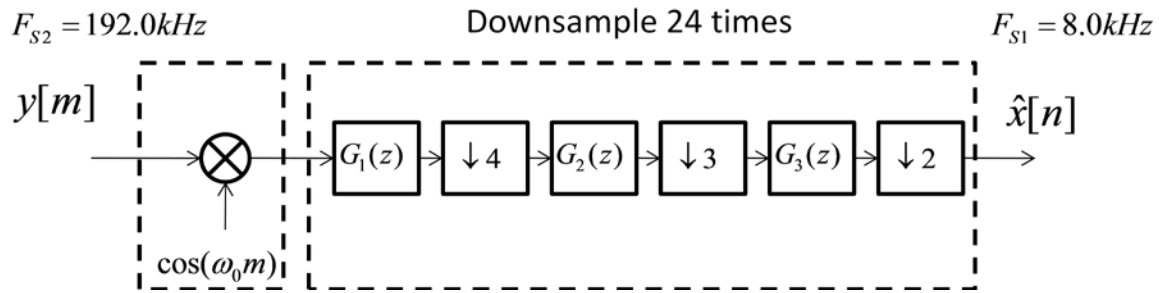
Problem 1. The figure below shows the modulator, where the $H(z)$ represents the FIR Hilbert filter, same as in the previous project.



The signal to be modulated has a bandwidth $B = 3.5\text{kHz}$, it is sampled at $F_{s1} = 8.0\text{kHz}$ and we want to SSB modulate it to a carrier with frequency $F_0 = 80.0\text{kHz}$. In order to do this, we upsample the signal by 24 times in three stages as shown. Design each filter as equiripple (using `firpm`) with an attenuation of 40dB in the stopband.

Problem 2. Check the modulator with the given signal “JH.WAV” and make sure that the modulated signal $y[m]$ has the desired frequency spectrum.

Problem 3. In a similar way, we can extend the demodulator to include downsampling as in the figure below.



Downsampling the signal 24 times is performed in three stages as shown. Design the three filters again as equiripple, with 40dB attenuation in the stopband.

Problem 4. Demodulate the signal from Problem 2 and check if it is the same you started with. You might have to rescale the amplitude, since modulation and resampling affects the amplitude of the signal.

Matlab Commands:

```
upsample(x,N)
downsample(x,N)
```