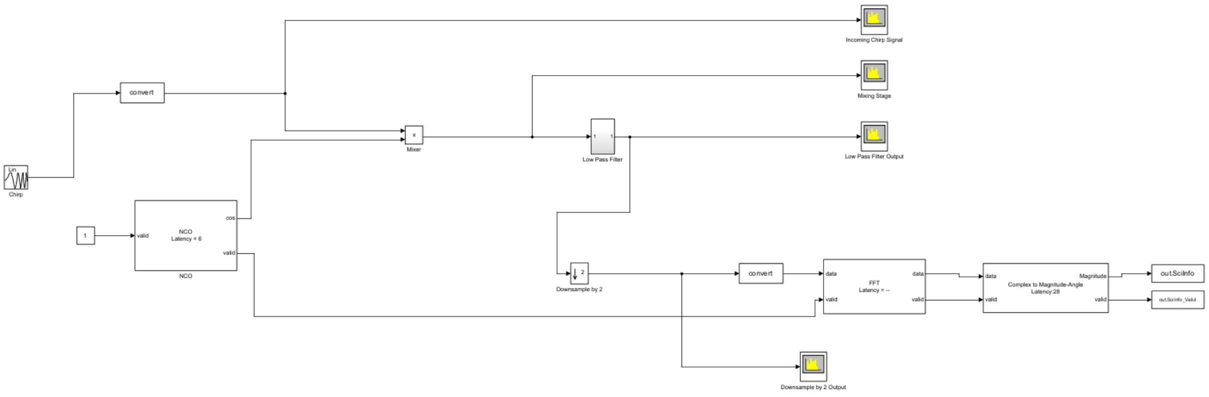
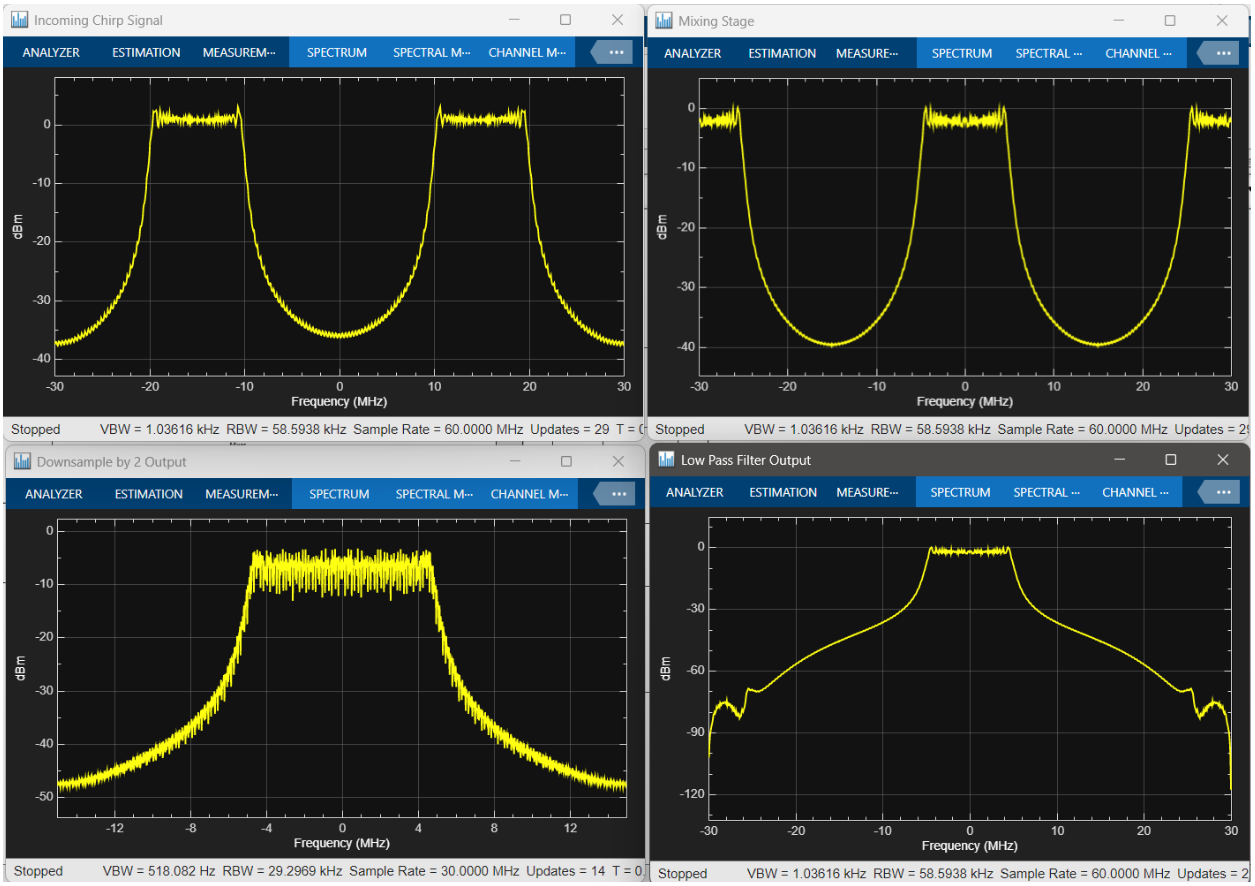
HW 13: SES 598

Simulink Model



Simulink Design



Stopped VBW = 1.03616 kHz RBW = 58.5938 kHz Sample Rate = 60.0000 MHz Updates = 29 T = 0

Stopped VBW = 1.03616 kHz RBW = 58.5938 kHz Sample Rate = 60.0000 MHz Updates = 2

Stopped VBW = 518.082 Hz RBW = 29.2969 kHz Sample Rate = 30.0000 MHz Updates = 14 T = 0

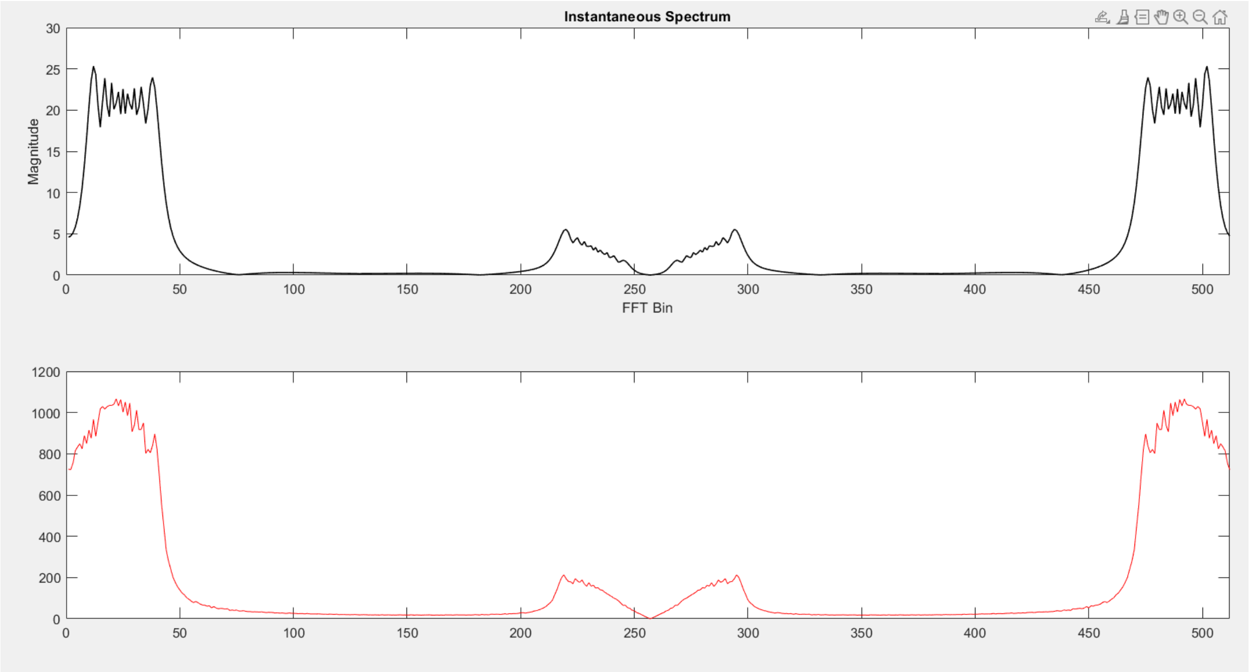
Stopped VBW = 1.03616 kHz RBW = 58.5938 kHz Sample Rate = 60.0000 MHz Updates = 2

Top left: Incoming Chirp Signal, Top right: Mixer Output, Bottom left: Downsample by 2 Output, Bottom right: LPF Output

MATLAB Script:

```
%% Cleanup
clc;
clear all;
close all;

%   DDC PARAMETERS
Fs = 60e6;
Fc = 15e6;
F0 = Fc;                 % Desired output frequency (Hz)
Deltaf = 0.05;           % Desired frequency resolution (Hz)
SFDR = 90;
Ts = 1/Fs;               % Sample period (s)
phOffd = pi/2;           % Desired phase offset (rad)

%% NCO Accumulator Size Calculation
N = ceil(log2(1/(Ts * Deltaf)));
Q = ceil((SFDR - 12)/6);
phIncr = round(F0 * 2^N * Ts);
phOff = (2^N * phOffd)/(2*pi);
ditherBits = N - Q;

%%   FIR LOW-PASS FILTER
L = 15;
Wn = Fc/(Fs/2);     % normalized cutoff (0.5)
win = blackman(L+1);
h = fir1(L, Wn, "low", win, "scale");
hf = fi(h, 1, 12, 11);      % fixed point representation

figure;
stem(h);
title("FIR Low-Pass Filter Coefficients");

figure;
freqz(h);
title("Frequency Response of Low-Pass Filter");


%%   RUN SIMULINK MODEL
% sim('YourDDC_SimulinkModelName');

%   EXTRACT FFT DATA FROM SIMULINK
FFT = 512;     % FFT size used in Simulink

%% Find the first index where valid == 1
k = find(out.SciInfo_Valid, 1, 'first');

% Extract magnitude starting at that valid index
fftData = out.SciInfo(k:end);
```

```
%%   PLOT FIRST (INSTANTANEOUS) FFT FRAME
a = (1:FFT)';

figure;
subplot(2,1,1);
plot(a, fftData(1:FFT), 'k', 'LineWidth', 1);
title("Instantaneous Spectrum");
xlabel("FFT Bin");
ylabel("Magnitude");
xlim([0 FFT]);

%%   SPECTRAL AVERAGING
specAvg = floor(length(fftData)/FFT);

% Keep only full frames
fftData_snr1 = fftData(1:specAvg * FFT);

% Reshape into a matrix of size 512 x (numAvg)
fftData_snr2 = reshape(fftData_snr1, FFT, specAvg);

% Sum along columns
fftData_snr3 = sum(fftData_snr2, 2);

subplot(2,1,2)
plot(a, fftData_snr3, 'r');
xlim([0 FFT])
```



Instantaneous FFT magnitude (top) and spectrally averaged magnitude (bottom) of the DDC output Averaging smooths the noisy instantaneous spectrum and stabilizes the frequency-domain response

In this assignment, a digital downconverter (DDC) was implemented in Simulink using a chirp input signal. The DDC included a Convert block, Mixer with an NCO-generated cosine, Low-Pass Filter, and Downsample-by-2 stage. To enable spectral analysis, an FFT block and a Complex-to-Magnitude block were added at the output of the DDC. The magnitude and its valid signal were exported to MATLAB using To Workspace blocks.

After simulating the DDC model, the resulting FFT magnitude samples were post-processed in MATLAB. The script identifies valid FFT frames, extracts the spectral bins, and plots the instantaneous spectrum, which represents the FFT of a single frame of chirp data. As expected for a chirp input, the instantaneous spectrum varies rapidly across frames and appears unstable.

Spectral averaging was then performed by reshaping the FFT data into multiple 512-bin frames and computing the sum (or mean) across frames. This averaging suppresses noise and stabilizes the frequency-domain output. The averaged spectrum appears smoother and demonstrates improved spectral stability, confirming the correct operation of the DDC, FFT, and MATLAB processing pipeline.